Journal of Theoretical and Applied Information Technology

www.jatit.org

RE-STRUCTURING HTML DOCUMENTS STRUCTURE AUTOMATICALLY THROUGH CLUSTERING

[1] SARWAR HADI, [2] DR S QAMAR ABBAS [3] SHEENU RIZVI

[1] Research Scholar, Integral University, Kursi Road, Lucknow, India

[2] Prof., Department of Information Technology, Integral University, Lucknow, India

[3] Amity University, Lucknow, India

E-mail: sarwarhadi@acm.org, qamar.abbas@rediffmail.com

ABSTRACT

In this paper we present a novel approach to automatically re-structuring HTML documents by extracting semantic structures from their header and body, The body of a web page is generally software generated via template and it's layout has a physical schema. Our approach is to extract trees that are based on hierarchical relations in HTML documents, for this task we used two algorithms, first is Header extraction Algorithm which extracts header trees from head of HTML document and second is an algorithm for automatically partitioning HTML documents into tree like semantic structures from body part of web pages. Then we use an application called layout changer which changes a layout of one web page to another by aligning extracted header trees and partition trees.

Keywords: *Document Structure, Semantic structures, Block clustering, Text Mining*

1. INTRODUCTION

Due to the wide use of web mark up language HTML, a lot of useful information resides on the web in HTML format. Even other, not so widely used, document formats are often being converted to HTML to make them easily accessible over the web. As a result, we have a very large base of knowledge available over the web. This makes it tempting to apply data mining techniques to web documents. This area of research has been known as "Web Mining"; i.e. data mining on the web. Web Mining comes in different directions. One direction is "Web Usage Mining", which tries to find usage patterns of data on the web, such as tracking user visiting sequence of web sites, and extracting statistical information about web site usage. Another direction is "User Profiling", which has to do with profiling users to find their interests and build a profile that could be used by marketing companies, or even for personalizing web content for different users. The direction we are dealing with here is "Web Content Mining", which is a form of text data mining applied to the web domain, and has to do with finding structures in the content of documents, classifying documents, and clustering documents. Since web documents are not generally well-structured, an effort has to be done to extract useful information from web documents to make it suitable for analysis..

2. DOCUMENT STRUCTURE

Web Documents in HTML (Hyper-Text Mark up Language) are known to be semi structured. The word "mark up" means tagging document elements in a way that facilitates the design of document layout and providing a means of referencing from one document part to another, or even from one document to another. This essentially means that HTML is not a way of structuring the content of a document; i.e. the different document elements are not planned according to a certain schema. Hence they are "semi-structured". Since HTML specifies the layout of the document, it is used to present the document to the user in a friendly manner, rather than specify the structure of the data in the document. Before putting the semi-structured documents into a usable structured form, we first must analyze the structure of web documents and find out what type of information it provides us with, and how we can make use of this information to make a conversion into a usable structured form.

2.1 HTML DOCUMENT STRUCTURE

According to theW3C specification, HTML is a non-proprietary format based on SGML (Standard

Generalized Mark up Language). SGML is a language for specifying document structure according to a certain schema. HTML is an application of SGML for describing web documents that are viewable by web browsers over the web. HTML uses tags to describe different parts of a document for the purpose of presentation to a user by means of a browser.

The global structure of an HTML document is composed of three parts:

1. a line containing HTML version information,

2. a declarative header section,

3. a body which contains the actual document content.

Delimiters (tags) are used to delimit each part of the document. An example of a simple document is presented here:

```
01: <DOC TYPE HTML PUBLIC " -/ /W3C/ /DTC
HTML
     / /EN" " http: //www.w3.org/TR/html 4/strict.dtd">
02: <HTML>
03: <HEAD>
04:  <TITLE> Times of India </TITLE>
05: </HEAD>
06: <BODY>
07: <P> Hello World !
```

The most important thing to note about that example is that information in an HTML document is separated into "meta" information that is described inside the HEAD element, while document "content" is put inside the BODY element.

