

Extracting Content Blocks from Web Pages

Manisha Marathe¹, Dr. S.H.Patil², G.V.Garje², M.S.Bewoor²

¹ PVG's College of Engineering & Technology, Pune, India

E-mail: manisha_marathe@yahoo.com

² Bharati Vidyapeeth College of Engineering, Pune, India

² PVG's College of Engineering, Pune, India

² Bharati Vidyapeeth College of Engineering, Pune, India

E-mail: shpatil@bvucoep.edu.in, garjegov@yahoo.com, msbewoor@bvucoep.edu.in

Abstract - User search for the required information using search engines. Search engines crawl and index web pages according to their informative content. User is interested only in the informative contents and not in non-informative content blocks. Web pages often contain navigation sidebars, advertisements, search blocks, copyright notices, etc which are not content blocks. The information contained in these non-content blocks can harm web mining. So it is important to separate the informative primary content blocks from non-informative blocks. In this paper three different algorithms for separating content blocks from non-content blocks developed by different authors are discussed. Removing non-informative content blocks from web pages can achieve significant storage and timing saving.

Index Terms- Primary content, Entropy, noisy blocks, Web mining, Web blocks

I. INTRODUCTION

Currently, the World Wide Web is the largest source of information. Huge amount of data is present on the Web. The innovation of the web creates numerous information sources published as HTML pages on the Internet. Search engines crawl the World Wide Web to collect web pages. These pages are then stored and indexed.

The user who is performing a search using search engine is interested in primary informative content of the web page. Because of the large amount of information available on WWW it is very difficult to identify useful information. A large part of these web pages is content that can not be classified as the primary informative content of the web page. Web pages often contain advertisements, image-maps, logos, search boxes, navigational links, related links, footers and headers, and copyright information along with the primary content. These blocks are not relevant to the main content of the page. These items are required for web site owners but they will hamper the web data mining and decrease performance of the search engines. Such blocks are referred to as non-content blocks. These blocks are very common in web pages.

Fig 1. shows a web page from BBC's website. In this page the text block having the news is the primary content block. Different blocks in the page are as shown in fig 1. Primary content block is shown by dotted line and other blocks are shown by solid line boxes.

The advantage of identifying non-content blocks from web pages is that if user does not want non-content

blocks these can be deleted. These non-content blocks are normally large part of the web pages so eliminating them will be a saving in storage and indexing. In this paper three page structure based algorithms developed by different authors for detecting content blocks are discussed.

II. RELATED WORK

Kushmerick [1] has proposed a feature based method which identifies internet advertisements in a web page. It is mainly used for removing advertisements and does not remove other non-content blocks. Bar-Yossef and Rajagopalan [3] have proposed a method to identify frequent templates of Web pages and pagelets. Page level template detection is done by D.Chakraborti et al.[6] They examine the page's features and these features are used to score the DOM tree nodes. Page level templates are generated by doing isotonic smoothing on classifier scores.

III. ALGORITHMS FOR EXTRACTING CONTENT BLOCKS

On Internet large amount of information is available in the form of HTML pages. Most sites use the same presentation style for maximum or all web pages. The non-content blocks share some common contents and presentation styles. While main content blocks are different in their content and /or their presentation style. Various algorithms developed to extract content blocks from web pages share this observation. Three algorithms discussed below are based on the same observation. Based on the Document Object Model, a web page can be parsed and represented as a tree structure. In this tree leaf nodes contain actual contents and intermediate nodes are different tags. All the algorithms discussed below use this DOM tree structure.

A. An Entropy Based Method

Many web sites use a HTML tag <TABLE> to layout their pages. Lin and Ho[2] used this observation to develop Infodiscoverer system to extract content blocks. In this method first a coarse tree structure is obtained by parsing a HTML page using a <TABLE> tag. Each internal node shows a content block containing one or more content strings as its leaf nodes. After parsing a

