Heading-Based Sectional Hierarchy Identification for HTML Documents

F. Canan Pembe[1,2] and Tunga Güngör[1]
[1]Dept. of Computer Engineering,
Boğaziçi University,
Bebek, İstanbul, 34342, Turkey
[2]Dept. of Computer Engineering,
İstanbul Kültür University,
Ataköy, İstanbul, 34156, Turkey

***Abstract*-Most of the documents found on the Web are prepared in HTML format which was basically designed for presentation of data. As a result, some limitations are encountered when these documents are accessed automatically for a semantic interpretation of their content. One such inadequacy is in representing the sectional hierarchy (i.e. sections and subsections) of these documents and the headings in this hierarchy. Automatically obtaining this information is a difficult task due to the underlying format and the cluttered structure encountered in most of the Web pages. In this paper, we propose a novel approach to extract heading-based sectional hierarchies of HTML documents. This is the first part of the research, where we aim to use this information in automatic summaries to improve Web search experience of Internet users.**

I. INTRODUCTION

Documents on the Web are generally prepared for visual access and browsing of users. However, they are also increasingly accessed and processed automatically for information retrieval and extraction purposes, for example by search engines. This increases the need for semantically exploiting Web document structure and content. Traditionally, Web documents are prepared in HTML format whose primary purpose is presentation of data which brings limitations when a more semantic analysis of document content is desired. For this purpose, semantic markup languages such as XML have been developed. However, HTML documents still dominate the Web. Therefore, better methods for processing HTML documents are needed.

In this paper, we address the problem of heading-based sectional hierarchy identification for HTML documents. In general, the structure of a document may be considered as a hierarchy where each document may have sections; each section may have subsections and so on, together with corresponding headings and subheadings. The variety in the structure and content of Web documents and the use of HTML format, which aims the presentation rather than semantics of data, make this analysis a difficult task. In the proposed system, we use a rule-based approach with HTML DOM (Document Object Model) [1] tree analysis in identifying the sections and subsections of documents together with the corresponding headings.

The heading-based sectional hierarchy of a Web document can be used for several purposes, including automatic summarization. Our research focuses on building effective summaries in order to improve Web search. Our aim is to incorporate the sectional structure and the headings in different levels into the output summaries. We suggest that using such summaries in the results of search engines can make the identification of relevant and irrelevant documents easier for the users than the traditional short extracts of documents by providing the context of searched terms in the documents. This issue is further discussed in Section VI.

The rest of this paper is organized as follows. First, related work is given in Section II. This is followed by the problem description in Section III and the proposed solution in Section IV. An evaluation of the method is given in Section V. Finally, further work and conclusions are presented in sections VI and VII.

II. RELATED WORK

Structural and semantic analysis of HTML documents is a rather young field of research. One of the motivations in this area is to filter important content from Web pages by eliminating ads and other cluttered parts which are very common to Web pages [2]. Another motivation is to convert HTML documents into semantically-rich XML documents to be utilized later [3]. This analysis may also be used for obtaining a hierarchical structure for the document including its sections and subsections [4, 5, 6, 7, 8, 9]. Some of this work is motivated by the need of displaying content in small-screen devices such as PDAs [7, 8], while others leave the usage open, including more intelligent retrieval of information, summarization, etc.

Most of the related work concentrates on exploiting HTML tags for the analysis; some of them do the analysis by building the explicit DOM tree [2, 3, 5, 7, 9]. The approaches used are mostly either rule-based [3, 5] or machine learning based [9]. Moreover, some of them target a certain domain such as resume documents [3], whereas others are domain-independent.

In this work, we employ a rule-based approach using DOM tree analysis with no domain restriction. Our work is close to [5] where also hierarchical semantic structures of documents are created. However, our work differs in that we concentrate on section and subsection headings and make use of these in building the hierarchy. Moreover, our work is based on a more robust DOM tree analysis than the string matching algorithm used in that work for the paths in the document DOM tree which can easily fail due to irregularities in HTML documents.

1-4244-1364-8/07/$25.00 ©2007 IEEE

In [7], semantic partitions of documents are determined and labeled using domain ontologies. In [9], a single title (i.e. the main title) for each document is obtained using DOM tree analysis and its effect on information retrieval performance is evaluated. In our work, the hierarchy of headings for each document is obtained. To the best of our knowledge, there is no work targeting heading-based hierarchy extraction from HTML documents using DOM tree analysis.

III. Problem

We consider the problem of automatically creating sectional hierarchy of a given HTML document based on its identified headings. The targeted HTML documents are general, with no domain restriction.

Web documents are typically heterogeneous, containing images, text in different formats, interactive forms, etc. Their content may also be diverse with sections on different topics, ads, etc. We consider textual parts of the documents. In Figure 1, a typical HTML document is given. In Figure 2, part of the sectional hierarchy for the document in Figure 1 is shown. As can be seen, headings in different levels can be identified as a hierarchy together with the sentences under the headings.

