Automated Evaluation of WordPress Theme Design

Derek Ohanesian[1]

November 25, 2014

[1]Advised by Professor Chris Fernandes

Abstract

A major obstacle in publishing a website is developing a well-designed user interface for the content being published. The World Wide Web democratizes publishing by providing an open platform on which even amateur web publishers can communicate, but a well-designed website remains less accessible to amateur publishers. Past research has focused on finding quantitative heuristics that could be used to provide useful design feedback to non-professional web designers. By analyzing the well-defined functions for displaying content in WordPress themes, as well as the CSS style sheets from those themes, a WordPress theme's design can be heuristically evaluated. The result of this evaluation can provide the theme's designers with feedback and suggestions for design improvement. For this research project, the following heuristics measured: color quantity, color contrast, balance, density of content, font size, and font quantity. Data was collected for over 200 WordPress theme designs and analyzed for these heuristics. The feedback that this automated system provides was then compared to a human heuristic evaluation, to assess the value of the feedback that the automated system provides.

Contents

List of Figures

List of Tables

1 Introduction

This research project began by asking if the design of web pages could be automated. The architecture of the World Wide Web separates the content of a website (HTML) from the presentation and design of the website (CSS). Given the content of a website and its HTML markup, could the CSS be generated? The content on the web today is displayed with a great variety of user interfaces, even for similar content. Blogs, for example, all have similar content and structure, but the user interface to present a blog's content varies greatly between different blogs. Instead of looking at the design of any site on the web, we focused the scope of this project to just two-column WordPress blogs. Because these blogs all have the same type of content, the variation in design can be isolated and studied. Why do blogs display the same content differently, and is one user interface better than another for the same content? By collecting heuristic data about the design of blogs from a database of free and open-source WordPress themes, we can determine trends that could provide web designers with an automated evaluation of the user interface of their designs.

1.1 Motivation

The World Wide Web is an immense catalog of media and information. The success of the Web is in part due to the low barrier of entry. Anyone with Internet access can publish their own website at little or no cost, and potentially reach an audience of millions. The Web provides the platform for showcasing content, and because of the variety of categories of content on the Web, there are a variety of user interfaces for displaying this content. Web design is as much a creative industry as it is technical. Each website has its own look and

feel. As the Web has evolved, many best practices and standards for the design, structure, and technical markup of websites have developed. These standards and best practices affect the speed, usability, accessibility and compatibility of a website.

The criterion for creating modern websites is becoming more complex as the Web evolves, and this threatens the low barrier of entry on the Web. It would be beneficial to a webmaster if he or she could focus on the content of their website and not need to be concerned about growing intricacies of web design. Given content to be published, can we automatically generate a website that adheres to modern web design standards and best practices?

The visual elements of websites are defined by HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets). HTML expresses the content and its structure, while CSS defines its appearance. The HTML of a website should be well structured and follow modern standards. Modern websites are written in HTML 5 and mark up content semantically according the standards of the language. CSS is simply for appearance. Implementing CSS properly means making a website usable or attractive. There are plenty of websites that have CSS implemented well for usability or to make the website look the way it was intended. Other websites, however, might have CSS that takes away from the website's usability. Many websites that are dynamically generated won't have well-designed CSS at all, because the time invested in improving that website is not worth the added enhancement. Many amateur websites simply do a poor job of implementing both HTML and CSS. If amateur webmasters had the expertise to construct websites properly, then the benefits in speed, appearance, compatibility, usability, accessibility, and ultimately traffic, would be significant.

By automating the development of websites, webmasters can publish content without worrying about the changing complexities of website design. This is vital to the World Wide

Web. The low barrier of entry is what makes the web so successful, and by automation the design of web pages, this barrier can remain low. The enhancement of user interface design on the web is an added benefit.

1.2 Scope of Research Question

Because of the growing complexity of HTML, especially with the enhancements in HTML 5, it is often easiest for web publishers to use a content management system (CMS) to generate the HTML and CSS for the content being published. In many ways, this lowers the barrier of entry to publishing. The most popular contentment management system for the web is WordPress. Wordpress separates the design of the website from the content of the website in a more complex way than just HTML and CSS. Publishers define their content in WordPress's back-end, and the presentation of the content is controlled by a WordPress *theme*. WordPress includes a catalog of free and open-source themes on the projects website (http://wordpress.org). These themes define not only the CSS of the page, but also the structure of the HTML. By installing a different theme on the same WordPress content, the user interface of the page will be changed without changing the content.

