Hunting for Headings: Sighted Labeling vs. Automatic Classification of Headings

Jeremy T. Brudvik, Jeffrey P. Bigham, Anna C. Cavender, and Richard E. Ladner
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98105 USA
{jtbrudvi, jbigham, cavender, ladner}@cs.washington.edu

ABSTRACT

Proper use of headings in web pages can make navigation more efficient for blind web users by indicating semantic divisions in the page. Unfortunately, many web pages do not use proper HTML markup (h1-h6 tags) to indicate headings, instead using visual styling to create headings, thus making the distinction between headings and other page text indistinguishable to blind users. In a user study in which sighted participants labeled headings on a set of web pages, participants did not often agree on which elements on the page should be labeled as headings, suggesting why headings are not used properly on the web today. To address this problem, we have created a system called *HeadingHunter* that predicts whether web page text semantically functions as a heading by examining visual features of the text as rendered in a web browser. Its performance in labeling headings compares favorably with both a manually-classified set of heading examples and the combined results of the sighted labelers in our study. The resulting system illustrates a general methodology of creating simple scripts operating over visual features that can be directly included in existing tools.

Categories and Subject Descriptors

K.4.2 [**Social Issues**]: Assistive technologies for persons with disabilities; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms

Design, Human Factors, Experimentation

Keywords

Blind Web Users, Web Accessibility, Heading Tags

1. INTRODUCTION

When users browse a web page, they try to accomplish a goal. They may be seeking the answer to a question, shopping for the latest bestseller, or socializing with friends. They are not browsing so that they can appreciate each detail of the web page. When blind web users browse the web, their screen readers present content in a linear format that can make it difficult to get past the details to the content of interest. When content is serialized into either aural speech or refreshable Braille, it can be time-consuming and frustrating for blind web users to locate the content that they really want to find because they have to wade through all the rest.

In addition to simply reading the screen, screen readers also provide useful shortcut keys that help web users skip through content. Most screen readers provide a shortcut to skip between headings. Defining heading tags in web pages is a primary way in which web developers can indicate to screen readers the logical divisions of content and how a user might effectively parse the web page in order to reach content of interest most directly. Figure 1 shows how searching through a page without headings can be much more cumbersome than searching using headings. Skipping directly from ‘community’ to ‘personals’ to ‘housing’ and down to ‘apts’ is much simpler (and more informative) than moving serially through each link before finally reaching ‘apts’.

Figure 1: In this simplified example from Craigslist, proper use of headings could reduce the number of steps required to find the “apartments” link from more then 40 to just 4.

Website	Category	Page chosen	URL
Google	Search engine	Search results for ‘internet’	http://www.google.com/search?q=internet
Yahoo	Web portal	Front page	http://www.yahoo.com/
MySpace	Social networking	Front page	http://www.myspace.com/
YouTube	Video	Front page	http://www.youtube.com/
eBay	Shopping	Front page	http://www.ebay.com/
Wikipedia	Reference	Front page in English	http://en.wikipedia.org/wiki/Main_Page
Craigslist	Classifieds	Front page for New York City	http://newyork.craigslist.org/
Blogger	Blog	Front page	https://www.blogger.com/start
Photobucket	Photo	Front page	http://www.photobucket.com/
IMDB	Entertainment	Front page	http://www.imdb.com/

Table 1: Websites used in evaluation of sighted labelers were based on popularity in the U.S., category, and then representative pages were chosen April 2008.

Unfortunately, web designers are not particularly reliable at incorporating logical heading tags into their web pages. Worse, our studies show that sighted labelers presented with relevant instructions from popular guidelines have difficulty agreeing on what constitutes a heading in the first place. In recognition of these facts, we have built a system called *HeadingHunter* that automatically identifies and inserts headings into web pages.

HeadingHunter first divides web pages into *text nodes*, individual, self-contained pieces of text or text leaves of the Document Object Model (DOM) tree [26], and then classifies each text node as either a heading or not. To validate the classifier, we compared HeadingHunter to headings consistently chosen by sighted participants.

HeadingHunter was constructed from a machine learning model over the visual characteristics of web pages. To construct this model, HeadingHunter uses as features both visual information about a text node, such as its font size, boldness, and color, and relational information that places each element within the context of other text on the page. After training with examples of labeled text nodes, the system is able to automatically identify headings and insert heading tags.

