WEB PAGE RETRIEVAL BY STRUCTURE

William Grosky and Gargee Deshpande

Abstract: Our research explores the possibility of categorizing webpages and webpage genre by structure or layout. Based on our results, we believe that webpage structure could play an important role, along with textual and visual keywords, in webpage categorization and searching.

Keywords: content-based retrieval, genre detection, layout ontologies.

1. Introduction

The amount of data available electronically on the web has increased dramatically in recent years. Users generally retrieve data by browsing and searching by keywords. This is an example of *content-based search*. In this approach, search is based on the words in the heading of the page or the contents of the images displayed on the web pages, or words occurring as meta-data in pages. The overwhelming amount of information on the web requires a powerful search service to render that information accessible and useful. Without such a search strategy, finding a specific web site can be as difficult as finding a book in a library that has no card catalogue and a completely random method of storing its books.

In recent years much research has been done on querying the web. In this research, the web is viewed as a collection of multimedia documents in the form of pages connected through hyperlinks. Unlike most web search engines, the aim here is to provide more database-like query functionality. Also, application of data mining techniques to the World Wide Web, referred to as *web mining*, has been the focus of several research projects and papers. Web mining has been categorized into *web content mining* and *web usage mining*. Web content mining is the process of finding information from the web, whereas web usage mining is the process of mining user browsing histories for access patterns [1].

We believe that it would also be desirable to see the layout of web pages when querying these pages and grouping them according to these layouts. The term *layout* connotes the spatial relationships between the page contents rendered by particular tags. Thus, web pages can be categorized according to their *layout ontology*. The term *ontology* means a specification of a conceptualization, a set of concept definitions. Broadly speaking, an ontology is a description (like a formal specification of a program) of concepts. Each web page has a structured hierarchy of tags that defines the layout ontology for that particular page. It is possible that two different web pages have a similar structure of their tag hierarchy. Then, the layout ontology of these two web pages is said to be the same. Our aim is to categorize web pages according to these structures. Our belief is that pages with similar layout ontologies have somewhat similar semantics, or at least can be categorized as belonging to the same environment. For example, we will present some preliminary experiments that show that pages from different newspapers are more similar in layout ontology to each other than to the layout ontology of commercial sites selling books. This concept can also be used for extracting data from the web depending upon content as well as the structure.

In this paper, we assume that each web page consists of HTML tags. HTML tags can be broadly categorized as *container tags* and *standalone tags*. Container tags can contain other tags inside them but standalone tags are atomic. Because of the containing capacity of some tags, each web page can be represented by a tree structure of its tags. Thus, for each web page, a tree structure of tags can be determined. Our hope is that we get a somewhat different tree structure of tags for each page from different environments.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related literature. Section 3 covers various conceptual details, while Section 4 discusses implementation details and the various technologies used in our experiments. In Section 5, we give the results of some preliminary experiments, while Section 6 gives some concluding remarks.

2. Literature Review

Our hope is that by automatically characterizing the environment of a particular web page, content-based information can more easily be extracted from it. In [2], information from unstructured and semistructured web documents is retrieved from web pages in chunks called *records*. A record is a group of information relevant to some entity. The final goal is to extract information from these records to populate a relational database. The paper describes a heuristic approach to discovering the record boundaries in web documents. It captures the structure of a document as a tree of nested HTML tags and locates the sub-tree containing the records of interest, identifying candidate separator tags within the sub-tree using five independent heuristics, finally selecting a consensus separator tag based on a combined heuristic.

The five heuristics are OM (ontology matching), SD (standard deviation), IT (identifiable separator tags), HT (highest-count tags), and RP (repeating-tag pattern). Each of these heuristics returns one or more candidate separator tags with a measure of certainty attached to each candidate. Finally, they provide a way to combine these individual heuristics to determine a consensus separator tag and hence discover record boundaries.

The technique we exploit in this paper is based on the work of [3]. In this paper, a computational geometry-based spatial color indexing methodology is examined for efficient and effective image retrieval. In this scheme, an image is evenly divided into a number of M*N non-overlapping blocks, and each individual block is abstracted as a unique feature point labeled with its spatial location, its dominant hue and its dominant saturation. For each set of feature points labeled with the same hue or saturation, a Delaunay triangulation is constructed and then a feature point histogram is computed by discretizing and counting the angles produced by this triangulation. The concatenation of these feature-point histograms serves as the image index. This research work has been the motivation for our research.