2.2 DOCUMENT HEAD

Meta information is of very great value to us if it contains information directly related to the contents of the document, which is the case with most tags found in the HEAD element. The most important information that describes the content of a document are listed here.

Element	Description
TITLE	Document title
META-KEYWORDS	Document keywords Used for indexing
META-DESCRIPTION	Brief description of the Document contents
META-AUTHOR	The document author

Although the elements described here are not rendered by a web browser as part of the document, they provide very useful information about the topic of the document in a concise manner, which is exactly what we are looking for. However, since the only mandatory element is the TITLE element, it is not always guaranteed that whoever created the document will provide the rest of the meta information described above. The TITLE element is of very special interest to us, since it is used to identify the content of the document. However, not all documents have context-rich titles; many document titles do not provide much contextual background, e.g." Introduction" instead of "Introduction to Relativity Theory". Thus no assumption should be made about the goodness of a document title when automating web mining tasks.

2.3 DOCUMENT BODY

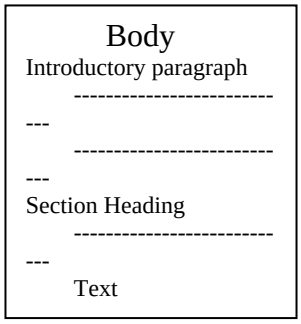
Document content is contained in the BODY segment. Since many HTML tags are used for document body presentation, we will focus only on the tags that provide hints on document structure rather than providing cosmetic layout. Generally speaking, there are two types of elements that are used in the document body: (a) "inline" elements (also known as text-level), and (b) "block-level" elements. Block level elements may contain inline elements and other block level elements, thus creating a hierarchical structure. Inline elements contain only inline elements. Inherent in this structural distinction is the idea that block elements create "larger" structures than inline elements.

Some elements could be used at both the block and inline-level. These are labelled as "Mixed". Some elements have direct relationship with document structure (such as the section heading elements H1,H2,. . . etc.), while others are related to document layout.

The semi-structure that web documents have is a result of the HTML specification.

The intent was to formalize a language that serve document structure as well as document layout. The result is that web documents usually do not conform to certain structuring of the data they contain. Some people will rely on layout tags to form the structure they want instead of thinking out the proper structure. This has resulted in a proliferation of misuse of the HTML language.

Thus, the rationale for restructuring web documents is that the current state of semi-structured web documents is hard to deal with directly, and needs to be mapped to a data-driven structured form first [3].



3. RE-STRUCTURING APPROACHES

The internet has become one of the most important information source, that is currently available on the net in HTML format which grows at a very fast pace, so that we may consider the web as the largest knowledge base ever developed and made available to the public. However HTML sites are in some sense legacy systems, since such a large body of data can not be easily accessed and manipulated. The reason is that web data sources are intended to be browsed by humans, and not computed over by applications. XML which was introduced to overcome some of the limitations of HTML, has been so far of little help in this respect. As a consequence, extracting semantic data from web pages remains a complex task. Most of data on web are available as pages encoded in markup languages like HTML intending for visual browsers. As the amount of data on web grows, locating desired contents accurately and accessing them conveniently become pressing requirements. Technologies like web search engine and adaptive content delivery are being developed to meet such requirements. However web pages are normally composed for viewing in visual web browsers and lack information on semantic structures. The HTML pages are designed to take both structural and presentational capabilities in the mind and these two were not clearly separated. We consider tags of HTML are not stable features for analyzing structures of HTML pages. For semantic rules based approaches, difficulties to learn new rules automatically restrict their feasibilities.

To re-structure layout of one HTML page to another by aligning extracted header trees from the header of two web documents. The header tree is seen as an indented list of blocks where the level of each node's indent is equal to the depth of the node. So the task is to give a depth to each block in a given web page. After that some heuristic rules are employed to construct header trees from a list of depths. Hence the impact to the system is a list of blocks and the output is a list of depths. The algorithm proceeds in two steps, separator categorization and block clustering, The first step estimates local block relations via probabilistic models for characters and tags that appear around separators. The second step supplements the first by extracting the undetermined relations between blocks by focusing on global features. The first algorithm proceeds in a bottle up manner by examining a given block list from trail to head, finding the block that is the most similar to the current block and collecting them into the same cluster. Then all blocks in the same cluster is assigned the same depth[1].