The method for generating the classifier used in HeadingHunter considered a large number of features, we have developed a general methodology for compiling a learned model into an efficient JavaScript script that can be incorporated into existing browsers today. This methodology was designed to be extensible beyond HeadingHunter and apply to other problems that may benefit from visual analysis, such as matching form elements with their labels, labeling the semantic structure of web pages, and labeling the row and column headings in HTML tables.

Our contributions are as follows:

- **Manual Labeling Evaluation** - an analysis of the difficulty experienced by non-expert labelers in assigning headings to web pages.

- **HeadingHunter Classifier** - a machine learning based method for automatically labeling page elements as headings that achieves performance comparable to that of sighted labelers.

- **HeadingHunter Script** - a script constructed using the HeadingHunter classifier that can insert headings into web pages and be integrated into existing tools.

- **General methodology** - a general methodology for creating machine learning classifiers over web page features and automatically converting those to Javascript scripts that can be easily integrated into existing tools and released to users.

2. RELATED WORK

Work related to HeadingHunter falls into four main categories: (i) work that has applied transcoding techniques in other settings, (ii) work relating to how heading tags are used and applied, (iii) tools that enable screen reader users to navigate content in semantically-meaningful ways, and (iv) techniques for deriving semantic knowledge using visual features.

Transcoding web content into a more accessible form has a rich history (See Asakawa *et al.* [2]). In SADIe, users associate a small number of example elements with a corresponding semantic description. The system uses the visual styling associated with each element as an address that can be used to find similar elements, which can be used as a low-cost way to specify transformations across an entire web site that will render it in a more accessible way [3]. With SADIe, these associations are provided manually; HeadingHunter uses visual characteristics in combination with other features.

Watanabe *et al.* explored the usability of heading elements [20]. They showed that both sighted and blind web users were able to more efficiently access web content when heading tags were used properly on structured web pages.

Existing screen readers, including JAWS [12] and aiBrowser [14], enable users to save placemarkers that enable users to quickly return to specific positions in the pages that they have viewed. To make use of these placemarkers, users must have already visited the page and assigned placemarkers to the page. Heading elements serve an analogous role for pages that blind users have not visited. HeadingHunter provides headings for pages that do not already have them and which users may not have already visited.

The Hearsay web browser visually segments web pages in order to present users with a semantically relevant VoiceXML tree that they can navigate [18]. Headings are simple in comparison, but can enable web users to quickly skip through web content in a semantically meaningful way. Importantly, their relative simplicity paired with their proven value means that users can benefit from headings that are automatically applied almost immediately, instead of waiting for specialized technology to be released.

Figure 2: Participants disagreed about which pieces of text where headings. Participant 5 circled many headings that Participant 1 did not, but the two raters did agree on a subset of headings.

The value of visual clues has been previously recognized in the area of web page segmentation [24]. Others have used visual features to assist in the extraction of information from visual lists and tables for the purposes of information extraction [8]. HeadingHunter uses similar clues to find text on the web page that is likely to signal the start of a new segment.

3. EVALUATION OF SIGHTED LABELERS

Properly labeling and using heading tags might be considered one of the easier accessibility features for web developers to implement because of its direct analog with the common task of outlining a document. Labeling heading tags is arguably not more difficult than providing alternative text for images, which web developers often overlook [5, 15]. A recent study of web browsing behavior found that only 53% of web pages visited by participants in the study contained at least one heading tag [4]. In this experiment, we seek to uncover how well sighted people label headings when directed to do so.

3.1 Experimental Setup

To investigate the ways in which sighted web browsers view headings (and to validate the results of our automated heading labeler) we asked 10 sighted participants (5 women, 5 men, ages 20-28) to manually label text that they considered headings for 10 web pages. The web pages were chosen for this part of the study to represent some of the most popular types of pages on the web today. Using Alexa [1], a web trafficking and statistic gathering service, we looked at the most popular websites visited from the United States as of April 2008. We included only one site from each of a set of general categories, filtering out pages with similar content (for example, MySpace and Facebook are both social networking sites, but MySpace is ranked higher in popularity, so we did not include Facebook). We also removed sites that

Figure 3: Participants were inconsistent in their own ratings. Participant 2 rated links from the search results of Google as headings, but did not label the search results of YouTube, which look very similar.

contained adult content and those sites that we were unable to correctly save due to a large amount of dynamic content. For each website, we chose one page that was most representative of the website as a whole rather than including only the initial page (for example, we chose the results page from a Google search for 'internet' and we chose the New York City version of Craigslist). Table 1 shows the websites, their categories, and pages within those sites that were chosen for the study.

