Extracting Context To Improve Accuracy For HTML Content Extraction

Suhit Gupta
Columbia University
500 W. 120th Street
New York, NY 10027
001-212-939-7184
suhit@cs.columbia.edu

Gail Kaiser
Columbia University
500 W. 120th Street
New York, NY 10027
001-212-939-7081
kaiser@cs.columbia.edu

Salvatore Stolfo
Columbia University
500 W. 120th Street
New York, NY 10027
001-212-939-7080
sal@cs.columbia.edu

ABSTRACT

Previous work on content extraction utilized various heuristics such as link to text ratio, prominence of tables, and identification of advertising. Many of these heuristics were associated with "settings", whereby some heuristics could be turned on or off and others parameterized by minimum or maximum threshold values. A given collection of settings – such as removing table cells with high linked to non-linked text ratios and removing all apparent advertising – might work very well for a news website, but leave little or no content left for the reader of a shopping site or a web portal We present a new technique, based on incrementally clustering websites using search engine snippets, to associate a newly requested website with a particular "genre", and then employ settings previously determined to be appropriate for that genre, with dramatically improved content extraction results overall.

Categories and Subject Descriptors

I.7.4 [**Document and Text Processing**]: Electronic Publishing; H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *Web-based Services*

General Terms

Human Factors, Algorithms, Standardization.

Keywords

DOM trees, content extraction, reformatting, HTML, context, accessibility, speech rendering.

1. INTRODUCTION

Users are spending more and more time on the Internet in today's world of online shopping and banking; meanwhile, webpages are getting more complex in design and content. Web pages are cluttered with guides and menus attempting to improve the user's efficiency, but they often end up distracting from the actual content of interest. These "features" may include script- and flash-driven animation, menus, pop-up ads, obtrusive banner advertisements, unnecessary images, or links scattered around the screen.

The automatic extraction of useful and relevant content from web pages has many applications, including enabling end users to access the web more easily over constrained devices like PDAs and cellular phones. Content extraction is particularly useful for the visually impaired and the blind. A common practice for improving webpage accessibility is to increase font size and decrease screen resolution; however, this also increases the size of clutter, reducing efficiency. Screen readers for the blind, like Hal Screen Reader, Microsoft's Narrator or IBM Homepage Reader, generally don't remove such clutter either and often read out raw HTML. Natural Language Processing (NLP) and information retrieval (IR) algorithms can also benefit from content extraction, as they rely on the relevance of content and the reduction of "standard word error rate" to produce accurate results [1]. Content extraction allows such algorithms to process only the extracted content, instead of either using cluttered data from the web, or writing specialized extractors for each web domain [2][3].

In our preliminary approach to content extraction, reported in [4], each web page is first passed through a conventional HTML parser, which corrects the markup and creates a Document Object Model tree. This enables the algorithm to extract information from large logical units as well as easily prune smaller units such as specific links from the tree, which is thus much easier to work with than the HTML text.

Our web proxy, Crunch, then applies a series of heuristics to the DOM tree, essentially tree pattern-matching rules, that determine what to retain and what to remove (or, in some cases, move elsewhere in the tree). The result is the "extracted" content, which can then be rendered by any HTML browser or as plain text. Crunch's set of heuristics can be customized by an administrator or a user, to retain more or less, and additional heuristics can be introduced by implementing plugins that conform to a standard interface [5].

Crunch's default settings for its heuristics show nice results for many web pages, but do not work so well for others. Manual tweaking of the heuristic settings can produce excellent results for nearly any web page, but is impractical for constrained-screen, web accessibility and many NLP/IR applications.

This reflects an inherent limitation of any context extraction technology: it is impossible to determine the intention of the author and the desires of the reader. However, our experiments with manually adjusting the thresholds at which some heuristics are applied, and turning other heuristics off or on, have shown that websites with similar content, or from the same industrial segment, tend to require the same settings to produce, what seems to us subjectively, the best results.

Thus, if we could automatically and efficiently characterize new websites as belonging to a "genre" for which preferred settings are already known - stored in a database available to the Crunch proxy - we can automate the tweaking

WWW 2005, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

process by simply reusing those settings. Manual intervention, or trial and error generation of heuristic settings, is needed only when "new" genres are identified. The likelihood of a user happening to browse a website insufficiently close to any known genre should decrease over time as the database grows.

2. APPROACH

Our approach to classifying websites based on genre involves a one-time initial processing stage where we cluster and pre-classify a variety of websites. We combine the textual content of the site itself with the search result snippets returned by searching for the site's domain name on three of the top Internet search engines (Google, Yahoo and Dogpile), a model first suggested by [6] but there employed only for organizing search engine results. Combining search engine results for each domain with the original text also improves the frequency of the occurrence of words that describe the function and content of the site. After removing all stop words, we pick all unique words that occur in at least one of the graphs more than five times, creating a master set of *Key Words*. A re-graph against this new set of Key Words produces a content genre *identifier* for each of these websites. We can now use our classifications in conjunction with a database containing preferred (manually tweaked) heuristic settings for each of the pre-processed sites.