Related research field to our approach is the research being done on semistructured data. For retrieving web pages by structure, structures of web pages have to be stored and retrieved effectively. For storing semistructured data, paper [4] argues that languages supporting deduction and object-orientation are particularly well-suited, as object-orientation provides a flexible common data model for handling semistructured data. Paper [5] presents the Lorel language designed for querying semistructured data. The main novelties of Lorel are that it makes extensive use of coercion to relieve the user from the strict typing of a query language, which is inappropriate for semistructured data, and that it provides powerful path expressions, which permit flexibility for declarative navigational access.

As against the data model that is underlying [5], [6] argues that semistructured data can be stored in relational format by exploiting the regularities inherent in existing semistructured data instances. The claim is that most of the data will be stored in relational format and future insertions can occur in a self-describing way. In [7], an approach of creating wrappers for storing semistructured data is discussed.

3. Web Page Retrieval by Structure

Motivated by the ultimate goal of automatically computing efficient and effective descriptors which symbolize web page structure, this research has been directed towards the management of information such as the levels of tags comprising a web page, the tag hierarchy, and the area covered by the tags on the web page. As nesting of tags plays important role in defining structure of the web page, dominance of tags is considered for each level.

Hope is that within broad domain of web pages this technique can be used to find the structure of web pages and categorize web pages according to the structure. Further to such categorized web pages, semistructured techniques can be applied for effective content retrieval.

The paper [2] has been the motivation behind this research. This paper examines the use of a computational geometry-based spatial color indexing methodology for efficient and effective image retrieval. In this scheme, an image is evenly divided into number of M*N non-overlapping blocks, and each individual block is abstracted as unique feature point labeled with its spatial location, dominant hue, and dominant saturation. For each set of feature points labeled with the same hue or saturation, a Delaunay triangulation is constructed, followed by computing a feature point histogram realized by discretizing and counting the angles produced by this triangulation. The concatenation of all these feature point histograms serves as the image index.

Following the same concept, we examine the use of a computational geometry-based web page structure analysis for effective web page structure matching. In this scheme, a web page is evenly divided into number of M*N non-overlapping blocks, and each individual block is abstracted as a unique tag that covers the maximum area in that block at its level. For each feature tag selected we get a set of feature points. For each set of feature points labelled with the same tag, we construct a Delaunay triangulation and then compute the feature point histogram as mentioned above. The concatenation of these feature-point histograms serves as our web page descriptor. Web page descriptors are further used to categorize different web pages.

As mentioned previously, we assume in this paper that each web page consists of HTML tags. HTML tags can be broadly categorized as *container tags* and *standalone tags*. Container tags can contain other tags inside them but standalone tags are by themselves. Examples of container tag are the TABLE tag and the PARAGRAPH (P) tag, while examples of standalone tags are the BASE tag and the AREA tag. Because of the containing capacity of the tags, a web page corresponds to a tag tree structure, called a *tag tree*. Not all web pages have similar tag trees. In this paper, we study page layouts to try to categorize web pages semantically.

For our analysis, the level of a tag plays an important role when finding tags covering the maximum area in a block. An example of a web page tag hierarchy is as follows:

```
<HTML>
        <HEAD>
                <TITLE>
                </TITLE>
        </HEAD>
        <BODY>
                <P>
                        <TABLE>
                                <TR>
                                        <TD>
                                        </TD>
                                </TR>
                        </TABLE>
                        <B>
                        <B>
                </P>
        </BODY>
</HTML>
```

In the web page example given above, the <HTML> tag is at the highest level. Nested in the <HTML> tag are tags <HEAD> and <BODY>. Inside the <BODY> tag is a <P> tag and inside the <P> tag is a <TABLE> tag and so on. When we consider the concept of area covered by a tag on a web page, the concept of level plays an important role. In the example given above, the level of tag <HTML> is 1, the level of the <BODY> tag is 2, the level of the tags <TABLE> and <B> are 3, the level of tag <TR> is 4, and so on. When calculating the dominant tag at level 3, both <TABLE> and <B> tags are analyzed to check which tag is covering the maximum area in which block on the page. As tag <TR> is inside the <TABLE> tag, the area covered by the <TABLE> tag on the web page contains the area covered by the <TR> tag. So, for the blocks in which the <TABLE> tag is dominant at level 3, it is possible that in this same block, tag <TR> is dominant at level 4.

Now, each web page consists of tag hierarchy. We consider a few tags as characterizing features $F = \{f_1,\ldots,f_k\}$. We believe that the spatial placement and dominance of these various feature tags can be used to characterize the web pages.

The web page is divided into N*M non-overlapping blocks. For each block, at each level, the tag covering the maximum area is found. Then we find for each of the predefined feature tags, which blocks that tag was marked as the predominant tag. We mark the center co-ordinate of all such blocks. The spatial arrangement of these points is an important aspect of our work. As mentioned earlier, we construct a Delaunay triangulation and then compute the feature point histogram by discretizing and counting the angles produced by this triangulation. The concatenation of these entire feature-point histograms serves as our web page descriptor.

