Separating XHTML Content From Navigation Clutter Using DOM-Structure Block Analysis

Constantine Mantratzis
Department of Computing
Macquarie University
E6A Room 256 – NSW 2109
Australia
+61 433 1546 88
cmantrat@ics.mq.edu.au

Mehmet Orgun
Department of Computing
Macquarie University
E6A Room 314 – NSW 2109
Australia
+61 2 9850 9570
mehmet@ics.mq.edu.au

Steve Cassidy
Department of Computing
Macquarie University
E6A Room 381 – NSW 2109
Australia
+61 2 9850 9581
cassidy@ics.mq.edu.au

ABSTRACT

This short paper gives an overview of the principles behind an algorithm that separates the core-content of a web document from hyperlinked-clutter such as text advertisements and long links of syndicated references to other resources.

Its advantage over other approaches is its ability to identify both loosely as well as tightly defined "table-like" or "list-like" structures of hyperlinks (from nested tables to simple, bullet-pointed lists) by operating at various levels within the DOM tree.

The resulting data can then be used to extract the core-content from a web document for semantic analysis or other information retrieval purposes as well as to aid in the process of "clipping" a web document to its bare essentials for use with hardware-limited devices such as PDAs and cell phones.

Categories and Subject Descriptors

I.7.4 [**Document and Text Processing**]: Electronic publishing

H.5.4 [**Hypertext/Hypermedia**]: Architectures, Navigation, User issues

General Terms

Algorithms, Management, Standardization, Design

Keywords

HTML document, XHTML document, DOM tree, information retrieval, content extraction, hyperlink lists

1. INTRODUCTION

Thanks to an ever-increasing ability of machines to produce a better semantic understanding of a web document's context, custom-made, content-relevant online advertisements are blended with a web document's main body, introducing only further clutter to surround the "core" content of a page.

HT'05, September 6–9, 2005, Salzburg, Austria.

Their subtle placement among the "core" content – both contextually as well as structurally - has began to "confuse" search engines who take such clutter into account during document crawls. False search results begin to appear as high as on the first or second page of a search engine's result set with references to irrelevant web pages that are either entirely riddled with text adverts and syndicated links or contain an advertisement section that simply happens to match the query's keywords.

A Google search on a recent popular topic of "drinking and voting", an article that was originally posted on the website AdJab[1] for example, listed an irrelevant web page from the technology website EnGadget about Nokia cell phones[2] as its 11th result.

This problem, also known as "intra-page redundancy" as described in Shian-Hua Lin and Jan-Ming Ho [1] will only be exacerbated by the increasing use of syndicated links in blogs and their web aggregators. The ability to separate content from link clutter, be it advertisements or syndicated links will not only improve the indexing accuracy of content-focused information retrieval services but will also increase the efficiency of "web-clipping" applications that work by discarding "less important" features of web documents with the aim of shrink-wrapping their core content for use on PDAs, cell phones and other such devices with limited display-area and memory specifications.

We begin in section 2 by looking at similar approaches and identifying their shortcomings.

Section 3 describes the principles behind our algorithm and analyses some of the key parts behind our Java implementation.

Section 4 proposes a couple of improvements to increase the efficiency of our algorithm.

2. RELATED WORK

The list of projects with similar objectives is not short.

Xie et al's study [2], narrows the focus on the benefits of block importance approaches for mobile devices and another research

[1] "*Drinking and Voting are Perfect Together*", AdJab, 13 May 2005, http://www.adjab.com/2005/05/13/drinking-and-voting-are-perfect-together/

[2] Peter Rojas, "*Nokia and the NBA: together forever*", EnGadget - Cellphones, 4 May 2004, http://cellphones.engadget.com/entry/2241572046171126/

looks into page layout analysis for the purposes of image searching within web documents (He et al [3]).

2.1 Large text body equals great importance

Some of the approaches such as those of Finn, Kushmerick and Smyth [4] and McKeown et al [5] assume that the document is formed by a simple structure (usually a table) with the core content occupying the largest portion of it. As also mentioned by Gupta et al [6] however, such approaches produce noisy results, especially when the analysed document follows a more modular structure adopted by most blogs, online forums or news sites today. Such sites tend to present the core content followed by repeated chunks of external references, advertisements etc., all of which invalidate the assumption of a single large content body, on which such approaches are based.