By installing different themes on the same WordPress content, we can study the design of the theme. By performing a heuristic evaluation of each theme, we can find trends that can offer web design advice to current and future theme designs. The purpose of this research was to evaluate a set of two-column Wordpress themes and determine what valuable advice, if any, could be provided to theme designers.

2 Background and Related Work

To automate the design of a website, it is important to understand not just the state of the art of web design from a technical perspective, but also from a content perspective. Different categories of content presumably need different structured web pages.

Research has shown that the category of website has a significant impact on the structure of that website [5]. A personal website will have a different structure than a corporate website. Shops have a distinct structure that is substantially different from that of blogs. Because of these differences, it will not be possible to apply a blanket "best" user interface to every website. Different kinds of websites will need to use different heuristics to evaluate the user interface. Today's websites increasingly have a template that is constant across the entire website for navigation and branding. The content portion of the website is the only portion that changes from page to page. Gibson *et al.* [3] have shown that as the web evolves, more websites have been using larger template areas. More interestingly, the style of the template (in this case the size) changes with the category of website, providing further evidence that different categories of website are subject to different structures [3]. For these reasons, it is important to restrict the scope of this project. By only evaluating two-column style WordPress themes, we can make a reasonable assumption that the design of all two-column style blogs can be evaluated using the same heuristics.

Designing a visual appearance for a website can be subjective. There are, however, some metrics that can provide measures of whether a design is of high quality. Research has been done on providing metrics for usability evaluation of web sites. Ivory *et al.* [6] analyzed Webby award-winning websites to show that factors such as link count, word count, color

count, graphic count, etc. contribute to a website design's quality. Another study by Ivory *et al.* [4] further studied web design metrics over a three-year period, comparing design patterns such as font sizes, validity of HTML, color combinations, and use of graphics. The heuristics used in these studies inspired the heuristics that were implemented in this project.

In 1997, Perkowitz *et al.* [7] proposed that websites of the future could use artificial intelligence to adapt to changing content and usage patterns. While Perkowitz *et al.* were mainly concerned with adapting current designs based on usage patterns, it introduced the idea of web design and adaptability being the responsibility of the web page and not of the webmaster.

3 Data Collection

We downloaded a set of WordPress themes from available themes on http://wordpress.org/. Filtering by "two-column" themes, this currently yields 1461 themes. A subset of about 200 of these themes were analyzed to collect data that could be used to heuristically evaluate them. A separate text set was set aside for evaluation later.

3.1 WordPress Theme Anatomy

WordPress themes are written in PHP, so the theme only generates HTML for the client. From the client-side data alone, it would be possible to examine each HTML element in the page and collect data about each element's style attributes. HTML also provides some semantic labels for data. For example, an H1 tag signifies a heading, and a P tag a paragraph. Because these WordPress themes are open-source, however, we can examine the PHP source

code and gain more information about what content is actually being displayed in the generated HTML. WordPress themes include content through "template tags". Each template tag displays a specific piece of content within the theme. For example, the bloginfo() method displays the title and tagline of the WordPress blog (given the appropriate parameters). By collecting this additional data, we can know that not only are we analyzing a heading on a page, but that the heading is actually the title of the blog.

3.2 Setup

WordPress was installed locally on a Mac OS X, Apache, MySQL, and PHP stack. The WordPress plugin Theme Test Drive [1] was installed within WordPress to allow for easy switching between themes using a URL parameter. The a set of themes was installed by copying each theme to the /wp-content/themes/ directory of the WordPress installation.

Each theme folder includes a functions.php file which defines functions specific to that WordPress theme. Using a PHP script, we traversed each theme directory and if the theme did not have a functions.php file already, we created one. To each theme's functions.php file we added a line of code to include our custom functions. Using a PHP include, we can use this to add a set of our own functions to every WordPress theme. The same PHP file with additional function was included in every theme.

The additional functions the we added to each WordPress theme were used to intercept calls to template tags. For example, when the theme calls the bloginfo() function to get the title of the blog, instead of the regular response, a separate function mybloginfo() will be called. mybloginfo() will call the bloginfo() function and return the resulting string, except

with a label appended. This label will semantically identify the string as the title of the website. While it would be convenient to append this label as an HTML tag, WordPress strips out HTML, so the label is simply a unique string. This process is achieved by defining WordPress "filters" in our custom functions.php file.