It has been shown that histogram intersection is especially suited for comparing histograms for content-based retrieval. Additionally, histogram intersection is an efficient way of matching histograms. The intersection of the histograms W_{query} and $M_{database,}$ each of n bins, is defined as follows:

$$D(W_{query}, M_{database}) = (\sum \min(W_j, M_j)) / \sum W_j$$

The histogram of a web page characterizes the web page depending upon the placement of tags forming the web page. Thus, the above mentioned formula can be used to check the similarity between the two web pages. If two web pages are similar in structure then the histogram of those two pages are bound to be similar. For such web pages, the above formula returns a value close to 1. Similarly, if two pages are very different in structure then the above formula returns a value close to 0.

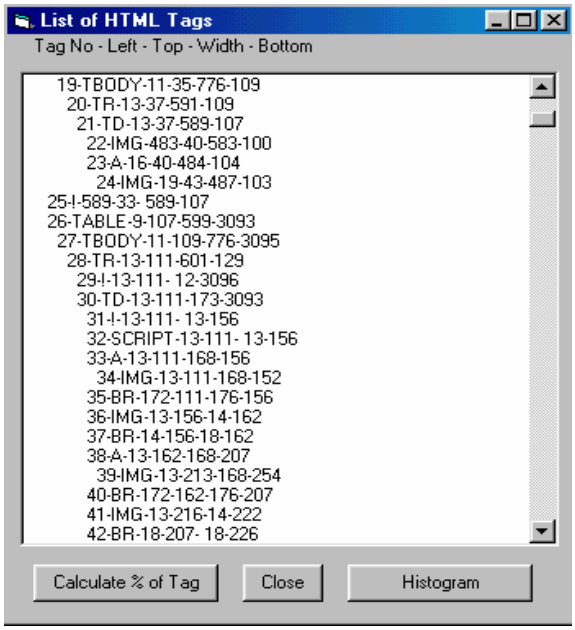
4. Implementation

The input to our system is a web page. Our feature representation is extracted from this web page and matched against those extracted from other web page of known semantics. In more detail, we do the following:

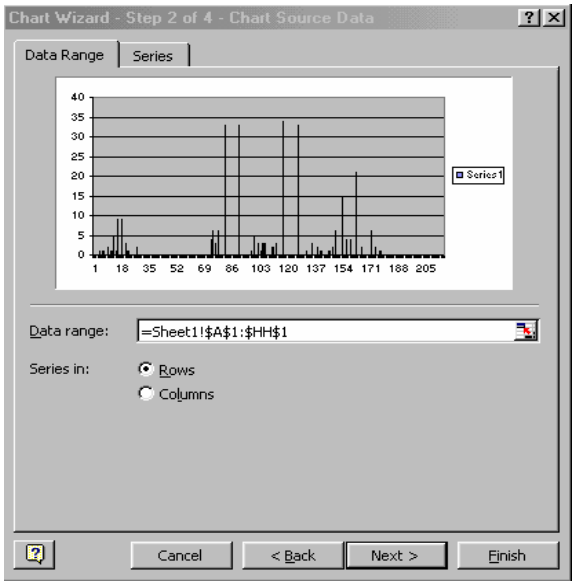
1. Our system accepts a URL as input and displays the given web page using the Internet Explorer engine.
2. The web page displayed is analyzed to get all tags on the page with left, top, right, bottom (X1, Y1, X2, Y2) co-ordinates of area covered by each tag on the page. For each tag, the level of nesting is also saved while gathering this data.
3. The page is normalized to size 512 * 512. The original calculated co-ordinates (X1, Y1, X2, Y2) are re-calculated to map to this normalized size.
4. The page is divided into N*M disjoint blocks. The relevant coordinates of each block is calculated.
5. For each block, it is found out that which tag covers how much area.
6. Depending upon the data gathered in step 5, it is found out for each level, for each block, which tag is covers the maximum area.
7. For each of the selected feature tags, the blocks are found in which the tag covers the maximum area. Center X and center Y coordinates of these blocks are written to a file.
8. Histogram program is run on the file and histogram points calculated by the program are read back into the system. The histogram program used to calculate these points is implemented for the two largest angles of each Delauney triangle using 36 bins. Thus, each bin corresponds to 5 degrees.
9. For each web page, descriptor of (36 * Number of feature tags) bins is calculated.
10. When the descriptors of all the web pages of interest are calculated using steps 1 through 9, the distances between these pages and the database pages are calculated.
11. For each query page, the nearest pages are chosen, based on the distances calculated in step 10.
12. The web pages selected in step 11 are analyzed for category information. The most occurring category is chosen. The query page is categorized using this category.