3.2 Procedure

Participants first read a heading description that was created using excerpts from various World Wide Web Consortium (W3C) Guidelines [11, 19, 10, 23] related to both HTML document structure and accessibility, including WCAG [21]. All sentences except the first were taken from these guidelines. In order avoid biasing participants' selections, we crafted the description to contain no specific examples of headings, no direct references to HTML, and, most importantly, no details about visual characteristics sometimes associated with headings. The final definition used was as follows:

> Web pages are often organized into sections. A section is a self-contained portion of written content that deals with one or more related topics or thoughts. A section may consist of one or more paragraphs and include graphics, tables, lists, and sub-sections.
>
> A heading element briefly describes the topic of the section it introduces. Since some participants skim through a document by navigating its headings, it is important to use them appropriately

Page	Fleiss' Kappa	Number of nodes
Blogger	0.32	74
Craigslist	0.80	530
eBay	0.53	103
Google	0.49	206
IMDB	0.74	374
MySpace	0.49	298
Photobucket	0.33	124
Wikipedia	0.63	535
Yahoo	0.59	280
YouTube	0.43	375
All 10 Pages	0.58	2901

Table 2: Fleiss' Kappa statistics for the 10 pages rated by our participants. Some pages (such as Craigslist and IMDB) were easier for our participants to agree on headings, while others had more disagreement between raters (such as Blogger and Photobucket).

> to convey document structure. Heading information may be used, for example, to construct a table of contents for a document automatically.
>
> Since headings indicate the start of important sections of content, it is possible for participants with assistive technology to jump directly to the appropriate heading and begin reading the content. This significantly speeds interaction for participants who would otherwise access the content slowly.

After reading the description, participants were given paper versions of the web pages and were asked to label all pieces of text that may be headings by drawing a shape, such as a rectangle or oval, around the text.

To compare headings labeled by various participants, we used Fleiss' kappa statistics [7] for multiple raters by dividing the pages into 'labelable' nodes and treating each circle as a vote for that component. A labelable node was defined as any visible node in the Document Object Model (DOM) that directly contains some non-empty, non-whitespace text node. Fleiss' kappa statistic is a measure of agreement among voters. High Fleiss' kappa values among participants suggest that the headings are easier to identify than headings on a page with low Fleiss' kappa values among participants.

3.3 Results

Participants rated over 2900 total text nodes on the 10 pages and agreement for some pages seems to come more naturally than others. Figure 4 visually shows all labeled headings for four websites in this study and summarizes several patterns of agreement observed in the study. Each graph shows the text nodes rated by each participant for a given page. We initially hypothesized that Blogger and Photobucket would be difficult for participants to label due to a perceived lack of structure on these pages. This was reflected in the Fleiss' kappa calculations for these pages. On the other hand, Craigslist has a simple design and a strongly hierarchical layout, which yielded a more consistent labeling among participants.

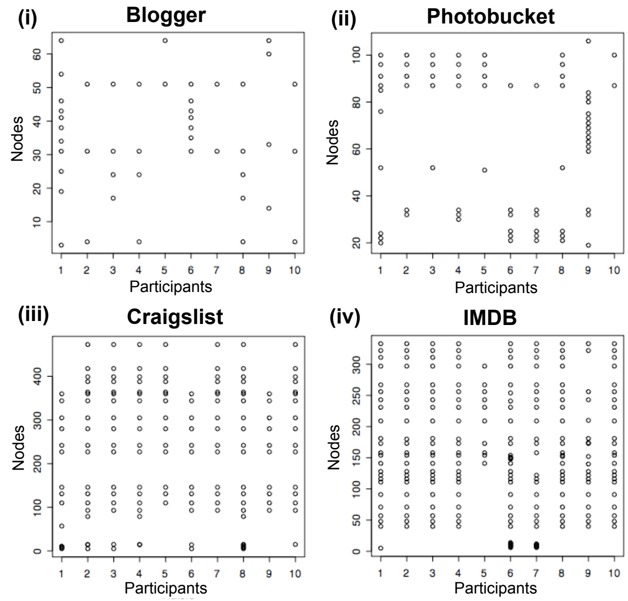

Figure 4: Headings provided by each participant for each text node indicating disagreement among participants. Participants rarely agreed on (i) Blogger and (ii) Photobucket, whereas participants often agreed on (iii) Craigslist and (iv) IMDB.