Figure 1. An example Web site.

The heterogeneity of Web documents and the underlying HTML format, whose primary purpose is the presentation of data, make it difficult to process HTML documents for a semantic analysis aimed at heading hierarchy identification. Actually, there are heading tags in HTML for different levels of headings: <h1> through <h6>. However, in most of the pages found on the Web, either they are not used or they are used inconsistently for headings in different levels. Sometimes, these tags are even used for non-heading text just for formatting purposes. Instead of using the heading tags, in most of the HTML documents, the headings are distinguished by formatting them in a way different from their surrounding text, e.g. font size, color, boldness etc.

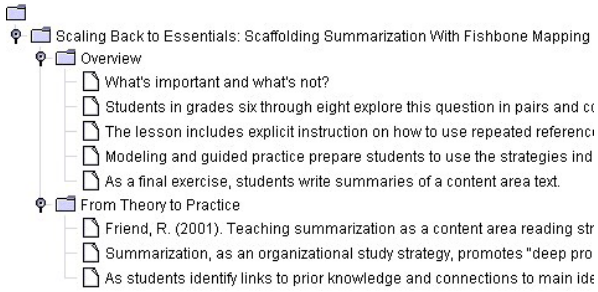

Figure 2. Part of the sectional hierarchy for the example Web site.

Another difficulty in structural analysis of Web documents is the complex organization found in most of them. A typical Web document can contain multiple columns and blocks of text each on different topics and even with different formatting. This organization is usually achieved by the use of <table> and related tags in HTML to group blocks of content. For successful identification of sectional hierarchy, this organization should be taken into account. For this purpose, HTML DOM tree of the document can be utilized which is also our strategy in this work.

IV. Sectional Hierarchy Identification

Our method for sectional hierarchy identification consists of three different steps. In the first step, the DOM tree is processed to obtain blocks of elements in the document. In the second step, possible headings are identified using heuristics. Then, in the third step, the hierarchy is restructured based on the identified headings. The details of the process are given in the following subsections.

A. DOM Tree Processing

The DOM tree of the document is given as input to this step. Here, HTML tags are considered as belonging into one of the two groups: container tags (e.g. <table>, <td>, <tr>, etc) which can contain other HTML tags or text and format tags (e.g. <b>, <font>, <h1>, <h2>, etc) which are usually concerned with the formatting of the text. Then, the DOM tree is traversed breadth-first and converted to a tree with only containment relations between elements using container tags; e.g. <table>. The leaf nodes of the tree contain text parts of the document. During this conversion, sentence boundaries are also taken into account such that each leaf corresponds to a sentence. Format tags such as <font> are passed as features to the text elements. The obtained tree is also a simplified version of the original tree. The assumption is that semantically related parts usually occur in the same or neighbor container tags.

B. Heading Identification

The aim of this step is to identify possible headings within the document. Also, heading-like text which is not actually heading is eliminated based on the contextual information. This is a difficult task especially for Web documents which are typically cluttered with text in various formats. The heuristics employed include:

- Headings do not end with '.', '!', ',', etc. Headings do not start with '(', etc.
- Headings start and end with new line. That is, headings are not in the middle of a paragraph. For this purpose, start and end of each text block is identified whenever certain tags, such as
, <p>, <td>, <li>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6> and <tr> are encountered.
- Menus (usually hyperlinks) at the beginning and end of the document, which are commonly used in Web pages, are not headings.
- A heading which has smaller font is not followed by a heading with larger font (according to heading hierarchy).
- A heading is not followed with text in the same format.
- A heading which is not bold is not followed by bold text.
- A heading is limited in length (i.e., the number of characters).
- Headings are not aligned to the right.
- Heading-like text with no following content is eliminated.
- Text fragments containing certain phrases (e.g. "click here", "skip navigation", etc) are not headings.
- Text fragments in <select> tags are not headings.

Once the headings are identified, their formats, obtained from the HTML tags, are also stored to be used in the next step. The features that are used to distinguish different levels of headings are given in Table I.

TABLE I
FEATURES USED FOR IDENTIFYING FORMAT OF HEADINGS

Feature	Description
h1	<h1>, level-1 heading
h2	<h2>, level-2 heading
h3	<h3>, level-3 heading
h4	<h4>, level-4 heading
h5	<h5>, level-5 heading
h6	<h6>, level-6 heading
B	<b>, bold
strong	
em	, emphasis
A	<a>, hyperlink
U	<u>, underlined
I	<i>, italic
f_size	<font size=...>, font size
f_color	<font color=...> font color
f_face	<font face=...> font face
allUpperCase	all the letters in uppercase
cssId	CSS id attribute if used
cssClass	CSS class attribute if used
li	, different levels of lists

At the end of the first two steps, we have the document tree portion in Figure 3 for the example document in Figure 1. Here, headings identified based on the heuristics are also shown underlined.