As an example, consider the following web page:

Here is a snapshot of part of the tag tree, along with the coordinates of the rectangular area covered by the rendering of each tag:



And here is the resulting histogram for the web page:



5. Experimental Results

Our proof-of-concept experiments are carried out on newspaper web pages and e-commerce web pages. Four newspapers and two e-commerce web sites are selected as categories. The categories are: Detroit News,Times of India, Tribune India, Esakal, Amazon.com, and Buy.com.

For each of the newspaper categories, six days of newspaper front pages were analyzed, while from the e-commerce web sites, six web pages were used. Thus, a total of 36 web pages were analyzed.

Initially, we defined a large set of feature tags to ensure a powerful set of independent features for the discrimination of our two classes. This initial set of 52 feature tags were: <A>, <APPLET>, <B>, <BIG>,
, <CAPTION>, <CENTER>, <CITE>, <CODE>, <COL>, <COLGROUP>, <DD>, <DIR>, <DL>, <DT>, <EM>, <FONT>, <FORM>, <H1>, <H2>, <H3>, <H4>, <H5>, <H6>, <HR>, <INPUT>, <LI>, <MENU>, <OBJECT>,

<OL>, <OPTION>, <P>, <PRE>, <SELECT>, <SMALL>, <STRONG>, <SUB>, <SUP>, <TABLE>, <TBODY>, <TD>, <TEXTAREA>,<TH>, <TITLE>, <TR>, <U>, <UL>, <FRAME>, <FRAMESET>, <IMG>, <MAP>, <AREA>.

We also conducted an experiment using a reduced set of tags. For each tag, we calculated a mean descriptor, by computing bin averages over all 36 web pages. We then calculated the deviation of each descriptor from its mean. We only kept those tags with high deviations, as these tags more easily discriminate among the various pages. The tags we kept for this experiment were <FONT>, <STRONG>, and <IMG>.

In all our experiments, we compared each individual web page, using the nearest neighbour approach, to the 35 remaining pages, using both sets of tags. We tried to determine both individualized categories as well as genre categories. The former takes a match as successful only if the two pages came from the same site, while the latter takes a match as successful only if the two pages came from the same genre: newspaper versus e-commerce. Here is the table of our results.

	Individualized Categories		Genre Categories	
	Matches	Failures	Matches	Failures
52 tags	26	10	33	3
3 tags	27	9	33	3

Based on these initial results, it seems that our technique has promise for genre detection.

6. Conclusions

The aim of this research was to analyze the possibility of categorizing webpages and webpage genre by structure or layout. The original insight comes from the fact that many newspaper sites, say, have the same look and feel. Based on our results, we believe that structure could play an important role, along with textual and visual keywords, in webpage categorization and searching.

Bibliography

[1] R. Cooley, B. Mobasher, and J. Srivastava, 'Web Mining: Information and Pattern Discovery on the World Wide Web,' Proceedings *of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97),* November 1997.

[2] D.W. Embley, Y. Jiang, and Y.K. Ng, 'Record – Boundary Discovery in Web Documents,' *Proceedings of the ACM SIGMOD Conference,* 1999, pp. 467-478.

[3] Y. Tao and W.I. Grosky, 'Spatial Color Indexing Using Rotation, Translation, and Scale Invariant Anglograms,' *Multimedia Tools and Applications,* Volume 15, Number 3 (December 2001), pp. 247-268.

[4] B. Ludäscher, R. Himmeröder, G. Lausen, W. May, and C. Schlepphorst, 'Managing Semistructured Data with FLORID : A Deductive Object-Oriented Perspective,' *Information Systems,* Volume 23, Number 8 (1998), pp. 589-613.

[5] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J.L. Wiener, 'The Lorel Query Language for Semistructured Data,' *International Journal on Digital Libraries,* Volume 1, Number 1 (April 1997), pp. 68-88.

[6] A. Deutsch, M. Fernandez, and D. Suciu, 'Storing Semistructured data in Relations,' *Proceedings of the Workshop on Query processing for Semistructured data and Non-standard Data Formats,* Jerusalem, Israel, January, 1999.

[7] N. Ashish and C. Knobolk, 'Wrapper Generation for Semi-Structured Internet Sources,' SIGMOD Record, Volume 26, Number 4 (December 1997), pp. 8-15.

Authors' Information

William I. Grosky – University of Michigan-Dearborn, Computer and Information Science Department, 4901 Evergreen Road, Dearborn, Michigan 48128, USA; email: wgrosky@umich.edu

Gargee Deshpande – Wayne State University, Computer Science Department, Detroit, Michigan 48202, USA