In addition to adding these functions to each theme, we also add the name of each theme to a 'themes' table in a MySQL database, so we can iterate through them.

3.3 Procedure

In order to collect data about each WordPress theme, it is necessary to render each theme in the browser. This process was automated using a browser extension for Safari. The browser extension takes in a number parameter in the URL and uses it to look up the next theme in the database by incrementing the URL parameter and changing the theme parameter to the next theme in the 'themes' database. The WordPress plugin then loads the next theme. When each theme is loaded in the browser, our Safari extension collects data about each theme and saves it to a second MySQL database.

Before we can collect data about the theme we have to remove the labels that have been added to the HTML of the page when select WordPress template tags were called. Using a regular expression, we replace these tags with HTML tags that are invisible to the design of the page.

We can then traverse each HTML element on the page and look for these labeled elements to save data about them to our database. In the MySQL database 'data', we store the name of the theme, the label of the element being analyzed, the name of the attribute of that

element, and the value of that attribute.

3.4 Elements Labeled and Data Collected

In addition to attributes of specifically labeled WordPress elements, general data was also collected about each theme that did not require access to the server-side WordPress theme. For every element on the page, (not just labeled elements) the text color and background color were recorded. The following types of WordPress elements were labeled with HTML tags signifying that they contain a specific template tag: the title of the blog, the tagline of the blog, the title of a blog post, the author of a blog post, and the content of a blog post. If any of these elements were repeated on the webpage, then data is collected about each instance. For each of these elements, the height and width were recorded, as well as the X/Y coordinates of each element. Additionally, the font size of the first instance of the blog title was recorded. A sample layout of these elements can by seen in the sample WordPress website shown in Figure 1. The following WordPress template tag functions were filtered to provide labeled sections of a blog theme's layout:

1) bloginfo() - with parameters that return the blog title
2) bloginfo() - with parameters that return the blog tagline
3) the_author() - the author's name on blog posts
4) the_title() - the title of individual posts
5) the_excerpt() - an abbreviated version of a blog post
6) the_content() - the full content of a blog post