3.4 Discussion

This study illustrates the difficulty of adequately providing heading tags to web pages. Many people disagreed on which parts of pages should be considered headings (see Figure 2). And many people were inconsistent within their own labels both between and across web pages. For example, in Figure 3, Participant 2 chose links in the Google search results as headings whereas links in YouTube results (very similar in structure and appearance) were not chosen as headings. Perhaps the difficulty in deciding when and how to use heading tags is one reason why they are so infrequently used.

Individually, participants most often labeled true headings as headings (a low false-positive rate), but they did so incompletely (a high false-negative rate). Combining the results of multiple participants increases the completeness of the sample overall. For this reason, we are confident that labelable nodes with high agreements scores are very likely to be headings. These subsets of headings with different levels of high agreements scores were used as test sets to validate the results of HeadingHunter in Section 5. These results confirm those of Mankoff *et al.*, who found that the most complete web accessibility annotations came from a combination of efforts from multiple labelers [13].

4. HEADINGHUNTER

HeadingHunter is a classifier that determines which text nodes in a web page should be labeled as headings based on the visual appearance of text on a rendered web page.

4.1 Feature Selection

While official guidelines [11, 19, 10, 23] are careful to avoid defining headings by appearance, it is clearly often the case

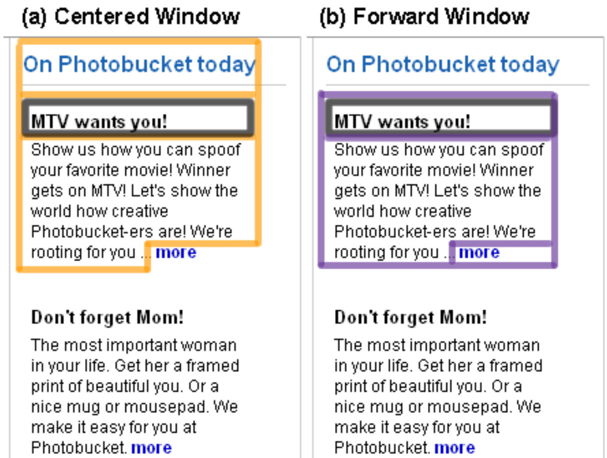

Figure 5: To determine whether a given text node is a heading, HeadingHunter compares the text node (highlighted in black) with the text around it using two different types of windows: (a) a "centered window" and (b) a "forward looking window". In this example, the window size is 3 text nodes.

that headings have different visual characteristics than other text elements. At a high level, newspaper pages are structured very similarly to web pages in the sense that they have section headings followed by sections of text. In newspapers, the headings are easy to find visually because they are generally much larger than the other text. A study by Zhang and Anderson [25] performed this task of identifying headings in scanned newspaper images. It is an oversimplification to compare web pages to newspapers since the web has a much wider range of styles, both across separate web pages and inside single pages, but this illustrates the visual distinction usually found with headings.

While there are potentially many telltale features associated with headings, we decided to keep the feature set fairly simple. We chose to examine four base features of every text node: font size, boldness, text length, and the width to height ratio. Other features were considered, but did not generalize because of the vast array of styles on the web page. For each node, we computed the z-score of each of these features with respect to multiple sizes of windows of text around the text node. We also looked at two types of windows: a "centered" window where the text node is the middle element, and a "forward" window where the text node comes before the rest of the text in the window (see Figure 5). We did not choose to use the "global" window, that is, the window with size equal to the number of text nodes in the page. Using this window generally caused HeadingHunter to perform poorly when testing on pages where headings in portions of the page, such as sidebars, may be smaller than non-heading text in the main content section of the page.