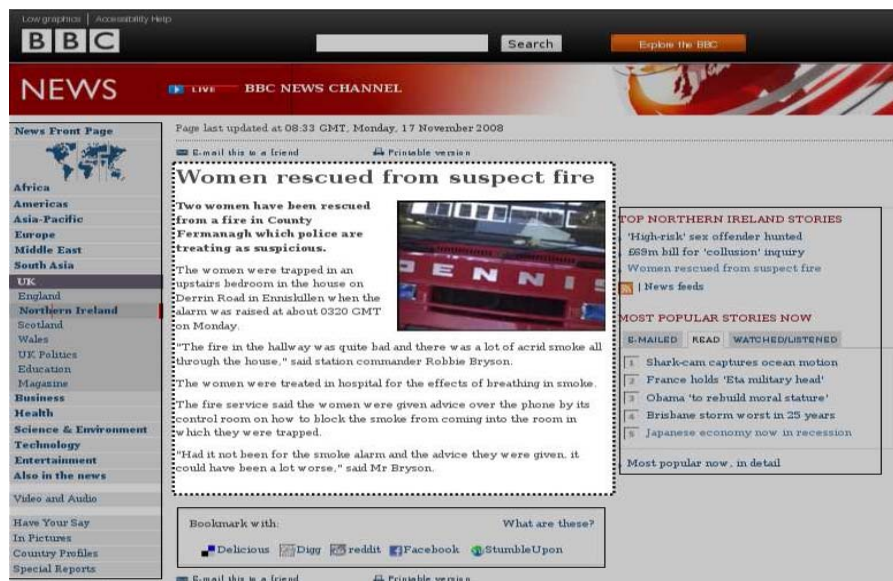


Figure 1. A web page from BBC and its blocks

web page into content blocks features of each block are extracted. Here features means the meaningful keywords. After extracting features entropy value of a feature is calculated according to the weight distribution of features appearing in a page cluster. Next step is calculation of entropy value of a content block. It is given by summation of its features entropies. i.e.

$$H(CB_i) = \sum_{j=1}^k H(F_j) \quad \text{where } F_j \text{ is a feature of } CB_i \quad (1)$$

with k features

The equation can be normalized as content blocks contain different numbers of features

$$H(CB_i) = \frac{\sum_{j=1}^k H(F_j)}{k} \quad (2)$$

The entropy of a content block $H(CB)$, is the average of all entropy values in that block. Using this $H(CB)$ a content block is identified as informative or redundant. If the $H(CB)$ is higher than a threshold or close to 1 then content block is redundant as most of the block's features appear in every page. If $H(CB)$ is less than a threshold then the content block is informative as features of the page are distinguishable.

B. Using Style Trees

Yi, Liu and Li [4] considers non-content blocks as noise in the web page. They use a tree structure called Style Tree, to capture common presentation styles and actual contents of the pages in the given web site. A Style Tree can be built for the site by sampling the pages of the site. This tree is referred as Site Style Tree (SST). As discussed above each HTML page corresponds to a DOM tree. A DOM tree can represent the presentation style of a single HTML page, but it is difficult to study the overall presentation style and content of a set of HTML pages. It

is difficult to clean pages based on individual DOM trees. So a tree structure known as Style Tree is used for this purpose. Figure 2 shows DOM trees and the style tree. DOM trees are combined in a style tree.

Fig. 2 shows two DOM trees t1 and t2. All tags in t1 have their corresponding tags in t2 except the bottom tags P-IMG-SPAN. These two trees are compressed in a style tree. A count is used to find out how many pages have a same style at a particular point in the style tree.

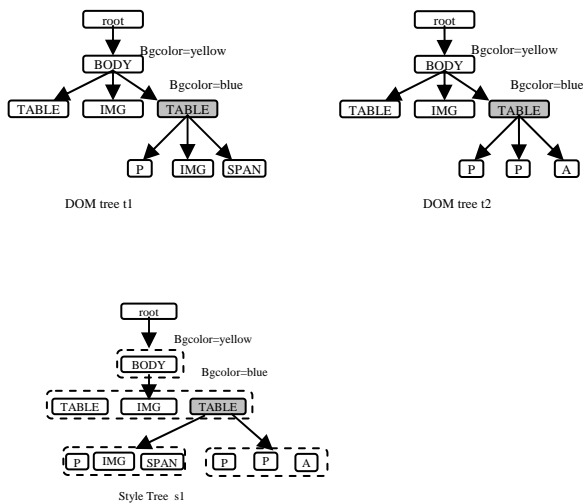


Figure 2. DOM trees t1 and t2 and a Style tree s1

TABLE-IMG-TABLE is the same presentation style used in both the pages. This sequence of tags is known as style node. A sequence of tag nodes inside the style node are known as element nodes. Trees t1 and t2 differ below the rightmost TABLE tag meaning that there are two different presentation styles. This is represented by two different style nodes, one is P-IMG-SPAN and other is P-P-A. So now it is easy to check which parts of the DOM are common and which parts are different. For building

Style Tree for pages of a web site first DOM tree for each page can be prepared and then these DOM trees are merged in a top down manner to form a style tree.