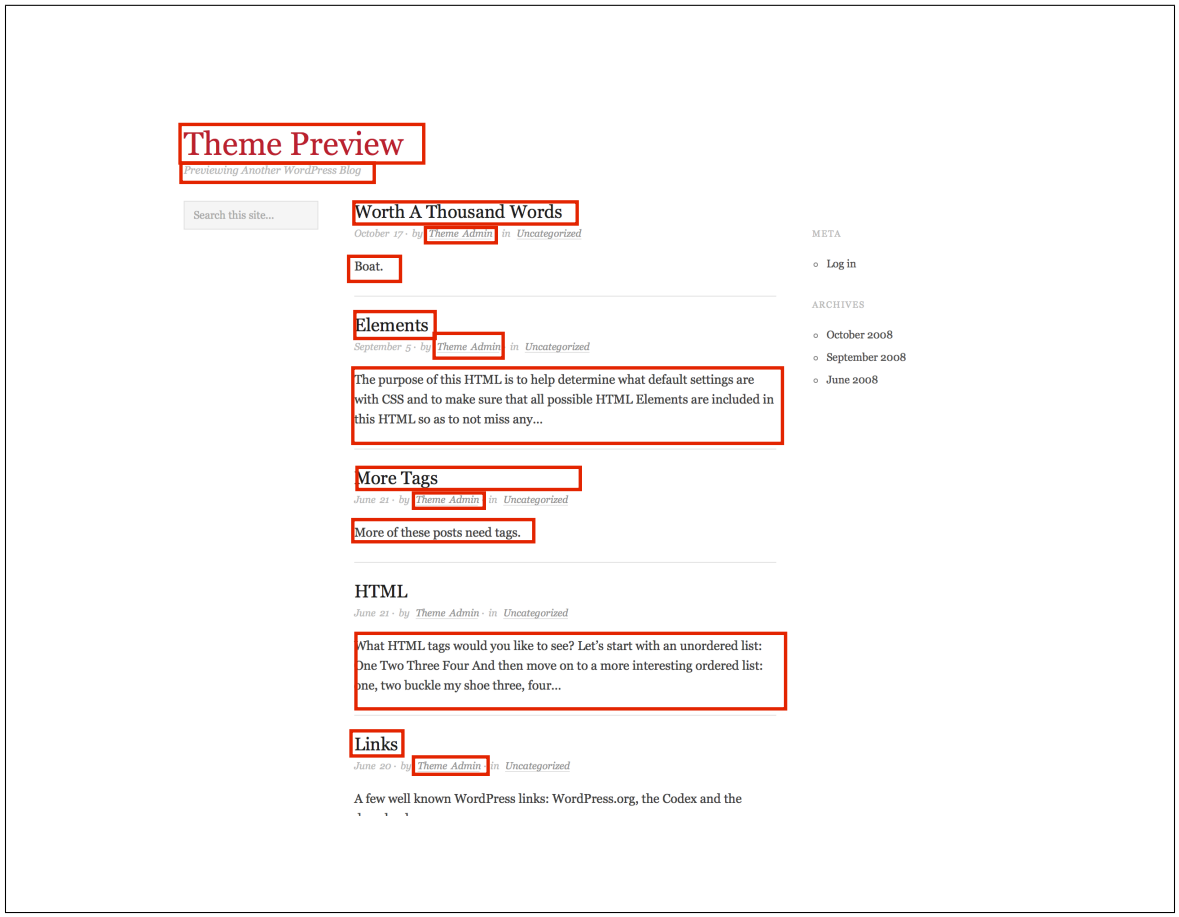

Figure 1: Sample WordPress theme with labeled elements outlined in red for data collection.

3.5 Heuristic Evaluation and Scoring

Data is collected from each theme using the Safari extension and inserted into a MySQL database. After running this process, the database contains data from about 200 WordPress themes. Using this data, a PHP script is run to score each theme in the categories for heuristic evaluation outlined in the preceding section.

4 WordPress Theme Design Heuristics

Inspired by the heuristics discovered to correlate with good design in previous research, as well as experience with design principles for the web, we automated the evaluation and scoring of each theme's use of color, balance, density, and fonts.

4.1 Colors

From each theme, the Safari browser extension collected the total number of colors used in every HTML element on the page. Because color is sometimes expressed as RGB values and sometimes as RGBA values, some colors in the set may be repeated, and the addition of alpha channels can make data sometimes unpredictable. The number of colors from each theme is directly saved to a the 'grades' MySQL database for each theme. In Figure 2, we can see the distribution of the number of colors used by each theme. We can see from these numbers that 5 color themes are most popular. Themes that use a color palette with 5 colors are not only popular but probably indicative of good design principles. This shows that many designers are working from a fixed color palette of the same size. Any number of

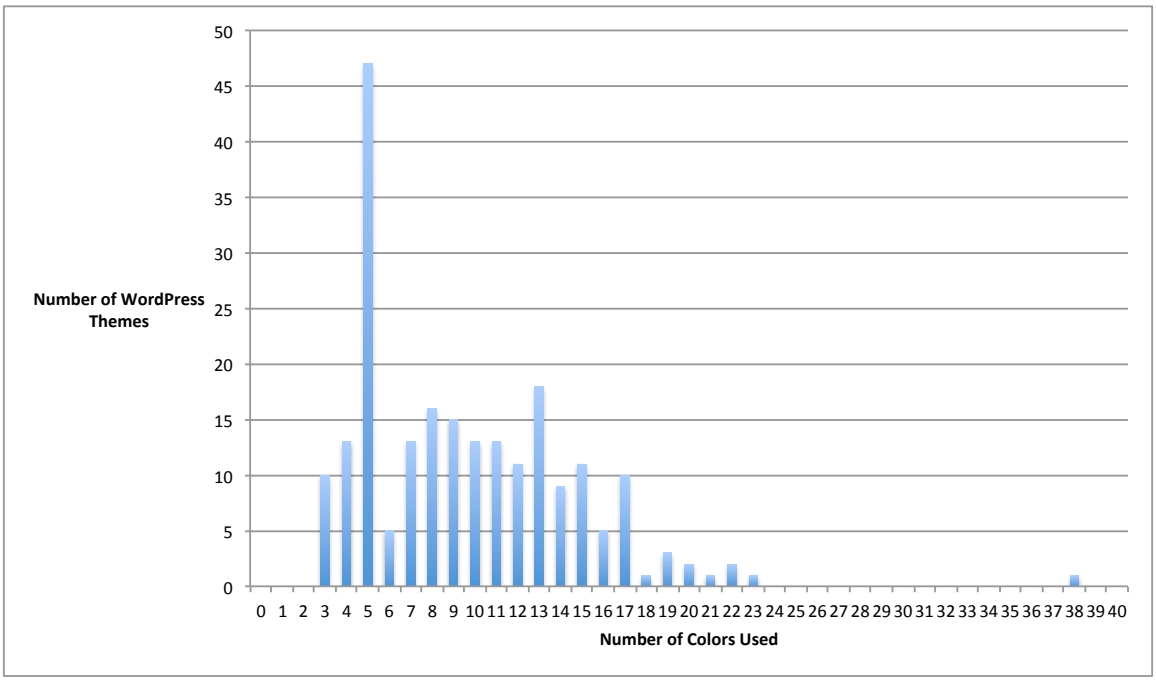

Figure 2: Distribution of the number of WordPress themes using each number of colors.

