PQual: Automating Web Pages Qualitative Evaluation

Waleed Hashmi[1], **Moumena Chaqfeh**[1], **Lakshmi Subramanian**[2], **Yasir Zaki**[1]
[1]NYUAD, Abu Dhabi, UAE
[2]New York University, NYU, USA
{waleedhashmi,moumena,lakshmi,yasir.zaki}@nyu.edu

ABSTRACT
The increasing complexity of web pages has brought a number of solutions to offer simpler or lighter versions of these pages. The qualitative evaluation of the new versions is commonly carried out relying on user studies. In addition to the associated time and financial cost, running user studies became extremely challenging during the pandemic of COVID-19. Moreover, comparing the results of different user studies is difficult due to the participating users' subjectiveness. In this paper, we propose PQual, a tool that enables the automation of the qualitative evaluation of web pages using computer vision. In comparison to humans, PQual can effectively evaluate all the functionality of a web page, whereas the users might skip many of the functional elements during the evaluation.

Author Keywords
Qualitative Evaluation, Web Pages, Similarity Score

CCS Concepts
•**Human-centered computing** → **User studies;**

INTRODUCTION
The complexity of modern web pages has substantially grown in the past decade, leading to larger pages that are computationally intensive for mobile devices. The increasing trend in accessing these pages via handheld mobile devices [5] has recently brought novel solutions to tackle the complexity of web pages [14], including developer tools [12], in-browser tools [10, 16, 17, 13], platform-based solutions [1, 2], and new browsers [3, 6]. Google has also attempted to offer lighter versions of web pages through the Accelerated Mobile Pages (AMP) [11] approach. While AMP aims to redefine how pages should be written, JSCleaner [8] aims to rapidly create lighter versions of existing pages for an improved browsing experience.

To evaluate the quality of the pages generated by these solutions, user studies are usually conducted. However, the shutdown that resulted from the COVID-19 pandemic has brought alternative evaluation approaches in human-computer interaction, especially when researchers are unable to directly interact with the users. These alternatives can be considered not only due to the pandemic situation, but also for reasons correlated with other constraints such as travel, finance, or health. In addition to the alternatives to lab-based user studies discussed in [15], we propose to utilize computer vision in situations where the technology can replace the human role in qualitative assessment. This also brings additional benefits to researchers in comparing their results with other solutions using a unified tool without relying on user studies that might be prone to subjectivity.

In this work, we present *PQual*; an evaluation tool that computes the similarity score of a web page offered by a web complexity solution in comparison to the corresponding original page. To evaluate the potential loss of functionality during the generation of the page to be offered, the similarity score considers not only the static structural view of the page but also the functionality of the different page elements. The loss in the functionality of a page element might be either complete (when the element is completely missing) or partial (when the element is no longer interactive). To the best of our knowledge, this might be the first attempt to assess the quality of the web pages offered by different solutions without conducting a user study.

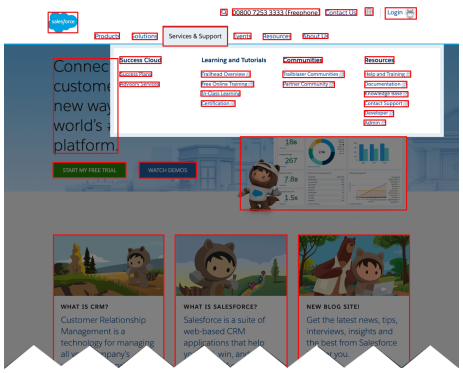

Figure 1. Extracted components marked on a web page

THE TOOL
PQual considers two input web pages in the similarity assessment. The first represents the original page, and the second represents the lighter counterpart page that is generated by a web complexity solution. We refer to the first page as the

UIST '20 Adjunct, October 20–23, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7515-3/20/10.
https://doi.org/10.1145/3379350.3416163

Original and to the second as the *New* page. *PQual* performs three main steps to output a similarity score for a qualitative assessment of the new page in comparison to the original, which are: identifying actionable tags, extracting components from these tags, and matching the extracted components. *PQual* is implemented using Selenium driver [4] to perform actions on HTML tags of a web page, and OpenCV [7] to check if the performed actions have caused visual changes on that page.

