ISSN 1512-1232

ELIMINATE THE NOISY DATA FROM WEB PAGES USING DATA MINING TECHNIQUES

Sekhar Babu Boddu

Assistant Professor, Department of MCA,
KL University, Guntur, Andhrapradesh, India -522501
sekhar99@gmail.com

Abstract

The research in web mining aims to develop new techniques to effectively extract useful information from web pages. Due to the heterogeneity and lack of structure of web data, automated discovery of targeted information is challenging. It calls for novel methods that draw from wide range of fields spanning data mining, natural language processing, machine learning, statistics, databases, and information retrieval. In the past few years there was a rapid expansion of activities in the web mining field, and in particular in area of web content mining. The objective of Web content mining is to extract specific information of interest to the user removing unnecessary things like, navigation panels, advertisement, hyperlinks, date and time formatted data, noise and redundant data. This research proposes a new approach for this purpose. The web pages resulting from search engines are divided into frames and each frame is then sub divided into blocks which are processed for removal of all unwanted information as stated above. The user will be presented with precise and specific web pages of his interest.

Keywords: *Web mining, Web Content mining, Block Splitting.*

1. Introduction

As the amount of information obtained on the web is increasing radically, more number of redundant of web content also grows at the same time. Therefore, updating incoming data and retrieving useful information without duplicate data from the web, the web mining research communities to concern an activity that there is quickly and efficiently information from the web. Maximum number of web search engines is typically works conventional information retrieval and data mining techniques to discover automatically useful and previously unknown information from the web content. Web Content mining aims to discovering useful information or knowledge from web page contents rather than hyperlinks and goes beyond using keywords in a search engine. Web content consists of information such as unstructured free text, image, audio, video, metadata, and hyperlink. Search engines, subject directories, intelligent agents, cluster analysis, and portals are used to find out what a user might be looking for. Web content categorization with a content database is the most important tool to the efficient use of search engines. A customer requesting information on a particular subject or item would otherwise have to search through thousands of results to find the most relevant information to his query. Thousands of results through use of mining text are reduced by this step. This eliminates the frustration and improves the navigation of information on the Web.

Web mining research is an emerging area from lot of research communities, for instance Database, Information Retrieval, and Artificial Intelligence, machine learning and also psychology and statistics. By using the crawler system, to collects the right information from the web is one of the fundamental issues of Web mining. , Web mining can be divided into three different types, which are Web usage mining, web content mining and Web structure mining. Web usage mining is a process of extracting useful information from server logs i.e. user's history. Web structure mining is the process of using graph theory to analyze the node and connection structure of a web site. There are three types of Web document data; those are core data, redundant data and hidden data.

ISSN 1512-1232

From a web page user can view the data that known as core data. Take example, the important information in the news article Web page is the core data. To improve the Web content business improvement redundant information is used. Web documents also comprise “hidden information” like HTML tags, script language and programming comments, which is called ‘hidden information’ since it is not visible for end users. The Web pages being different types called heterogeneous and lot of pages are semi structured and noisy. The challenging issue must be discover and extract the useful information from the web pages. The recognition and elimination of the noise is the difficult to extract the useful information from the web pages. Generally web contains different types of data formats which includes text, image-maps, logos, advertisements search boxes, footers and headers, navigational links, related links and copyright information along with the primary content.

In this paper, we present an approach to partitioning of Web page content that is more general than that in prior work. This approach is based on the observation that information about spatial locality is most often used to **cluster**, or draw boundaries around groups of items in a Web page, while information about presentation style similarity is used to **segment**, or draw boundaries *between* groups of items. These two types of information are complementary. By clustering items into large groups using spatial locality first, and then by segmenting them within those groups using presentation style, we can reduce errors due to selection of incorrect patterns. The key novel features of our approach are:

i) **Determining spatial locality** Instead of relying on the DOM tree, where information about spatial locality is limited to parent-child relationships between HTML tags, we use the Mozilla frame tree to get exact coordinates for the locations of pieces of content.

ii) **Finding presentation similarities** we permit relaxed matching on presentation styles: presentation styles that are similar but not identical can form parts of patterns for segmentation.

iii) **Finding visual separators** we use the presence of separators as strong segmentation cues. We find separators using HTML tags and visual layout information from the Mozilla frame tree.