colors from 3 to 17 is popular. Few themes use more colors, with one outlier that used 38 colors.

In addition to the number of colors, the color color contrast of foreground text on background was collected. The information was stored in the database as pairs of RGB or RGBA values. The World Wide Web Consortium (W3C) recommends a specific amount of color contrast for accessibility as part of the Web Content Accessibility Guidelines (WCAG).[2] Color pairs were graded against this algorithm, to produce a score for this heuristic. The final grade was represented as the percent of color pairs in the theme that pass the WCAG test. There was a broad mix of data for this heuristic. Many WordPress themes received a 17% of Themes received a 0% indicating that the theme had no passing color combinations.

33% of themes received a 100%, indicating that all color combinations passed the accessibility test. The remaining 50% were distributed fairly evenly between the two extremes. The surprising number of failed results, on what looked like themes with quite readable color combinations could be do to a poor testing methodology. The algorithm used to text for contrast was from the WCAG 1.0. The WCAG 2.0 specification includes a much better color contrast algorithm that better accounts for luminance. The scoring script for this heuristic could be improved to include that updated algorithm.

4.2 Fonts

There is additional work to be done with font heuristics, such as counting typefaces used. To begin with, we counted the font sizes used for the tile of each WordPress blog. This data is show in Figure 3. Theme developers often choose to display the title of the blog multiple times on a page, so we assumed that the first instance of the title is comparable across themes. Interestingly, font size of titles was distributed across popular presets (16px, 24px, 36px, 48px). This either shows that not much care was taken in picking font size, or that presets are an example of good design and can act as familiar sizes to readers.

4.3 Balance

For each of the WordPress elements labeled by our data collection algorithm, we calculated whether each element rendered on the left or right side of the page. Elements that overlapped with the center of the page were not counter towards either side. We did not track every element on the page for balance, just the labeled WordPress elements, assuming that they