4.2 Training Data

The following sets of training data were collected for use in building our classifier: (A) 34 pages out of top 100 most-popular web sites [1] manually determined to be well-labeled, and (B) 36 web pages in a set of the top 50 pages returned from a Google query for "web accessibility" in December

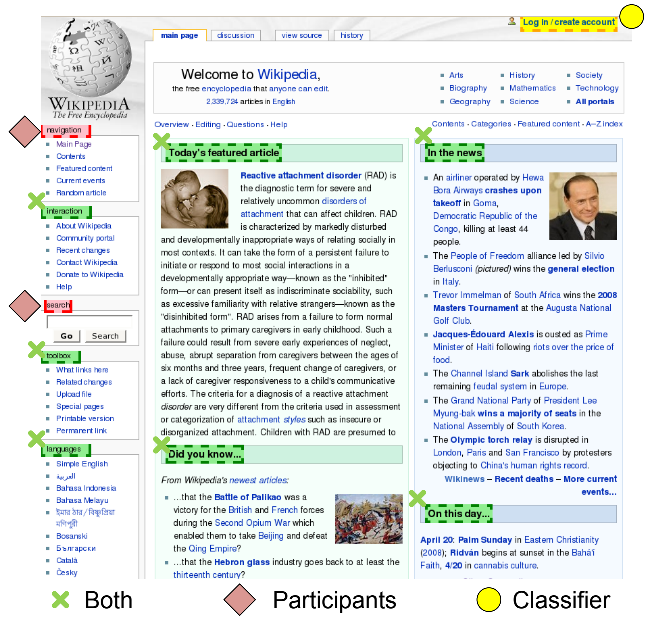

Figure 6: Example result of applying HeadingHunter to Wikipedia pages.

2007 manually determined to be well-labeled. We excluded from both sets the pages that were used in the evaluation of sighted labelers (Section 3) and the intersection of the two sets was empty. Determining whether a page was well-labeled was subjective, but none of the selected pages were altered for training or used for final testing.

To generate rendered pages and access information about visual characteristics, we created a Firefox browser extension that uses JavaScript to gather data from a list of pages. Firefox's implementation of the DOM allows straightforward access to many computed visual attributes through JavaScript, an important aspect of our classifier. For each text node, we gather information about all visual attributes rendered by Firefox and output these as potential features for the machine learning classifier.

4.3 Classifier

To analyze the data gathered, we used the Weka [22] implementation of the J48 decision tree algorithm [17]. Decision trees are a predictive model that, once trained on labeled instances of a class, can accurately classify new instances into different classes. In this case, instances could belong to either the "heading" or "non-heading" class. We chose to use decision trees as our model instead of another classifier because once trained, decision trees are easy to represent as a series of if/else conditions in JavaScript. Because web browsers natively handle JavaScript, this simplicity is important in ensuring that the resulting classifier can be released to end users.

To judge the performance of our classifier we considered the precision, recall, and f-measure. In our case, *precision* is defined as the fraction of text nodes that were classified as headings that were true headings, and *recall* is defined as the fraction of text nodes that were true headings that were accurately labeled as headings. The goal of HeadingHunter is to provide headings to web pages in order to

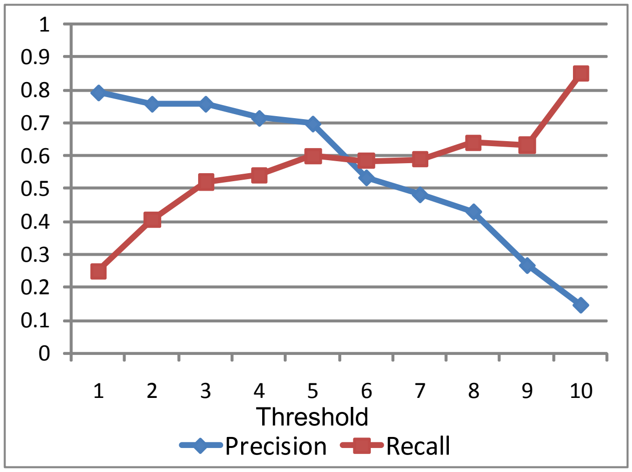

Figure 7: Precision vs. recall curves by threshold level indicating an optimal threshold of 5.