2. Review of Recent Research Work

Segmenting Web Pages & Detecting Noise is the technique [1] that normally Web page is consists of different blocks or areas, e.g., core content areas, navigation panels, advertisements area, etc. It is automatically separated the area using this technique for several practical application. For example, in Web data mining, e.g., classification and clustering, identifying main content areas or removing noisy blocks (e.g., advertisements, navigation panels, etc.) enables one to produce much better results. It was shown in, that the information contained in noisy blocks can seriously harm Web data mining. The new approach is Web browsing using a PDA such as small screen device. To Identifying different content blocks allows one to re-arrange the layout of the page so that the main contents can be seen easily without losing any other information from the page. In this past two years, several papers have been published on this topic this research also includes detecting common layout and templates of Web pages.

G.Poonkuzhali et.al [2] proposed that the Web documents are extracted from the search engines by giving query by the user to the web. Then the obtained webs documents are preprocessed, i.e., stop words, stem words and except text other data such as hyperlinks, sound, images etc. are removed. Then the number of documents extracted on the web is counted. Next, n x m matrix representation are generated for all the extracted documents based on four tuples namely, number of pages, paragraphs, lines and word occurrences. Then all the elements of 4 tuples taken from n x m matrix of first two documents are compared and its outcome is stored using signed approach. Finally, Redundancy computation is done based on the results of similarity computation.

Giuseppe Antoio Di Lucca et al. [3] proposed an algorithm based on clone detection and similarity metrics to detect duplicate pages in web sites and application implemented with HTML which works only for structured web documents.

Min-yan Wang et al. [4] suggested a web page de-duplication method in which the information including original websites and web titles are extracted to eliminate duplicated web

ISSN 1512-1232

pages based on feature codes with the help of URL hashing. Through this method large-scale duplicated web pages can be eliminated but extraction of feature codes takes much time.

In order to cope with Web page noise and to boost web mining, a feature weighting system has been introduced by Lan Yi *et al*. [5]. At first, for capturing the general structure and comparable blocks in a group of Web pages, the technique built a compacted structure tree. Then information based measure was employed to assess the significance of every node in the compacted structure tree. On the basis of the tree and its node significance values, their technique allotted a weight to each word characteristic in its content block. The resultant weights were employed in Web mining. Using two Web mining tasks, namely Web page clustering and Web page classification, the proposed technique was assessed. Experimental outcome revealed that their weighting technique was able to considerably develop the mining results.

A.K. Tripathy and A.K. Singh [6] were introduced A technique which was employed to eliminate noise from web page. A tree structure, called the Pattern Tree was proposed to capture the general presentation styles and the definite essence of the pages in a specified Web site. A Pattern Tree called the Site Pattern Tree (SPT) was put up for the site, by sampling the pages of the site. A measure which was based on the information was then introduced to decide which parts in SPT represent noises, and which parts represent the core contents of the site. By mapping any Web page to the SPT, the noises were detected and eliminated from that particular Web page. Web clustering which is a data-mining task evaluated the proposed technique.

To remove key information out of web pages that comprised of noisy information, a technique was introduced by Chao Wang et al. [7]. Two steps were involved in this technique: the first was to take out a list of candidate key information, and the second one was to apply entropy maneuver for filtering the noisy information and find out key information. As per the experimental outcome it illustrated that the technique was efficient in finding out vital information. G Poonkuzhali et al. [2] by distinguishing the redundant links from the web documents utilized set theoretical (classical mathematics) such as subset, union, intersection etc., proposed an algorithm for mining the web content. Then for obtaining the requisite information, the redundant links were taken out from the original web content by the user.

3. A new approach for finding blocks using clustering techniques

Semantically related pieces of content in a Web page tend to be located near each other, often sharing the same alignment. Since a frame tree represents the visual layout of a Web page, geometric alignment of frames may imply semantic relatedness. If all descendants of a frame are consistently aligned either along *X* or *Y* axes, we call such a frame *consistent*. A *maximal semantic block*, or simply *block*, is the largest of the consistent frames on the path from a leaf to the root of a frame tree. Thus, it is likely to be the largest possible cluster containing semantically related pieces of content.