Identifying and Recording Actionable Tags

PQual starts by identifying and recording the HTML tags in both pages (the original and the new) that cause structural page changes when the user interacts with them. We refer to these tags as *actionable tags*. Two main types of user interactivity events are considered in identifying these tags: hover and click. To identify these tags, the structural similarity of a page screenshot captured before the occurrence of a tag event is compared to the screenshot captured after the occurrence of the event, where less than 99.5% similarity identifies the tag to be actionable. For a given page, we refer to the set of actionable tags' screenshots as *page snapshots*. Figure 1 shows a sample snapshot of a hover event over a div tag with class = "menu-item-container". The snapshots of both the original page and the new page are stored to create a baseline for computing the similarity score. Instead of considering all HTML tags, the identification of actionable tags has reduced the overall complexity of the computation by an average of 62.8%.

Extracting Components from the Snapshots

PQual extracts the basic components from the snapshots of both the original and the new page to generate two sets of components, one set for each page. Components are extracted by removing the background pixels and finding the interconnectedness between the dilated foreground pixel. An example of the extracted components can be seen in Figure 1 where each red rectangle represents an individual component. This process is essential due to two reasons: First, a simple pixel-by-pixel snapshot comparison would not give an accurate score due to the dynamic nature of HTML. For example, if an element (such as an ad) is missing in the new page, the HTML would shift the relative position of all subsequent elements in the page. Second, the background pixels of web pages generally cover large areas, which (if included in the comparison) would result in a false higher score.

Matching Components and Computing the Score

This step aims at matching the components extracted from the original page snapshots to the components extracted from the new page snapshots, which is performed using *Image Integrals* [9] search algorithm. *PQual* gives a score of 1 for components with exact matches, while it looks for partial matches using OpenCV's structural similarity score (which outputs a value between 0 and 1) for each component with no exact match. The match with the highest score is considered for a given component with no exact match. When all the components are matched, a weighted average is computed using the area of the components (since it is more probable for larger components to have more content). This process is repeated for all the snapshots to compute a final page similarity score by averaging all of the snapshots' scores.

EVALUATION

We selected 100 popular web pages with their counterpart versions simplified using [8] and used *PQual* to compute the similarity score of the simplified versions in comparison to the original web pages. We make the set of paired pages available online for user evaluation. Twenty-two users were recruited from an international university campus by posting online ads on their popular social media groups. They were informed to spend a maximum of 30 minutes to evaluate a set of paired pages that were shown side-by-side. An institutional review board (IRB) approval was given to conduct the user study. The users were asked to evaluate the similarity between each pair of pages that were displayed within the time limit, by answering the following questions:

- Rate the look similarity and styling on a scale from 0 to 10.
- Rate the content similarity on a scale from 0 to 10.
- Mention all types of missing contents that apply: (text, images, advertisements, video, layout/beautifiers, others).

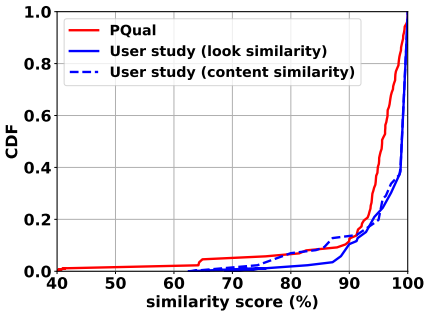

Figure 2. The similarity score CDF of PQual in comparison to the similarity score CDF of the user study

Figure 2 shows the page similarity cumulative distribution function (CDF) between the simplified and the original pages for both the user study and *PQual*. Results show that for almost 90% of the pages, the simplified pages achieve a similarity score above 90% compared to their originals. This is true for both PQual and the user study. Results also show that *PQual* computes a comparable similarity score to the user study scores, with minor differences. It can be seen that *PQual* is less forgiving than the users, evident by the smooth and gradual increase of the scores between 90% and 100%. In contrast, the user study results show a sudden and sharp increase in the score within the same segment. This can be explained by the fact that real users tend to overlook minute differences between pages, and that they are more forgiving in their assessment when major elements in the two pages are matching.