Page	Accuracy	Precision	Recall	f-measure
Blogger	0.97	0.5	1.0	0.67
Craigslist	0.98	0.75	0.53	0.62
eBay	0.98	1.0	0.6	0.75
Google	0.98	0.86	0.86	0.86
IMDB	0.96	0.75	0.52	0.62
MySpace	0.96	0.58	0.47	0.52
Photobucket	0.97	0.67	0.67	0.67
Wikipedia	0.99	0.79	0.85	0.81
Yahoo	0.95	0.8	0.24	0.36
YouTube	0.95	0.57	0.74	0.64

Table 3: Performance metrics at a threshold of 5 participants in agreement.

improve browsing efficiency. To facilitate that goal, the system should be both correct (precision) and complete (recall). The *f-measure* combines the two values using the following formula:

$$f = \frac{2 * recall * precision}{recall + precision} \quad (1)$$

The f-measure is used in the information retrieval community to balance both precision and recall in a single metric. The f-measure is similar to the geometric average of precision and recall. A f-measure near 1 means the classifier performs well in both precision or recall. A f-measure near 0 means either precision, recall, or both precision and recall are poor.

The final version of HeadingHunter trained on training data set A achieved a precision of 0.92 and recall of 0.81 when tested on data set B. We evaluate this classifier on the data labeled by user study participants in the next section.

4.4 Inserting Headings

The final step is to output the Weka-generated decision tree as a JavaScript function. This function is used as the principle component of a script that automatically adds headings to web pages when they are loaded. Web users can install the HeadingHunter script so that it is triggered when pages load using a method of their choosing, such as using Greasemonkey [9] or bookmarklets.

5. EVALUATION OF HEADINGHUNTER

We evaluated the performance of HeadingHunter using the human-labeled headings collected during the user study described in Section 3. Clearly, not all of the headings labeled by participants are likely to be good headings, especially in the cases where very few people agreed. Setting different thresholds on the number of positive labels a text node must receive before being considered a 'heading' enables us to create test sets of varying quality. For example, a threshold of 5 would yield a set of headings that contains only those headings labeled by 5 or more participants.

The observed precision and recall values of HeadingHunter over the human-labeled test set vary based on this threshold. As the threshold is increased, the test set rapidly shrinks, which causes the precision to fall and the recall to rise.

The maximum f-measure of HeadingHunter's classification occurred at a threshold of 5. (Figure 7) This threshold seems like a reasonable choice of threshold for the test data, because at least half of the people have agreed upon all of the headings in the dataset. This allows the noise of individual raters to be filtered, while still keeping many headings that were agreed upon by many.

The overall precision of HeadingHunter at threshold 5 was 0.70 and recall was 0.60, resulting in an f-measure of 0.65. The average precision, recall, and f-measure for study participants with respect to a threshold of 5 were 0.76, 0.76, and 0.75, respectively. Clearly, this second set of numbers are unfairly biased toward the participants because their own labels may be a part of the "correct labels" in the threshold of 5, but they still serve as a useful reference point. From these results we can determine that the performance of HeadingHunter was quite similar to that of the sighted labelers.

At the threshold of 5, HeadingHunter also performed much better on some pages than others, as shown in table 3. Wikipedia and Google had f-measures of 0.81 and 0.86, respectively, with nearly balanced precision and recall values. Yahoo had a low f-measure of 0.36, which was caused by a low recall of 0.24. HeadingHunter was unable to learn a consistent model of the page because of great variety in the style of the headings labeled by participants. While accuracies are high, it is not a very useful metric because it disproportionately weights the effect of true-negatives (i.e., non-heading elements that are accurately labeled as not being headings). This is because most elements on a page are in fact not headings. In a sense, precision and recall only give credit for "correct" answers, that is, they increase only if the classifier classifies things that should be headings correctly.

Another measure of performance is the recall of important headings. If we make the assumption that importance of a heading is strongly correlated with the percentage of users that chose it as a heading, this yields 10 importance levels. We would hope that a good classifier has better recall on headings that were more agreed upon by participants than those that were only indicated by few participants. Table 4 shows that there is a significant correlation between importance level and recall ($R = .828$, $p = .003$).

Importance Level	Recall	Classifier-Labeled / Total
10	0.85	17 / 20
9	0.48	14 / 29
8	0.66	19 / 29
7	0.35	6 / 17
6	0.55	6 / 11
5	0.66	19 / 29
4	0.11	2 / 18
3	0.31	5 / 16
2	0.00	0 / 48
1	0.03	4 / 151

Table 4: HeadingHunter tends to perform better on headings with higher importance levels (i.e. headings with more agreement between participants).