The *Find Blocks* algorithm is used to find the blocks in a frame tree. The algorithm runs a depth-first search over the frame tree and recursively determines whether the frames are consistent, ignoring the alignment of leaf frames. A frame is *consistently X-aligned* if all of its non-leaf descendants are X-aligned. Similarly, a frame is *consistently Y-aligned* if all of its non-leaf descendants are Y-aligned. Otherwise, the frame is not considered to be consistent. In this case, all of its children are marked as blocks.

Algorithm *FindBlocks*
Input: *Frame*: node of a frame tree
Output: *Blocks*: set of maximal semantic blocks
1. Identify all children *C*1, *C*2, . . . , *Cm* of *Frame*
2. *Frame.IsConsistent*←**true**
3. **for** *j* ← 1 **to** *m*
4. **do if** *Cj .IsLeaf* = **false**

ISSN 1512-1232

5. **then** *FindBlocks(Cj)*
6. **if** *Cj .Alignment* = NONE
7. **then** *Frame.IsConsistent*←**false**
8. **if** *Frame.IsConsistent* = **false**
9. **then for** *j* ← 1 **to** *m*
10. **do if** *Cj .Alignment* _= NONE

11. **then** *Blocks*←*Blocks* ∪*{Cj}*

12. **else** *Frame.Alignment*
←*GetAlignment(Frame)*
13. **if** *Frame.Alignment* = NONE
14. **then for** *j* ← 1 **to** *m*
15. **do if** *Cj .Alignment* _= NONE

16. **then** *Blocks* ←*Blocks* ∪

{Cj}
17. **return** *Blocks*

The *FindBlocks* algorithm uses the *GetAlignment* algorithm to check whether the children of a frame have matching alignment. The *GetAlignment* algorithm determines that a frame is *X-aligned* if all of its children are aligned on the left, right, or center of the X-axis. Y-alignment of a frame is computed in a similar fashion.

Algorithm *GetAlignment*
Input: *Frame*: node of a frame tree
Output: *Alignment* : alignment of *Frame*'s descendants
1. Identify all children *C*1, *C*2, . . . , *Cm* of *Frame*
2. *XFirst*←*C*1.*X*; *Y First* ←*C*1.*Y*
3. *XAlignedDescends*←**true**
4. *Y AlignedDescends*←**true**
5. *Alignment*←NONE
6. **for** *j* ← 2 **to** *m*
7. **do if** *Cj .IsLeaf* = **false**
8. **then** *XCord* ←*Cj .X*
9. *YCord* ←*Cj .Y*
10. **if** *XCord* _= *XFirst*
11. **then** *XAlignedDescends*←**false**
12. **if** *Y Cord* _= *Y First*
13. **then** *Y AlignedDescends*←**false**
14. **if** *Cj .Alignment* _= *XAlign*
15. **then** *XAlignedDescends*←**false**
16. **if** *Cj .Alignment* _= *YAlign*
17. **then** *Y AlignedDescends*←**false**
18. **if** *XAlignedDescends* = **true**
19. **then** *Alignment*←*XAlign*
20. **if** *Y AlignedDescends* = **true**

ISSN 1512-1232

21. **then** *Alignment*←*YAlign*
22. **return** *Alignment*

4. Segmenting: The Partition Finder

In web page content, geometric alignment is generally a strong indicator of semantic relatedness. However, content that is geometrically aligned can frequently be segmented into smaller content-related groups. For example, on the NY Times home page in Figure 2 the headline story items all have the same Y-alignment, but they form 2-4 individual stories, each composed of an article title, a by line and short abstract, and related links or multimedia. Semantically related content also exhibits similarity in presentation style. Lists of pieces of content that are all semantically related will form presentation style patterns. For example, on the NY Times home page each headline has a large font title in a blue colour, a smaller font by line in grey, a short abstract in a small black font, and an optional list of related stories in a small blue font. If we can find these patterns in the presentation styles of frame tree node sequences, then we can segment content into semantically related partitions.