The other re-structuring strategy is to find a method for extracting semantic structure from the body part of HTML page which is generated by software via template. We use the second algorithm to automatically discover and generate a semantic partition tree. Each partition will consist of items related to a semantic concept. Normally in Web pages there is implicitly a fixed "schema" and what changes is the content. Informally a schema for a Web page represents concepts and relationships among them in a hierarchical fashion. Knowledge of the schema is the key to transforming legacy HTML document into more semantics oriented document formats such as XML. This algorithm formulate the problem of schema discovery from HTML documents as one of "automatically discovering semantic structures in HTML documents". The idea underlying in this approach is based on the key observation that in web documents semantically related items, as discerned in their rendered views, exhibit special locality. All the taxonomic items and the corresponding hyperlinks under them are all spatially clustered together in the rendered view. Each of these items appear as the leaf nodes in the DOM (Document Object Model) tree corresponding to the page. It turns out that in HTML documents spatial locality can be captured as "similarity" of path structures in DOM trees. A notion of similarity between paths based on their path structures requires that they be identical while weaker forms can be defined based on edit distance. Based on this notion we can group all the heading and its associated links in one partition, all the links under other heading and their associated links in a different partition, and so on.

Continuing in this fashion we yields a partition tree[2]. To transform the DOM tree of a HTML document into a tree like semantic structure, we simply invoke the top-level algorithm Partition tree on the root of the given DOM tree. This algorithm first traverses the DOM tree top down and then restructures it bottom up. In the algorithm Partition Tree, all the leaf nodes are typed. Internal nodes with only one child are handled. In such case, the type of this only child node is computed and then simply propagated up the tree. However, for an internal node with multiple children, we first invoke Partition Tree on all of its children to collect their type information. Then the Find Partition is invoked upon this node to perform a pattern discovery on its children nodes. The output of this algorithm is a tree of semantic partitions. The information associated with a partition is the content in the leaf nodes of the partition. This information is summarized by a label of the partition. It is important to label the semantic partitions for the purpose of recovering the implicit schema of the document[4,5,6]. In the last we use an application called layout changer which change a lay out of one web page to another by aligning extracted header trees and partition trees of two HTML documents which we get from the above two algorithms.

4. CONCLUSION

In this paper, we described an automatic method of re-structuring HTML pages by first extracting header trees and partition trees that give hierarchical structures through header (Most important information of the document that describes the content of a document are found in the HEAD) and the body part (Document content is contained in the BODY segment) of a web page, And there after using layout changer to change the layout of web pages by aligning header trees and partition trees. Using these two algorithms at the same time on test documents has improved the performance of restructuring without loss of semantics. Finding other re-structuring strategies in addition to the ones proposed in this paper is also important future work.

5. REFRENCES

[1] Minoru Yoshida, Hiroshi Nakagawa, "Re-formating Web Doduments via Header Trees", *Information Technology Center, University of Tokyo, Japan*.

[2] Saikant Mukherjee, Guizhen Yang, Wenfang Tan, "Automatic Discovery of Semantic Structures in HTML Documents", *State University of Newyork NY USA, In Proceedings of ICDAR 2003.*

[3] Khalid M. Hammouda, "Web Mining: Identifying Document Structure for Web Document Clustering", *University of Waterloo Canada*.

[4] M. Garofalakis, A. Gionis, R. Rastogi, S. Seshadri, and K. Shim, "XTRACT:A system for extracting document type descriptors from xml documents.", *In ACM SIGMOD, 2000.*

[5] R. Goldman and J. Widom, "DataGuides: Enabling query formulation and optimization in semi structured databases." *In VLDB, 1997.*

[6] S. Nestorov, S. Abiteboul, and R. Motwani, "Extracting schema from semistructured data", *In ACM SIGMOD, 1998.*