C. *Hierarchy Restructuring*

In this step, the obtained document hierarchy is rearranged in order to adjust the hierarchy based on the identified headings. The restructuring works from bottom-up in the document tree; that is, first smaller blocks of text (deeper in the hierarchy) are restructured according to the headings. The algorithm is based on the idea that, in a semantic block of text, the headings in the same level usually have the same format. Then, headings in different levels can be identified using their format. For example, a heading with format "bold, font size: 3" belongs to a different level in the hierarchy than a heading with format "underlined, font size: 2, all letters uppercase". The document tree is rearranged based on this hierarchy. The headings and text are connected as child nodes to the headings in higher levels. The algorithm to restructure a given block within the document tree is summarized in the following.

```
Algorithm RestructureTree(p)
input
    p: parent node of the nodes within the block
begin
1:  Remove all the children of p to a list L
2:  textAppendPoint = p
3:  headingAppendPoint = p
4:  for each node n in L
5:      if (n is not a heading)
6:          Append text as child to textAppendPoint
7:      else
8:          Check headingFormats list
9:          if (there is no entry for that format)
10:             Add the new heading format to headingFormats
                list as next level in the hierarchy.
11:         end if
12:         Update headingAppendPoint
13:         Append heading as child to headingAppendPoint
14:         Update textAppendPoint
15:     end if
16: end for
end
```

After the application of the restructuring step on the subtree of Figure 3, the subtree in Figure 4 is obtained. As can be seen, the headings "Overview" and "From Theory…" are correctly shown under the higher-level heading "Scaling Back…". Also, the sentences are correctly shown under the corresponding headings.

We implemented the proposed method of hierarchy identification using GATE framework for text engineering [10, 11] as the underlying development environment, which is an open source project using component-based technology in Java. We made use of HTML processing capabilities, English Tokeniser and Sentence Splitter modules of GATE and added our own modules for HTML document structure analysis.

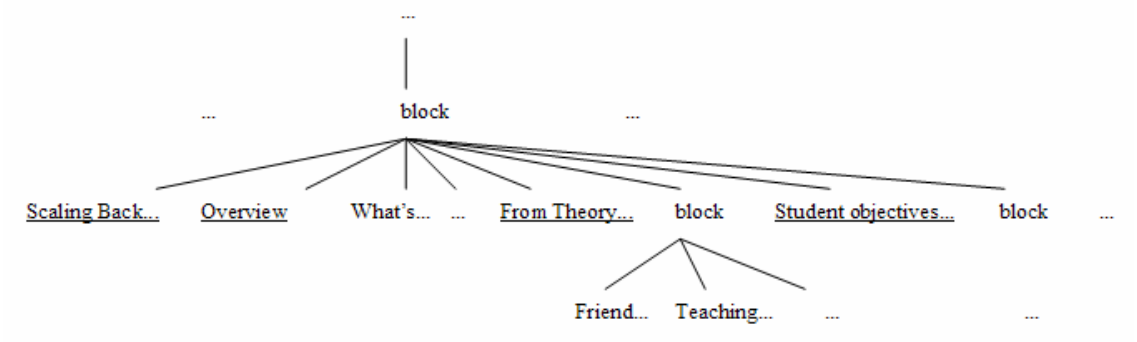

Figure 3. Part of the document tree (identified headings are underlined).

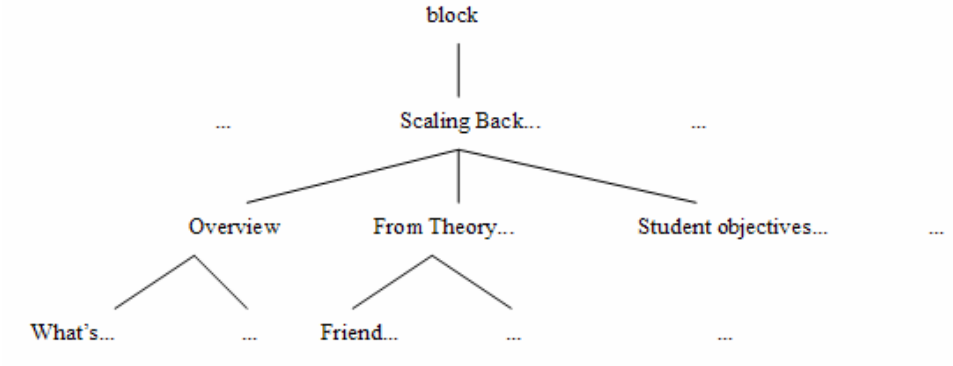

Figure 4. Part of DOM tree after restructuring.