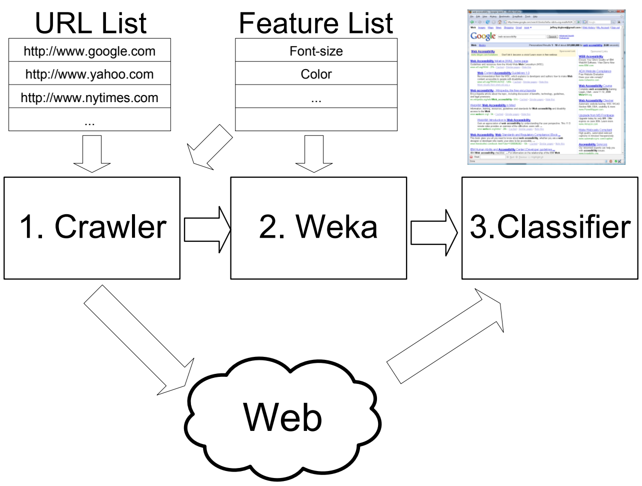

Figure 8: The 3 steps of our general methodology for turning rendered visual features of web pages into a classifier that can be used to increase accessibility.

6. GENERAL METHODOLOGY

The methodology used to create HeadingHunter can be generalized to support other classifiers operating over the visual features of rendered web pages in order to automatically supply other markup targeted at improving accessibility. These classifiers, once compiled to Javascript, can be used immediately as part of web browsers and, therefore, quickly reach end users.

We have created a set of tools that can be used to support this approach. These tools include (i) a web crawler that operates over visual pages, (ii) tools for extracting visual features from web pages elements and exporting them into Weka-compatible files, and (iii) a tool for converting Weka-generated decision trees to JavaScript code that can be used in scripts to make changes in the web pages that users view. (Figure 8)

7. FUTURE WORK

We see the following opportunities for future work:

Implications for Accessibility Guidelines. In our study of manual heading labeling we found that it was difficult for participants to correctly and completely label headings even when provided with a definition adapted from popular standards. Although we did not target a single standard or guideline, our study indicates that those who are responsible for creating and updating accessibility standards and guidelines should pay close attention to writing them in a way that encourages web developers to implement them accurately and completely.

Analysis of End User Experience. The motivation of this work came both from our discussions with blind web users who mentioned headings as being one of the most powerful ways to increase browsing efficiency and from the work by Watanabe *et al.* that showed that headings can improve usability [20]. Both these comments and Watanabe's study referenced web pages with a clear hierarchical structure. The study investigating the ability of sighted labelers to provide headings presented here made clear that all web pages are not created equally - some have a clear hierarchy of information, while the content on others are difficult to conform into a hierarchy. A future direction is to investigate these differences.

Integration into Existing Scripting Frameworks. We plan to integrate HeadingHunter into our existing Accessmonkey Framework [6] as a *general script* that can apply to any web site that users visit.

Extending HeadingHunter. Currently, HeadingHunter does not distinguish between different heading levels. In future work, we may extend the classifier to learn models of different heading levels. HeadingHunter may also benefit from site-specific classifiers targeting the generation of headings for a particular web site.

Applications to Related Problems. We believe that the general methodology presented in this paper for applying machine learning to accessibility problems in a way that can be released quickly and easily as JavaScript scripts is a promising method for addressing additional problems. Examples of such problems include finding column and row labels in tables, automatically discovering labels for form elements, and identifying logical sections of web pages.

8. CONCLUSION

We have presented an evaluation of both automatic and manual heading labeling for web pages. Results of the manual labeling indicate that even informed sighted labelers often do not agree on what should be a heading and do not label web pages completely. Our HeadingHunter system automatically inserts headings into web pages and has comparable performance to the combined efforts of the participants in our study. The approach used here capitalized on the visual features of a web page to create a machine learning classifier that then resulted in a script that can be integrated into existing technology to improve accessibility in general.

9. ACKNOWLEDGMENTS

This work has been supported by National Science Foundation Grant IIS-0415273 and a Boeing Professorship. We thank Zach Lattin and Lindsay Yazzolino for their comments that inspired this work. Finally, we thank our study participants for their invaluable feedback.