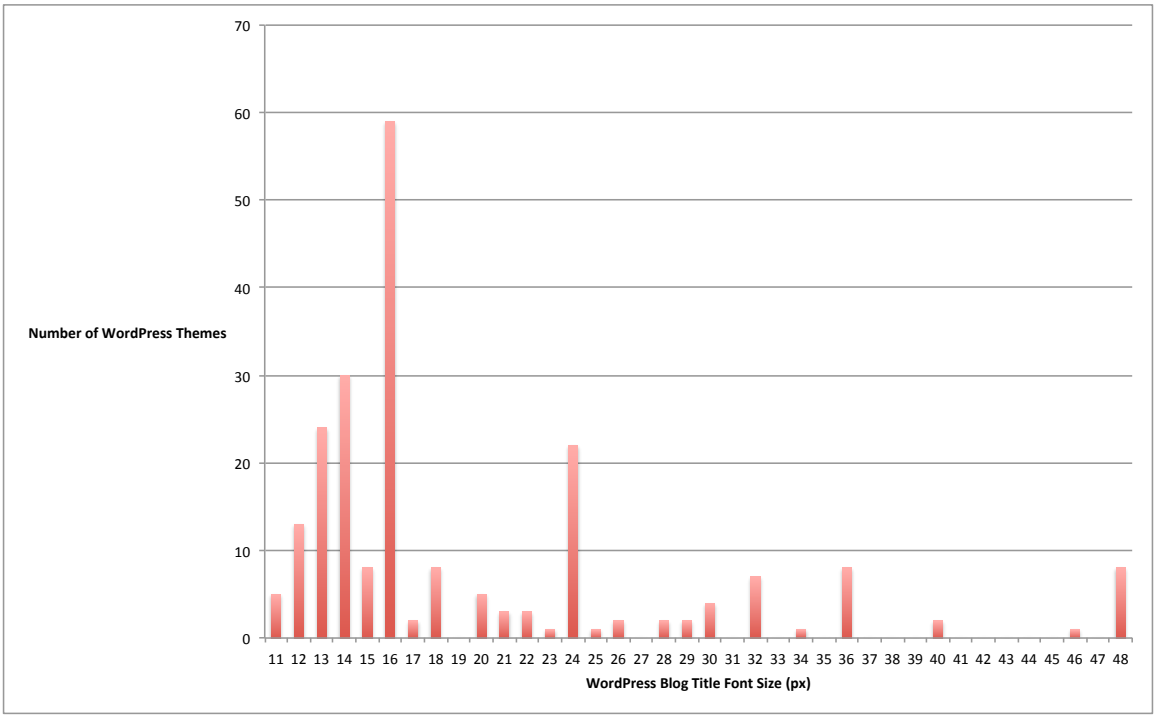

Figure 3: Distribution of the number of WordPress blog title font size (px).

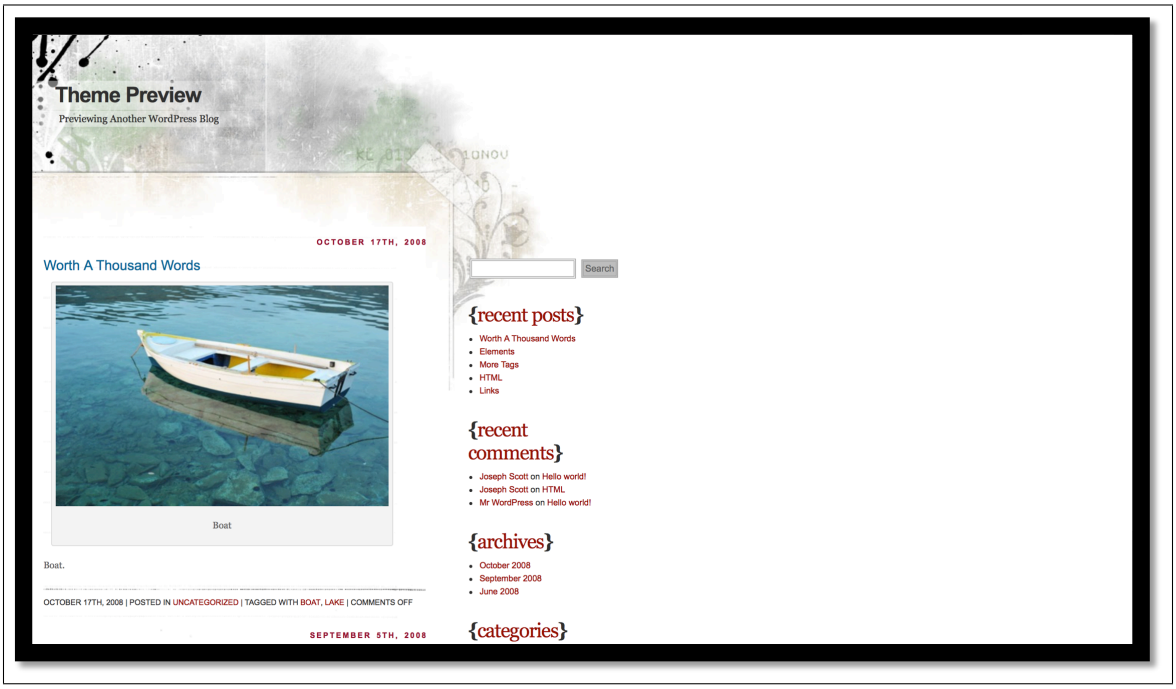

Figure 4: Unbalanced WordPress theme on a 1440x900 display.

represented most of the content. A score for balance was represented in our database as a percentage. 100% represents all of the tracked content falling on the left side of the page. 0% represents all of the content falling on the right side of the page. A balanced theme would have a score of 50%. Figure 4 shows a theme that scored a 100%; 100% of tracked content falls on the left side of the page. Pages were rendered for this heuristic using the full screen on a display with a resolution of 1440x900. Often, the balance of a theme would depend on the size of the browser window. The score, however, is still justified. If a theme is unbalanced on just one display, even at an unpopular resolution, it could still be an important factor in the quality of the design.