When a user encounters a website not already included in the database, we use the text from the new site and corresponding search engine snippets, to perform a frequency match against the frequently-occurring Key Words. We then use the *Manhattan histogram distance measure algorithm* to measure the distance between the website in question and previous classifications. The formula is defined as

$$D_1(h_1, h_2) = \sum_{i=0}^{n-1} | h_1[i] - h_2[i] |$$

The histogram (h_1, h_2) is represented as a vector, where n is the number of bins in the histogram (i.e., the number of words in our Key Word Set). We use the settings associated with the website whose distance is closest to the one being accessed. More precisely, we use the settings of the website, among those with known preferred settings, that is closest – since the truly closest may be another website added since the initial processing stage. Note, however, that preferred settings can always be added for these new websites later on, without burdening the user while browsing.

3. EXPERIMENTAL RESULTS

For our experiments, we chose 200 sites often visited by members of our group. Using these sites as our corpus, we worked on clustering them as specified in the above technique. We found that there is existed natural clustering in websites. We also found that most sites of similar genre followed similar layouts as they shared the same content and consequently the same target audience. We were able to use these facts towards loading filter settings that worked well for one site in a particular genre, on all the sites of that genre, to get excellent results towards content extraction. An example of our results can be seen in Figure 1 where the original site is shown in part a; part b shows bad extraction results when Crunch settings specific to shopping sites are loaded. This is finally corrected and the best version is shown in part c when the CNN page is recognized correctly as a news site by doing genre analysis.

To conclude, we have started to work towards dynamically detecting the context of the website, in terms of its content genre. Using this information and comparing this to previously known results that work well for certain genres of sites, we are able to select settings for our heuristics and achieve the same results automatically that previously required a human administrator.

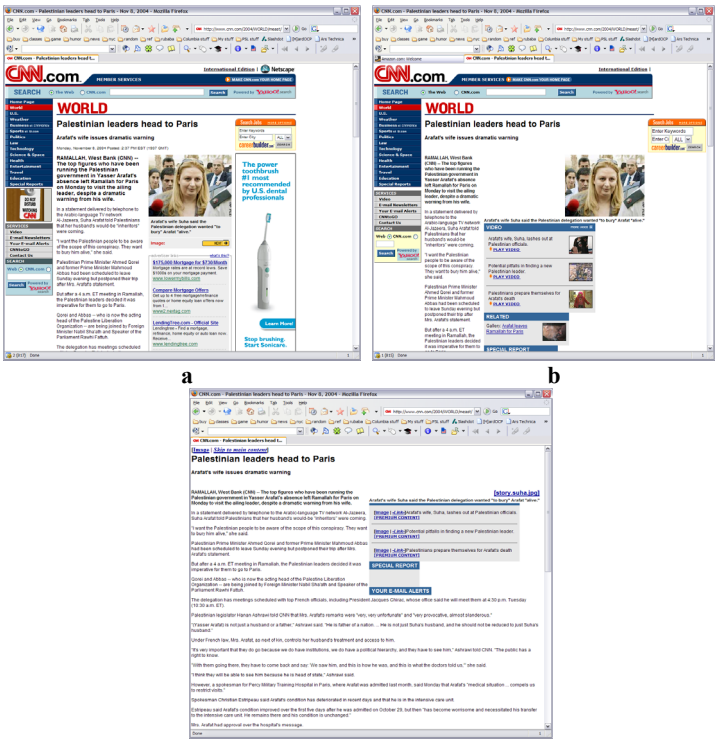

Figure 1 - CNN (a. original; b. extraction with shopping genre settings; c. with correct news genre settings)

4. ACKNOWLEDGEMENTS

The Programming Systems Laboratory is funded in part by National Science Foundation grants CNS-0426623, CCR-0203876 and EIA-0202063. Part of the work reported in this paper emanated from research from the Columbia IDS lab, which has been supported by grants form NSF and HS ARPA.

5. REFERENCES

[1] Wolfgang Reichl, Bob Carpenter, Jennifer Chu-Carroll, Wu Chou, "Language Modeling for Content Extraction in Human-Computer Dialogues", In International Conference on Spoken Language Processing (ICSLP) 1998

[2] Ion Muslea, Steve Minton and Craig Knoblock, "A Hierarchal Approach to Wrapper Induction", In Proc. of 3rd Int. Conf. on Autonomous Agents (Agents'99), 1999

[3] Min-Yen Kan, Judith Klavans, Kathleen McKeown, "Linear Segmentation and Segment Relevance", In Proc. of 6th Int. Workshop of Very Large Corpora (WVLC-6), 1998

[4] Suhit Gupta, Gail Kaiser, David Neistadt, Peter Grimm, "DOM-based Content Extraction of HTML Documents", 12th International World Wide Web Conference, May 2003

[5] Suhit Gupta; Gail E Kaiser, Peter Grimm, Michael F Chiang, Justin Starren, "Automating Content Extraction of HTML Documents", World Wide Web Journal, In Press

[6] Oren Zamir, Oren Etzioni, "Web Document Clustering: A Feasibility Demonstration", SIGIR, 1998