CONCLUSION

PQual offers a unified approach for qualitative evaluation of web pages, and can inspire alternative evaluation approaches in human-computer interaction.

REFERENCES

[1] 2015. Apple News - Apple. (2015). https://www.apple.com/apple-news/ Accessed: 2020-03-21.

[2] 2015. Instant Articles | Facebook. (2015). https://instantarticles.fb.com/ Accessed: 2020-03-21.

[3] 2017. What is Amazon Silk. (2017). https://docs.aws.amazon.com/silk/latest/developerguide/introduction.html Accessed: 2020-03-21.

[4] 2019. Selenium WebDriver. Browser Automation. https://www.seleniumhq.org/projects/webdriver/. (2019). Accessed: 2019-05-14.

[5] 2019. The State of Mobile Internet Connectivity 2019. https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2019/07/GSMA-State-of-Mobile-Internet-Connectivity-Report-2019.pdf. (2019). Accessed: 2020-03-17.

[6] Andreas Bovens. 2015. Opera Browsers, Modes & Engines. (2015). https://dev.opera.com/articles/browsers-modes-engines/ Accessed: 2020-03-21.

[7] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[8] Moumena Chaqfeh, Yasir Zaki, Jacinta Hu, and Lakshmi Subramanian. 2020. JSCleaner: De-Cluttering Mobile Webpages Through JavaScript Cleanup. In *Proceedings of The Web Conference 2020*. 763–773.

[9] Franklin C Crow. 1984. Summed-area tables for texture mapping. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*. 207–212.

[10] Mohammad Ghasemisharif, Peter Snyder, Andrius Aucinas, and Benjamin Livshits. 2018. SpeedReader: Reader Mode Made Fast and Private. *CoRR* abs/1811.03661 (2018). http://arxiv.org/abs/1811.03661

[11] Google. 2019. AMP is a web component framework to easily create user-first web experiences - amp.dev. https://amp.dev. (2019). Accessed: 2019-05-05.

[12] Sarah Lim, Joshua Hibschman, Haoqi Zhang, and Eleanor O'Rourke. 2018. Ply: A visual web inspector for learning from professional webpages. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 991–1002.

[13] Ravi Netravali, Ameesh Goyal, James Mickens, and Hari Balakrishnan. 2016. Polaris: Faster Page Loads Using Fine-grained Dependency Tracking. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. USENIX Association, Santa Clara, CA. https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/netravali

[14] Ravi Netravali, Vikram Nathan, James Mickens, and Hari Balakrishnan. 2018. Vesper: Measuring Time-to-Interactivity for Web Pages. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*. USENIX Association, Renton, WA, 217–231. https://www.usenix.org/conference/nsdi18/presentation/netravali-vesper

[15] Albrecht Schmidt and Florian Alt. 2020. Evaluation in Human-Computer Interaction - Beyond Lab Studies. https://amp.ubicomp.net/wp-content/uploads/2020/04/Evaluation-in-Human-Computer-Interaction-Beyond-Lab-Studies.pdf. (2020). Accessed: 2020-07-01.

[16] Xiao Sophia Wang, Aruna Balasubramanian, Arvind Krishnamurthy, and David Wetherall. 2013. Demystifying Page Load Performance with WProf. In *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*. USENIX, Lombard, IL, 473–485. https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/wang_xiao

[17] Xiao Sophia Wang, Arvind Krishnamurthy, and David Wetherall. 2016. Speeding up Web Page Loads with Shandian. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. USENIX Association, Santa Clara, CA, 109–122. https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/wang