A DOM-Tree based Representation of Web Document Structure for Web Mining Applications

Manoj Kumar Sarma
Department of Computer Science
Gauhati University
Guwhaati-781014, Assam, India
e-mail: manojksarma@yahoo.com

Anjana Kakoti Mahanta
Department of Computer Science
Gauhati University
Guwhaati-781014, Assam, India
e-mail:anjanagu@yahoo.co.in

Abstract—Among the three broad areas of Web mining, Web Structure Mining is the method of discovering structure information from either the web hyperlink structure or the web page structure. In order to apply data mining techniques on web pages, a good and efficient representation of web pages is required that could depict the actual hierarchical structure of web pages. The work presented here aims to find out a representation of web documents that could be used as input for different data mining techniques. The present research work further aims at applying this representation for efficient clustering of web documents where clustering will be performed based on not only the web page content but also the structural layout of a web page.

Keywords*-WebMining; TREE, HTML DOM, String Encoding.*

*****

I. Introduction

Web mining [1] is a special domain of data mining that is defined as the application of data mining techniques in order to mine knowledge from web data. The attention received by web mining from research, IT industry, and several web-based organizations has acquired significant experience.

Web mining has three broad sub-areas- web content mining, web structure mining, and web usage mining.

Web content mining is the method of extracting useful information from the contents of web documents that may contain text, images, audio, video, lists, and tables. Application of text mining to web content is one of the emerging research areas. Web structure mining is the method of discovering structure information from either the web hyperlink structure or the web page structure. Web usage mining is the use of data mining techniques so as to discover web usage patterns from web usage data, generally stored in web server logs.

Applications of Web mining include developing intelligent web search engines, developing e-com (business/auction) sites based on customer preference, navigation patterns, and behavior (especially B2C commerce), understanding Web communities, personalized Portal for the Web, load balancing on web servers, Digital Library and Autonomous Citation Indexing, etc.

There are many available representations and clustering algorithms for text documents. Although widely used, the Vector-Space Model does not preserve the order of the words. Hence it is not suitable for web documents.

An efficient representation of web document has not been found that could reflect the exact hierarchical structure of a web document along with other components apart from text (images, scripts, multimedia, database records, etc.)

II. Literature Review And Motivation

M. J. Zaki [2] in July 2002 proposed a string representation Tree data structures. The author pointed out that in contrast to the standard ways of tree representation, viz. an adjacency matrix or adjacencylist. the adopted string representation of a tree is more versatile and that could be applicable for efficient subtree counting and manipulation. He also pointed out that the representation could be extended to tree-structured documents like HTML or XML documents with proper customization for various web mining applications.

Ziv Bar-Yossefet. al. in 2007 [4] had presented a novel algorithm, DustBuster that could discover rules to transform a given URL to others that are likely to have similar content. The auhors mentioned that the given algorithm performs the task without examining page contents but from previous crawl logs or web server logs.

PradnyaPurandare in 2008 [5] explored the web mining concept with emphasis to how it can be useful and beneficial to the business improvement with the help of facilitating its applications over the Internet.He observed that Web mining could be helpful in making business decisions for further trends.He pointed out that the possible applications could be in online social networking, bioinformatics, e-governance, and e-learning.

KabitaThaoroijam [6] in 2011 in her thesis has presented Text Document Clustering Using a Fuzzy Representation of Clusters. She adopted a two-phase agglomerative approach with the QROCK (Quick ROCK) algorithm to efficiently cluster text documents.The model used for representation of text documents was the Vector Space Model [8].

ManojPandiaet. al. in 2011 [7] have recently reviewed the current research trends in Web Mining. In this paper they tried to give an overall view of Web mining with special emphasis to Web Usage Mining.

ChhaviRana in March 2012 [8] has reviewed the Trends in Web Mining for Personalization (the process of customizing a Web site according to the needs of different users, with the help of the knowledge acquired via analysis of the navigational behavior of the users). She pointed out the need for further experiments - with different combinations of system’s functionalities, along with the study of the contextual behavior of the system from user’s perspective, andthe need for observing how the user profiles evolve over time.

A. B. Manwaret. Al. [9] in May 2012 elaborated the use of the Vector Space Model for information retrieval. The Vector Space Model can be used to represented text documents.