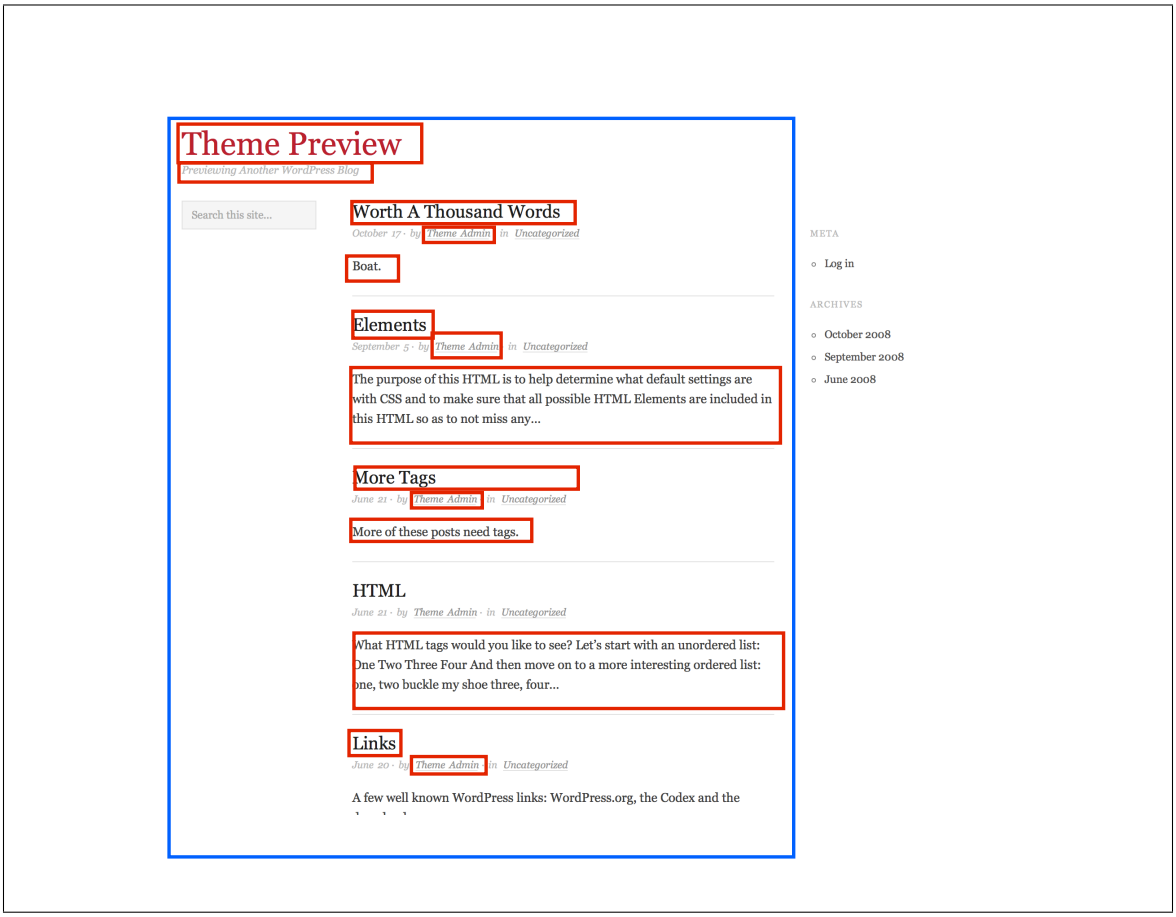

Figure 5: Blue bounding box for calculating density of red tracked elements.

4.4 Density

The density of the content was calculated and expressed as a percent of the total area used. Instead of using the total area of the page, we created a bounding box around all of the tracked elements. The percentage of the area used within this bounding box gave us the score for the density heuristic. Figure 5 shows the red boxes as tracked content consuming a large area of the blue bounding box. This sample theme would have a high content density.

5 Evaluation

We have set aside a test set of themes which have not undergone a heuristic evaluation. We chose 10 of these themes randomly and did a human evaluation of each for the following heuristics: number of colors, color contrast, balance, and density. If, based on our knowledge of user interfaces and web design, these heuristics contributed to or detracted from the design and usability of the website, then we assigned a score of "pass" or "fail" accordingly. to the to an evaluation of the same theme heuristics. We then ran our automated evaluation for the same heuristics, to assess the effectiveness of the automated algorithm.