The *Partition* algorithm is used to find the partitions in a block in the block tree. The algorithm runs bottom-up over the frame sub tree formed by a block. For each node *N*, it determines the maximal repeating pattern(s) in presentation style in *N*'s children. Two issues complicate this process.

First, the patterns may not be exact; for example, there may be no list of related items in one headline. We use separators to determine how much content a pattern should cover and help choose between multiple possible patterns [2]. We define a separator as follows: A *separator* is either one of the HTML tags HR or P, or is white space (horizontal or vertical) between two adjacent nodes that is greater than the mean amount of white space between adjacent nodes in this node list. Second, the elements of the pattern may not be exact matches; for example, on the NY Times home page the first headline has a slightly larger font size than the other headlines but all the headlines have the same colour and font. We use relaxed matching to deal with slight variations in presentation style. The presentation style of a node is a tuple formed of the font type, font size and font style (colour, bold face, italics etc.) of text in the node. We say that two nodes have similar presentation style if the font type and font style of the two nodes is identical, and the font sizes are within two points of each other.

The *PartitionList* algorithm finds maximal repeating patterns in a list of nodes. The *GetFirstSequence* and *GetNextSequence* methods return a sequence of nodes bounded by separators. The method *SimilarSequence* computes the longest common subsequence of two node sequences. If this subsequence covers at least 60% of each node sequence, then the two sequences are considered to be similar.

Algorithm *PartitionList*
Input: *L*←(*C*1, *C*2, . . . , *Cm*): children of block root *N*
Output: *P*: A partitioning of these *C*1, *C*2, . . . , *Cm*
1. **for** *j* ←1 **to** *m*
2. **do** *Replace*(*Cj, Partition*(*Cj*)*, L*)
3. *Current* ←*GetF irstSequence*(*L*)

Figure 2. Block and partition finding in the NY Times home page

4. **if** *Equal*(*Current,L*)
5. **then return** *L*
6. *Replace*(*Current, PartitionList*(*Current*)*, L*)
7. **while** *NotEmpty*(*Current*)
8. *Next*←*GetNextSequence*(*L,Current*)
9. *Replace*(*Next, PartitionList*(*Next*)*, L*)

ISSN 1512-1232

10. **if** *SimilarSequence*(*Current,Next*)
11. **then** *addT oGroupNode*(*Current,Next,L*)
12. *Current* ←*Next*
13. **return** *L*

5. Examples

In this section, we show example output from our system for a range of Web pages. For each example, a part of the page is shown, along with the block tree (top) and partition tree (bottom). Nodes marked “Group Node” and “Pattern Node” is created by the Partition Finder. Nodes marked with box icons are created by the Block Finder. All other nodes are from the frame tree. Arrows connect content in the page to elements in the block and partition trees. Figure 3 shows a portion of the frame tree and partition tree for a Blackboard user page, focusing on a list of system functionalities. Clustering groups each sub list, including its title and related image. Segmenting merely adds a grouping of the elements of the sub list. This example shows the power of exact visual layout information. Figure 1 shows a portion of the block and partition trees for the Amazon home page, focusing on the vertical menu on the left side of the page. At first impression, the structure of this part of this page seems similar to that of Blackboard.