Based on the literature review done so far, the following are some important observations-

1. There are many available representations and clustering algorithms for text documents.
2. An efficient representation of ***web document contents*** has not been found that could reflect various components apart from text (images, scripts, multimedia, database records, etc.). Although widely used, the Vector-Space Model does not preserve the order of the words. Hence it is not suitable for web documents.
3. An efficient representation of ***web document structures*** has not been found that could reflect the exact hierarchical structure of a web document.
4. Although proposed in 2002, the method of string of tree representation of Trees by M. J. Zaki [2] has not been used on web documents for web mining applications. Moreover no alternative method has been proposed or explored for web mining applications considering web page structure as a part of input.

The above observations have motivated us to find out an appropriate representation of Web Documents, based on page structure information, so that the same can be used to apply various data mining techniques on web pages.

III. Background of The Proposed Work

A. The HTML DOM

As per W3C specification, the Document Object Model (DOM) [10] is a programming API for HTML and XML documents that defines the logical structure of documents and the way a document is accessed and manipulated.

With the Document Object Model, programmers can create and build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model (with a few exceptions).

As a W3C specification, one important objective for the Document Object Model is to provide a programming language-independent standard programming interface that can be used in a wide variety of environments and applications.

```
<HTML>
    |--<HEAD>
    |     |--<TITLE>MY PAGE</TITLE>
    |  |--<SCRIPT></SCRIPT>
    |  </HEAD>
    |--<BODY>
|--<DIV>
      |--<B>THIS IS
                        |--<I>MY
                            |--<U>PAPER</U>
</I>
</B>
</DIV>
</BODY>
</HTML>
```

Figure 1. A Simple HTML document showing the DOM tree

B. Salient feature of HTML DOM

The HTML DOM recognizes everything as a node. The whole document is interpreted as a document node, HTML elements as element nodes, attributes as attribute nodes, the text inside HTML tags are text nodes, and even comments are comment nodes.

C. Tree Representation of Web Documents based on HTML DOM

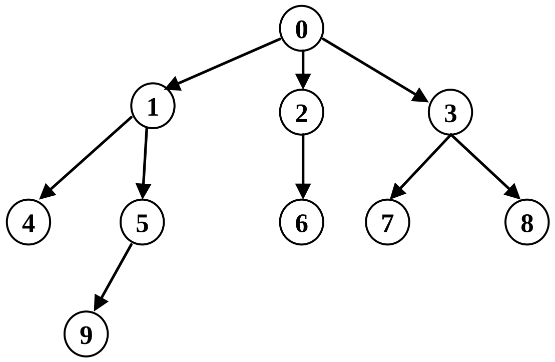
The proposed representation of web documents is based on a representation of tree explained by M. J. Zaki [2]. The author adopted a string representation of a tree that could be applicable for efficient subtree counting and manipulation. The procedure adopted is as follows:

- The generated *string encoding* is to be denoted by τ, of a tree T.
- Initially set τ = ∅.
- Then perform a depth-first preorder search starting at the root, adding the current node's label *x*to τ .
- Whenever to backtrack from a child to its parent add a unique symbol −1 to the string (assuming −1 ∉ L, L= List of nodes).

The author claimed that this format allows to conveniently representing trees with arbitrary number of children for each node. Moreover, since each branch must be traversed in both forward and backward direction, the space usage to store a tree as a string is exactly 2m+1=2n–1; where m=no. of nodes and n=no. of branches.

This has made the string encoding more space-efficient than other representations. Moreover, it is simpler to manipulate strings rather than adjacency lists.

Tree Representation:



String Representation:

014-159-1-1-126-1-137-18-1-1

OR,

0 1 4 -1 5 9 -1 -1 -1 2 6 -1 -1 3 7 -1 8 -1 -1

Figure 2. Example of string representation as explained in [3]

IV. Proposed Work

A. The Proposed Algorithm:

Based on the string representation of trees as proposed by M. J. Zaki [2], an algorithm has been proposed that will read each web page, which is basically assumed to be a HTML document, and will convert the same to a string.

The Algorithm is as follows-

Procedure preorder_depth_first (node node)

```
1.  {
2.    if (node !=NULL)
3.       {
4.          print “[node_name]”
             // Print name of the HTML tag
5.          print “,”
// Print a separator “,” between two HTML tags
6.          for_eachchildnodec of node
7.           {
8.          pre_order_depth_first(c)
                 //  Recursively call the procedure on each
                  // children of the parent
9.               print “$”
                 // Print “$” to backtrack from child to parent
10.         }           // end_for
11.      }// end_if
12.  }      // end of Procedure
```

As shown in the above algorithm, the algorithm uses a recursive call to the procedure pre_order_depth_first() to traverse all nodes in the HTML DOM-Tree in a depth-first manner and prints the node names, separated by comma(,). Whenever the algorithm reaches a child node, it backtracks to the parent node of that child node and marks the backtracking by a ($). Although pre-order traversal of a tree is not sufficient to get the tree back, as different trees may have the same pre-order traversal, the output produced by the above algorithm is unique for each tree, and the same input tree can be constructed back from the output of the algorithm.