2.2 Other hybrid approaches

Gupta et al [6] extends the above techniques by suggesting the use of a range of "filters" with specific tasks that involve the phased elimination of common irrelevant elements in a web document such as long lists of hyperlinks, advertisements and empty tables. This approach removes "clutter" by relying on a public "black-list" of advertisement servers in order to identify candidate hyperlinks, a method unable to detect anchors to ad servers that are not listed or direct links to websites that serve their own adverts. Additionally, the approach in [6] seems to offer a limited view of what constitutes a list of links and the criteria for their quantitative content analysis by limiting them to table cells with the entire list residing inside one such cell.

Due to each individual web document's style requirements, it is very likely that lists of hyperlinks are constructed in a much more complicated manner involving perhaps groups of tables within table cells, structures and sub-structures, all of which eventually form a list of hyperlinks and their headings. The approach followed in Gupta et al [6] would struggle to identify such lists by examining individual table cells since the arrangement only becomes a hyperlink list once examined from a much higher, collective level. Lists may also be constructed in an even less "traditional" manner, by using `<DIV>` tags or even a simple sequence of anchor tags separated by symbols such as the pipe (|).

Perhaps the greatest issue with the solutions discussed above is that they expect the end user to make many of the decisions that will affect the granularity of the original document's clipping or the detail of the resulting summarisation.

While such techniques work well for situations where humans apply them interactively (e.g. on a PDA or cell phone), they tend to produce noisy results when left to operate alone as part of a machine-driven information retrieval procedure (such as on search engines and document indexers).

3. DESCRIPTION OF THE ALGORITHM

Taking the above into account, our aim is to introduce an algorithm that can separate true content in a web document from cluttered lists of hyperlinks without making too many strict assumptions about what constitutes a "link list" or having to rely on external information (such as "black-lists" of hosts) in order to exclude redundant data (i.e. advertisement hyperlinks). Additionally, our algorithm should be able to work unattended as part of a machine-driven analysis as well as to allow direct or indirect customisation of its operation by end users.

We split our algorithm's operations down to the following key stages.

At first, the retrieved web document is checked to ensure that its DOM is error-free. Once the document has been cleaned up, we open it using JDOM [7] in order to create a Java-accessible object representation of the retrieved document's object model.

We begin from the DOM's root and recursively traverse its children and their descendants, noting the following information about each element we encounter:

- ***Element*** A document consists of one or more (X)HTML elements. We store each element's name
- ***Parent*** Each element has at most one parent element. For each element we encounter, we record a reference to its parent.
- ***Structure elements*** A pre-defined list of elements that are considered "structurally important" such as `<table>`, `<tr>`, `<td>` or `<div>`

The next stage "relaxes" the DOM tree by "flattening" the hierarchy of all "non-structurally important" document elements (such as `<span>`, `<strong>` or `<a>`) against the ones that contribute to the creation of table-like structures, i.e. a `<table>`, `<tr>`, `<td>`, `<ul>` or `<div>`.

Once all such "structurally-important" elements of our DOM have been identified, we establish which ones classify as potential roots of a sub-structure (i.e. a child structure such as a group of rows, paragraphs, list elements etc. within the document).

We are now ready to take each one of these structures and their children in turn and analyse their content for hyperlinks. Our aim here is to determine how many anchors exist among other tags within a structure and how much of the content within each structure is hyperlinked. This is done by taking each "structurally-important" element of the DOM in turn and recursively visiting its children elements. Therefore, for each parent element that is marked as "structurally-important", we recursively record the following:

- Quantity of that parent element's children that are anchor tags and contain at least one character of content.
- Quantity of all of that parent element's "structurally insignificant" children tags (i.e. the total number of tags under a parent tag) that contain at least one character of content.
- Tags that contain no text are ignored
- Length of the string within each of that parent element's children that are anchor tags.
- Length of the string within all of that parent element's "structurally insignificant" children tags.