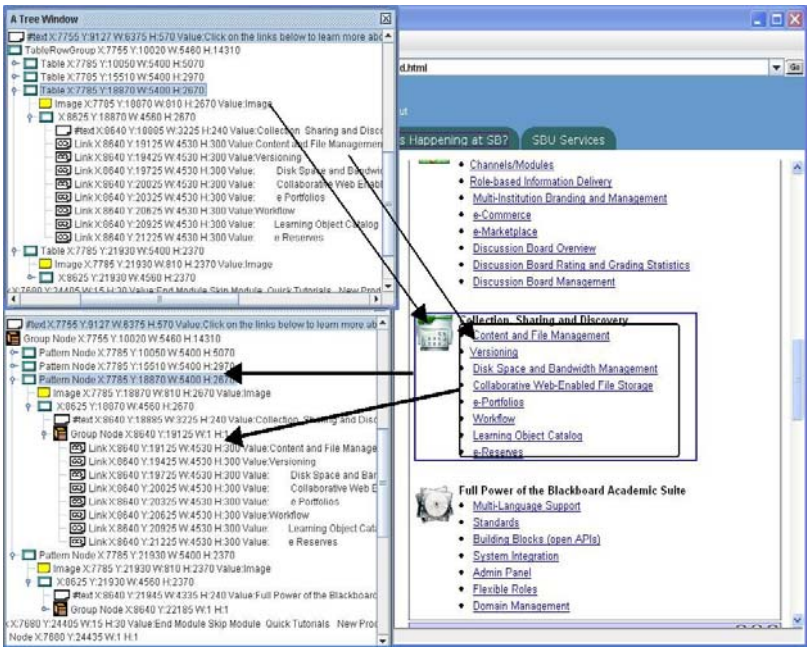

Figure 3. Block and partition finding in the Blackboard system

ISSN 1512-1232

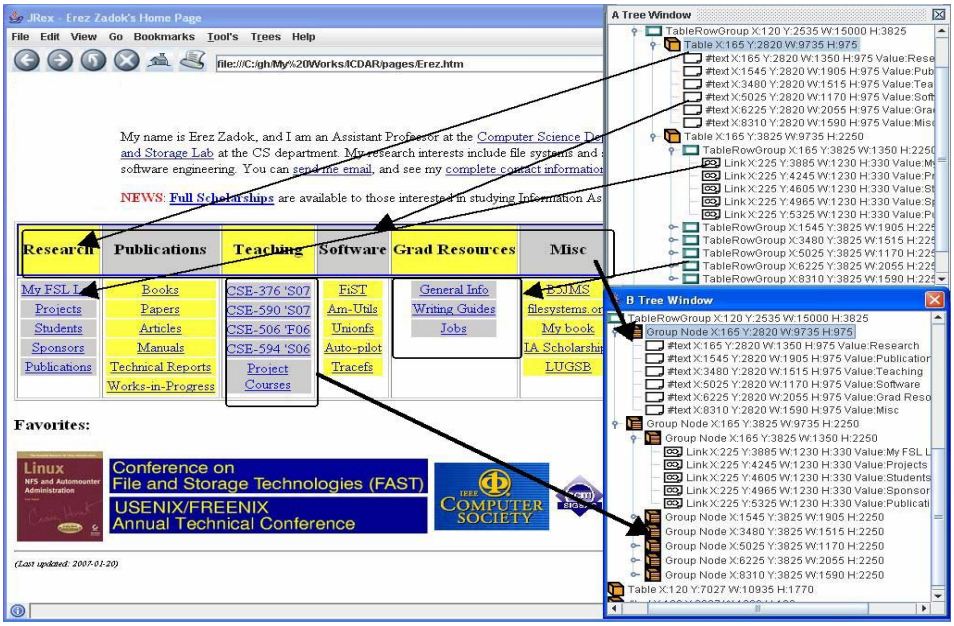

Figure 4. Block and partition finding in a personal home page

However, because the menu items and sub items are all vertically aligned links, clustering groups all the menu items and sub items into one block. Segmenting gives a sequence of partitions, one for each submenu. Partitions are found even though some submenus have more items than others, because the presentation style information in submenu titles and separators serve to identify patterns. Figure 2 shows a portion of the block and partition trees for the NY Times home page, focusing on the headline news. Clustering groups the content relating to the headline news stories into one block, but does not give a block for each story. Segmenting gives a partition for each headline news story, including the information related to that story. Note that even though the font size for the first headline is a little larger than the font size for subsequent headlines, the first headline news story is a partition in the partition tree because segmenting includes relaxed matching for presentation Style information. Figure 4 shows a portion of the frame tree and partition tree for a personal home page that includes a table. There are two ways for items to be grouped in a table: as a set of rows (horizontal alignment gets priority) and as a set of columns (vertical alignment gets priority). The Block Finder currently prioritizes horizontal alignment, so the set of blocks and partitions for this table is incorrect from the user's perspective. In the future, the Block Finder may also use separator detection – because the white space between columns in this table is greater than that between rows, this would help to disambiguate in this case.