10. REFERENCES

[1] Alexa web search – data services, 2006. http://www.alexa.com.

[2] C. Asakawa and H. Takagi. *Web Accessibility: A Foundation for Research*, chapter Transcoding. Springer, 2008.

[3] S. Bechhofer, S. Harper, and D. Lunn. Sadie: Semantic annotation for accessibility. In *Proceedings of 5th International Semantic Web Conference*, 2006.

[4] J. Bigham, A. C. Cavender, J. T. Brudvik, J. O. Wobbrock, and R. Ladner. Webinsitu: A comparative analysis of blind and sighted browsing behavior. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '07)*.

[5] J. Bigham, R. S. Kaminsky, R. Ladner, O. M. Danielsson, and G. L. Hempton. Webinsight: Making web images accessible. In *Proceedings of 8th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '06)*, 2006.

[6] J. Bigham and R. Ladner. Accessmonkey: A collaborative scripting framework for web users and developers. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility (W4A '07)*, pages 25–34, 2007.

[7] J. L. Fleiss. Measuring nominal scale agreement among many raters. In *Psychological Bulletin*, Vol. 76, No. 5 pp. 378–382

[8] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Kroepl, and B. Pollak. Towards domain-independent information extraction from web tables. In *Proceedings of the 16th Internationa Conference on the World Wide Web (WWW '07)*, pages 71–80, Banff, Alberta, Canada, 2007.

[9] Greasemonkey firefox extension. http://greasemonkey.mozdev.org/.

[10] H69: Providing heading elements at the beginning of each section of content — techniques for wcag 2.0. World Wide Web Consortium, 2008. http://www.w3.org/WAI/GL/WCAG20/WD-WCAG20-TECHS/H69.html.

[11] Html techniques for web content accessibility guidelines 1.0. World Wide Web Consortium, November 2000. http://www.w3.org/TR/WCAG10-HTML-TECHS/.

[12] JAWS 8.0 for windows. Freedom Scientific, 2006. http://www.freedomscientific.com.

[13] J. Mankoff, H. Fait, and T. Tran. Is your web page accessible?: a comparative study of methods for assessing web page accessibility for the blind. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '05)*, pages 41–50, New York, NY, USA, 2005.

[14] H. Miyashita, D. Sato, H. Takagi, and C. Asakawa. Aibrowser for multimedia: introducing multimedia content accessibility for visually impaired users. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '07)*, pages 91–98, New York, NY, USA, 2007. ACM.

[15] H. Petrie, C. Harrison, and S. Dev. Describing images on the web: a survey of current practice and prospects for the future. In *Proceedings of Human Computer Interaction International (HCII '05)*, July 2005.

[16] M. Pilgrim, editor. *Greasemonkey Hacks: Tips & Tools for Remixing the Web with Firefox*. O'Reilly Media, 2005.

[17] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106.

[18] I. Ramakrishnan, A. Stent, and G. Yang. Hearsay: Enabling audio browsing on hypertext content. In *Proceedings of the 13th International Conference on the World Wide Web (WWW '04)*, 2004.

[19] Understanding WCAG 2.0. World Wide Web Consortium, 2007. http://www.w3.org/TR/2007/WD-UNDERSTANDING-WCAG20-20071211/navigation-mechanisms-headings.html.

[20] T. Watanabe. Experimental evaluation of usability and accessibility of heading elements. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility (W4A '07)*, pages 157 – 164, 2007.

[21] Web content accessibility guidelines 2.0 (WCAG 2.0). World Wide Web Consortium, 2006. http://www.w3.org/TR/WCAG20/.

[22] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[23] World Wide Web Consortium. *The global structure of an HTML document*, 2008. http://www.w3.org/TR/html401/struct/global.html.

[24] Y. Yang and H. Zhang. Html page analysis based on visual cues. In *Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR '01)*, pages 859–864, 2001.

[25] W. Zhang and A. T.L. Using artificial neural networks to identify headings in newspaper documents. In *Proceedings of the International Joint Conference on Neural Networks, 2003*, volume 3, pages 2283– 2287. IEEE, 20-24 July 2003.

[26] Document Object Model (DOM) Level 3 Core Specification. http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/.