Having collected hyperlink data for each parent tag's first-level descendants, we are now ready to traverse the tree from the

bottom up and update recursively the hyperlink values for each parent-section of each sub-section. This step allows us to recursively "pull out" of a structure and progressively examine it within the context of its larger, parent structure. It gives us the flexibility of examining complex lists of hyperlinks that are not constructed in a straightforward, single-table manner but instead employ a collection of sub-structures, each of which on its own reveals no list characteristics but when looked at collectively, can form such a candidate structure. In order however for our algorithm to award points of relevance to each structure fairly, it needs to take into account that the further we pull out of a sub-structure, the less important the hyperlink data from that structure become. In other words, the outer-most table of an enclosed table structure, will adopt the inner table structure's hyperlink quantity data but will award them less significance when calculating the overall ratio of hyperlinked to simple content. At this stage, our algorithm traverses the DOM from the bottom up. Child elements with a common parent have their anchor-tag and total tag quantities as well as their hyperlinked and total content length values summed up and added to the respective values of their parent tag. This recursive procedure continues for all elements and all of their parents until the top of the DOM tree is reached for all recursions. Before each addition of the children sums to their parents' values, each number is decreased by a percentage that is analogous to the distance between the child structure and the currently visited parent.

Once all hyperlink data of every parent and its parent elements have been updated using the bottom-up traversal described earlier, we sweep the DOM one last time and produce a result about each parent element's likelihood of being a root to a hyperlink list. We do this by calculating the ratio between non-empty anchor tags over the overall number of tags for each section with a threshold of 0.5 as well as the ratio of the length of hyperlinked text over the total length of text for each section with a threshold of 0.4.

These two evaluations add each, one point to a section root element's total "hyperlink score". If a section receives points from the evaluation of both criteria, then there is a high probability that this section and its children are part of a list of hyperlinks rather than "core" content. If the root of a section receives one point instead, then it is up to the system that implements this algorithm to decide what weight to apply to this decision and act accordingly.

4. PROPOSED OPTIMISATIONS

We have described a process by which our algorithm progressively identified areas of high hyperlink concentration within a web document and its attempt to separate them from the "core content". We did so by examining structures that form lists of hyperlinks at different levels of the DOM and assigning scores to them according to each section's ratio of hyperlinked to "ordinary" content.

The algorithm performed well in situations where pages contained one or more bodies of text along with hyperlink clutter and was able to correctly identify lists of hyperlinks that were constructed using tables (simple or nested), `<DIV>` tags, simple sequences of anchor tags separated by symbols such as the pipe (|) character or bullet-pointed lists. The proposed solution however becomes less efficient when used in front pages of portals or other such web documents where an entire page is filled with short summarisations (describing a section) and their hyperlinks without giving prominence to any one body of content.

To optimise the efficiency of our approach, the following main issues need to be improved:

- The fixed threshold for the two ratios mentioned earlier could be replaced by a dynamically-adjustable ratio that is continuously refined for each content block based on semantic an structural comparisons with other documents from the same domain
- Support for lists of hyperlinks that use images instead of text by comparing the ratio of hyperlinked image tags to the ratio of those in a similar section in another document of the same domain. The analysis can be further enhanced by comparing the dimensions of the images on both sides to further reduce the noise of the result.

5. REFERENCES

[1] Shian-Hua Lin and Jan-Ming Ho, "*Discovering Informative Content Blocks from Web Documents*", Eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 2002

[2] Xing Xie, Gengxin Miao, Ruihua Song, Ji-Rong Wen, Wei-Ying Ma. "*Efficient Browsing of Web Search Results on Mobile Devices Based on Block Importance Model*", Percom, vol. 00, no. , pp. 17-26, Third 2005

[3] X. He, D. Cai, J.-R. Wen, W.-Y. Ma and H.-J. Zhang, "*ImageSeer: Clustering and Searching WWW Images Using Link and Page Layout Analysis*", Microsoft Technical Report, MSR-TR-2004-38, 2004

[4] Aidan Finn, Nicholas Kushmerick and Barry Smyth, "*Fact or fiction: Content classification for digital libraries*", Joint DELOS-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries (Dublin), 2001

[5] K.R. McKeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, M.Y. Kan, B. Schiffman and S. Teufel, "*Columbia Multidocument Summarization: Approach and Evaluation*", Document Understanding Conference, 2001

[6] S. Gupta, G. Kaiser, D. Neistadt, P. Grimm, "*DOM-based Content Extraction of HTML Documents*", WWW2003, 20 May 2003

[7] JDOM, version 1.0, 2005, http://www.jdom.org/