B. *Time Complexity of the Proposed Algorithm:*

It is evident that steps 2,4,5& 9 will require a constant amount of time C. If the tree has n nodes then the procedure (step 8) will recursively run n times. Hence the time complexity of the algorithm is O(n + C) which reduces to O(n).

V. Experimental Results

A. *Implementation Details:*

- ***Hardware Environment:*** Intel Core2 Duo CPU T6570 2.10 GHz, 3.00 GB DDR2 RAM.

- ***Software Environment:*** Windows-7 32 bit, R version 3.2.2, RStudio Version 0.99.489.

B. *Results:*

- ***Description of Dataset used (UW-CAN-DATASET):***

  A collection of web documents used for web mining purposes. The document data set has been used for testing the effect of phrase-based document similarity calculation, ascompared to using traditional single-term similarity measures. The dataset was primarily used for the work presented in [3]. The data set consists of 314 web pages from various websites at the University of Waterloo, and some Canadian websites. The data is categorized into 10 categories. Each category is in a separate folder.
- URL:http://pami.uwaterloo.ca/~hammouda/webdata/uw-can-data.zip

(Download size: 1.14 MB; after Unzip: 4.12 MB)

C. *Experimental Results obtained:*

No. of files	314
Processing Time	25.53 Sec
File Size after conversion	279 KB

D. *Example of Output:*

Input: the HTML document shown in Figure 1.
Output:
html,body,p,$title,$script,$div,b,i,u,$$$$$

VI. Conclusion and Future Works

The work presented here aims to find out a representation of web documents that could be used as input for different data mining techniques. The work further aims at applying this representation for efficient clustering of web documents where clustering will be performed based on not only the web page content but also the structural layout of a web page.

References

[1] Jaideep Srivastava, Prasanna Desikan and Vipin Kumar, “Chapter 3: Web Mining- Accomplishments & Future Directions”, URL: www.ieee.org.ar/downloads/Srivastava-tut-paper.pdf.

[2] Mohammed J. Zaki, “Efficiently Mining Frequent Trees in a Forest”, Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), Edmonton, Canada, July 2002.

[3] Hammouda, K. and Kamel, M. "Phrase-based Document Similarity Based on an Index Graph Model", In Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), Maebashi, Japan, 2002. IEEE Computer Society.

[4] Ziv Bar-Yossef, IditKeidar and Uri Schonfeld, “Do Not Crawl in the DUST: Different URLs with Similar Text”, Proceedings of the WWW 2007, May 8–12, 2007, Banff, Alberta, Canada. ACM 978-1-59593-654-7/07/0005.

[5] Pradnya Purandare, “Web Mining: A Key To Improve Business On Web”, IADIS European Conference Data Mining 2008. ISBN: 978-972-8924-63-8 © 2008 IADIS.

[6] Kabita Thaoroijam, “Document Clustering Using a Fuzzy Representation of Clusters”, Ph. D. Thesis submitted to the Department of Computer Science, Gauhati University, Guwahati. 2011.

[7] Manoj Pandia, Subhendu Kumar Pani, Sanjay Kumar Padhi, Lingaraj Panigrahy, and R. Ramakrishna, “A Review Of Trends In Research On Web Mining”, International Journal of Instrumentation, Control & Automation (IJICA), Volume 1, Issue 1, 2011.

[8] Chhavi Rana, “Trends in Web Mining for Personalization”, IJCST Vol. 3, Issue 1, Jan-March 2012. ISSN: 0976-8491 (Online), ISSN: 2229-4333 (Print).

[9] A. B. Manwar, Hemant S. Mahalle, K. D. Chinchkhede, and Vinay Chavan, “A Vector Space Model For Information Retrieval: A Matlab Approach”, Indian Journal of Computer Science and Engineering (IJCSE) Vol. 3 No. 2 Apr-May 2012, ISSN : 0976-5166, pp. 222-229.

[10] https://www.w3.org/TR/WD-DOM/introduction.html.