Two importance values, presentation importance and content importance, are used to find importance of an element node. The presentation importance is used to detect noises with regular presentation styles while content importance is used to detect those main contents of the pages that may be represented in similar presentation styles. More presentation styles an element node has more important that node is and more diverse the actual contents of an element node are more important that node is. The greater the combined importance of an element node is, more likely it is the main content of the web page. To eliminate noise from any web page a Site Style Tree is mapped to the web page.

C. Content-Extractor Algorithm

To design content-extractor algorithm Debnath et al [5] used the same basic concept used by Lin[2], that a <TABLE> tag is used to design maximum web pages. Unlike Lin they make use of some other html tags also while designing the algorithm. Similar blocks across different web pages obtained from different web sites can also be identified using this algorithm. In a table occurring in a web page, each cell is considered as a block. Where tables are not available, blocks can be identified by partitioning a web page into sections that are coherent. Many times news articles written by global news agencies appears in many news papers. User wants only one of these several copies of articles. These copies of articles differ only in their non-content blocks, so by separating non-content blocks from content blocks these same copies can be identified. As only unique articles are returned this will improve search results.

Content block can be identified based on the appearance of the same block in multiple web pages. The algorithm first partitions the web page into blocks based on different HTML tags. The algorithm then classifies each block as either a content block or a non-content block. The algorithm compares a block, B , with the stored block to check whether it is similar to a stored one, if so then it is not necessary to store that block again.

A block or web-page block B is a portion of a web-page enclosed within an open-tag and its matching close-tag. The open and close tags belong to an ordered tag-set T that includes tags like <TABLE>, <TR>, <P>, <HR>, and . <TABLE> comes as the first tag in that list. The order of the tags is based on the observations of web-page design. For example, <TABLE> comes as the first partitioning tag since more instances of in <TR> / <TD> which are sub-element of <TABLE>, than <TABLE>s coming inside . Algorithm partitions a web-page into blocks, based on the first tag in the list. It continues sub-partitioning the already-identified blocks based on the next tags in the list. Blocks may include other smaller blocks, and have features like text, images, applets, javascript, etc. Most, but not all, features are associated with their respective standard tags. For

example, an image is always associated with the tag .

This algorithm eliminates redundant blocks depending upon the inverse block document frequency (IBDF) of a block. The IBDF is inversely proportional to the number of documents in which the block occurs. The blocks that occur in multiple pages are redundant blocks and block which appear in one page is a content block.

To extract content block similarity between two blocks must be found out. For this block feature vectors of two blocks are used. These features are number of images, number of terms etc. If a feature is present in a block then its corresponding entry in the feature vector is one otherwise it is zero. Two blocks are identical if the similarity feature between two blocks is greater than a threshold value.

CONCLUSION

To summarize all the techniques find content blocks efficiently. Content-Extractor uses simple heuristics to detect primary content blocks. If there are web pages whose elements have the same style but different contents which are non-content blocks, then the algorithm would not be able to detect that. Style trees is a efficient technique to detect content blocks but constructing these style trees is a complex task. In Content-Extractor there is no overhead of constructing Style trees. These algorithms reduces storage requirements and provide fast search.

REFERENCES

- [1] N. Kushmerick, "Learning to remove internet advertisements," In third annual Conf. on Autonomous Agents, ACM press, NY 1999.
- [2] S. H. Lin and J. M. Ho, "Discovering Informative Content Blocks from Web Documents", Proc. Eighth ACM SIGKDD Int'l conf. Knowledge Discovery and Data Mining, pp. 588-593, 2002.
- [3] Z. Bar-Yossef and S. Rajagopalan. Template detection via data mining and its applications. In the Eleventh International World Wide Web Conference (WWW 2002). ACM press, New York, 7-11 May 2002.
- [4] B. Liu, K. Zhao, and L. Yi, "Eliminating Noisy Information in Web Pages for Data Mining", Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 296-305, 2003.
- [5] S. Debnath, P. Mitra, and C.L. Giles, N.Pal "Automatic Identification of informative sections of Web Pages, IEEE Transaction on Knowledge and Data Engineering, 2005.
- [6] D. Chakraborti, R. Kumar, K. Punera, "Page level template detection via isotonic smoothing", in WWW'07, 2007
- [7] Hui Wang, Bing Wang, Zhang, "Primary Content Extraction with Mountain Model", IEEE, 2008.
- [8] World Wide Web Consortium, World Wide Web Consortium Hypertext Markup Language.