Theme	**Colors**	**Contrast**	**Balance**	**Density**
kotenhanagara	Pass	Fail	Pass	Pass
matala	Pass	Fail	Pass	Pass
puddle	Pass	Fail	Pass	Fail
rgb	Pass	Pass	Pass	Fail
appointment	Fail	Fail	Pass	Pass
timeless	Fail	Pass	Pass	Pass
suffusion	Pass	Fail	Pass	Pass
spartan	Pass	Pass	Pass	Fail
random-background	Fail	Fail	Pass	Fail
autoadjust	Pass	Pass	Pass	Pass

Table 1: Human Heuristic Evaluation of 10 WordPress themes

Table 1 shows the human evaluation of each theme, while Table 2 shows the automated evaluation numbers. The "pass" and "fail" values can be used to inform the effectiveness of the automated scoring algorithm. These values can also be used to show which heuristics can be best predicted and which are most important to the design.

Table 3 shows a variety of values comparing passing scores to failing scores. For the number of colors heuristic, we cannot just compare averages, because there may be such a

Theme	Colors	Contrast	Balance	Density
kotenhanagara	4	0	100	6.3
matala	3	0	100	29.3
puddle	9	75	100	5.5
rgb	11	60	80	17.0
appointment	15	50	20	8.9
timeless	4	100	100	3.76
suffusion	9	50	75	12.29
spartan	12	85.7	100	15.19
random-background	3	0	100	13.5
autoadjust	8	0	100	34.21

Table 2: Automated Heuristic Evaluation of 10 WordPress themes from human evolution.

thing as both too few and too many colors and an average would remove this detail. We can see, however, that the average number of colors for passing scores was 8. Interestingly, the three failing scores were both for themes with the two lowest numbers of colors (3 and 4) as well as the theme with the highest number of colors (15).

Theme	Colors	Contrast	Balance	Density
Passing	8 average	61.4% average	All	15.89% average
Failing	3, 4, 15	29.17% average	None	12.8% average

Table 3: Notes and comparison of passing and failing of human scores compared to automated scores.

The contrast heuristic shows a higher (61.4% compared to 29.17%) average level of contrast between text, informing us that the automated algorithm is producing usable data, even if it is imprecise.

The balance heuristic is hard to compare, because the human evaluation gave every theme a passing score. All of the themes except for one had a balance that was left-heavy (closer to or at 100%). This is understandable, given that the design of most two-column style blogs

have the more content-heavy column typically on the left side of the screen. Additionally, left-aligned websites are far more popular than right-aligned ones.

While passing scores for density were awarded to more dense themes by human evaluation, there is enough overlap that this difference is not statistically significant.

Given the number of variables present, it is surprising that this data presents little to contradict the idea that our automated heuristic evaluation is working as expected.

6 Conclusion and Future Work

User interface design, especially for the evolving web, is complex. One way that we can simplify that complexity is by making the design process easier. The first step in developing a future where design is automated is understanding what constitutes good design. In this project, we can see how design heuristics, that previous research has shown to indicate good design, is applied to the realm of two-column WordPress blog themes. A human evaluation provides a simple first step at validating these findings.

There are, of course, dozens more heuristics that can be evaluated and weights that can be developed to better score WordPress themes. Beyond the realm of two-column blogs, different heuristics can be applied to different types of websites. Further heuristics and further evaluation will yield better automated design and make he web more accessible.

References

[1] Theme Test Drive plugin. http://wordpress.org/plugins/theme-test-drive/. Accessed: 11/19/2014.

[2] Web Content Accessibility Guidelines (wcag) 1.0. http://www.w3.org/TR/WCAG10/. Accessed: 11/19/2014.

[3] Kunal Punera David Gibson and Andrew Tomkins. The volume and evolution of web page templates. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, WWW '05, pages 830–839, New York, NY, USA, 2005. ACM.

[4] Melody Y. Ivory and Rodrick Megraw. Evolution of web site design patterns. *ACM Trans. Inf. Syst*, 23(4):463–497, October 2005.

[5] Christoph Lindemann and Lars Littig. Coarse-grained classification of web sites by their structural properties. In *Proceedings of the 8th annual ACM international workshop on Web information and data management*, WIDM '06, pages 35–42, New York, NY, USA, 2006. ACM.

[6] Rashmi R. Sinha Melody Y. Ivory and Marti A. Hearst. Empirically validated web page design metrics. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '01, pages 35–60, New York, NY, USA, 2001. ACM.

[7] Mike Perkowitz and Oren Etzioni. Adaptive web sites: an ai challenge. In *Proceedings of the 15th international joint conference on Artificial intelligence - Volume 1*, IJCAI '97, pages 16–21, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.