6. Conclusions and Future Work

In this paper, we have presented a novel approach to partitioning Web page content. This approach uses visual layout and presentation style information in a two-stage process that gives good results for a range of Web pages from different domains. We are currently planning an evaluation of our partitioning system. We are also working on automatic labelling of blocks and partitions using natural language processing techniques. This partitioning system is already in use in two applications: one for Web accessibility, and one for transaction modelling.

References

[1] A. Arasu and H. Garcia-Molina. Extracting structured data from web pages. In p*roceedings of SIGMOD 2003*, 2003.

[2]. G.Poonkuzhali; K.Thiagarajan; K.Sarukesi2009): Elimination of redundant links in web pages –Mathematical Approach,World Academy of Science, Engineering and Technology,V52,pp.

ISSN 1512-1232

[3]. Giuseppe Antoio Di Lucca; Massimiliano; Anna Rita Fasolina(2002): An Approach to identify Duplicated web pages, in proceedings of the 28th Annual International Computer Software and Applications Conference, IEEE computer Society press

[4]. Min-yan Wang ; Dong-Sheng Liu(2009): The Research of web page De-duplication based on web pages Re-shipment Statement, First Interrnational Workshop on Database Technology and Applications, pp.271-274 [5]. Lan Yi and Bing Liu, "Web Page Cleaning for Web Mining Through Feature Weighting", In Proceedings of the 18th International Joint Conference on Artificial Intelligence,Vol.18, pp.43-50, August 09 - 15, Acapulco, Mexico, 2003.

[6]. A. K. Tripathy and A. K. Singh, "An Efficient Method of Eliminating Noisy Information in Web Pages for Data Mining", In Proceedings of the Fourth International Conference on Computer and Information Technology (CIT'04), pp. 978 – 985, September 14-16, Wuhan, China, 2004.

[7]. Chao Wang, Jie Lua, and Guangquan Zhanga, "Mining Key Information of Web Pages: A Method and Its Application", Expert Systems with Applications, Vol.33, No.2, pp.425-433, August 2007.

[8]. Hui Guo; Mahmud, J.; "A General Approach for Partitioning Web Page Content Based on Geometric and Style Information ",Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference onIssue Date: 23-26 Sept. 2007 On page(s): 929 - 933 Location: Parana ISSN: 1520-5363

[9] D. Cai et al. VIPS: A vision-based page segmentation algorithm. Technical Report MSR-TR-2003-79, Microsoft Research, 2003.

[10] S. Chakrabarti. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In *Proceedings of WWW 2001*, 2001.

[11] N. Chambers et al. Using semantics to identify web objects. In *Proceedings of AAAI 2006*, 2006.

[12] V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Towards automatic data extraction from large web sites. In *Proceedings of VLDB 2001*, 2001.

[13] D. Embley and L. Xu. Record location and reconfiguration in unstructured multiple- record web documents. In *Proceedings of WebDB 2000*, 2000.

[14] O. Etzioni et al. Web-scale information extraction in Know- ItAll. In *Proceedings of WWW 2004*, 2004.

[15] H. Guo and A. Stent. Taxonomy based data extraction from multi-item web pages. In *Proceedings of the Workshop on Web Content Mining with Human Language Technologies at ISWC 2006*, 2006.

[16] S. Mukherjee, G. Yang, and I. Ramakrishnan. Automatic annotation of content-rich HTML documents: Structural and semantic analysis. In *Proceedings of ISWC 2003*, 2003.

[17] I. Muslea, S. Minton, and C. Knoblock. Active learning with strong and weak views: A case study on wrapper induction. In *Proceedings of ICJAI 2003*, 2003.

[18] I. Ramakrishnan, A. Stent, and G. Yang. Hearsay: Enabling audio browsing on hypertext content. In *Proceedings of WWW 2004*, 2004.

[19] H. Takagi, C. Asakawa, K. Fukuda, and J. Maeda. Site-wide annotation: Reconstructing existing pages to be accessible. In *Proceedings of ASSETS 2002*, 2002.

[20] Y. Yang and H. Zhang. HTML page analysis based on visual cues. In *Proceedings of ICDAR 2001*, 2001.

[21] Y. Zhai and B. Liu. Web data extraction based on partial tree alignment. In *Proceedings of WWW 2005*, 2005.

Article received: 2011-11-17