V. EVALUATION

For the evaluation, five queries from TREC-9 [12] (see Table II) were used to build a collection of 50 documents. The document sets were obtained from the top results returned by Google [13] in response to the queries. Then, for each document, the actual sectional hierarchy is manually determined in order to be used in the evaluation.

TABLE II
QUERIES USED TO RETRIEVE DOCUMENTS FOR EVALUATION

Document Set	Query Keywords
1	hunger
2	parkinson's disease
3	Mexican food culture
4	how e-mail benefits businesses
5	Calcium

We have run our hierarchy extraction method on these documents. In Table III, the average depths of the DOM trees prior to the processing and the average depths of the obtained hierarchies are given for each document set. Also, the hierarchical structure obtained for each document is compared with the actual hierarchical relationships manually marked. To evaluate the accuracy of the proposed method, the number of correct parent-child relationships in the obtained hierarchy is counted and its ratio to the total number of parent-child relationships is computed. We obtained 78% average accuracy for the heading-based sectional hierarchy identification (Table III).

TABLE III
RESULTS RELATED TO THE HIEARCHY IDENTIFICATION

Document Set	DOM Tree Depth	Hierarchy Depth	Hierarchy Accuracy
1	17,20	3,50	0,89
2	14,00	4,70	0,75
3	15,00	5,80	0,77
4	16,60	7,40	0,68
5	11,50	5,10	0,82
Overall	14,86	5,30	0,78

In Table IV, results related to the heading identification part are given. First, the averages for the actual number of headings in the documents are given. Then, recall (R), precision (P) and f-measure (F) values for the heading identification experiment are presented. Recall is calculated as the ratio of the number of headings correctly identified to the number of actual headings. Precision is calculated as the ratio of the correctly identified headings to the number of headings identified. F-measure is a combined measure of recall and precision.

TABLE IV
RESULTS FOR HEADING IDENTIFICATION

Document Set	Headings Number	R	P	F
1	5,30	0,91	0,78	0,84
2	7,10	0,93	0,70	0,80
3	7,20	0,89	0,68	0,77
4	7,40	0,84	0,54	0,66
5	16,20	0,91	0,91	0,91
Overall	8,64	0,90	0,72	0,80

Then, we considered the reasons of the errors in heading and hierarchy identification experiment. Most of the errors were caused by cluttered structure commonly encountered in Web pages. Especially, heading-like text, e.g. text with no punctuations and with distinct formatting than the surrounding text may be wrongly identified as headings to other parts. Also, headings which are part of images can cause problems. Some errors in hierarchy identification are caused by inconsistent table usage for presentation purposes rather than combining semantically related blocks. Inconsistent format usage for different levels of headings also causes problems in determining the heading hierarchy.

The accuracy of the heading-based sectional hierarchy identification experiment is also shown in Figure 5. As can be seen, for majority of the documents, an acceptable accuracy is obtained. Compared to the related work in [9] where a single title is obtained for each document, our work concentrates on extracting all the headings together with their level in the hierarchy. The accuracy obtained in [9] range from 0,698 to 0,909. We believe that the heading hierarchy identification is a much more difficult problem where we obtained 0,78 accuracy.

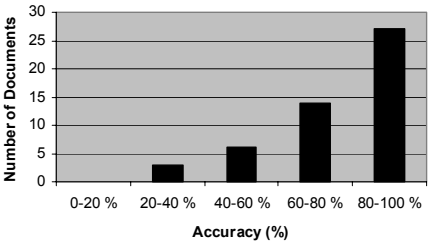

Figure 5. Accuracy in hierarchy identification.

VI. FURTHER WORK

As a future work, the heading-based sectional hierarchy obtained in this research will be incorporated into automatic summaries of documents for Web search tasks. In the proposed system, the summaries provided in search engine results will be longer. For this purpose, similar to a previous work in the literature [14], only the titles of each link will be listed, and in a separate frame, the summary for that document will be displayed when the user moves the mouse on a particular link.

An example summary is given in Figure 6 for demonstration. The summary is organized to reflect the sectional hierarchy. Also, consecutive fragments and headings are separated with dots (...) indicating that more material follows in between. As can be seen, the structure of the actual document as well as its coverage, main theme, size, etc. are much more explicit compared with two-line extracts provided by current search engines. In this way, it is expected that the user can judge the relevancy of each document better than the traditional approaches without the necessity to load the actual page.

VII. CONCLUSION

In this paper, we considered the rather unexplored problem of heading-based sectional hierarchy identification for HTML documents which dominate the Web. We proposed a novel method to solve this problem using a rule based approach and DOM tree analysis. We tested the efficiency of our method on a set of 50 documents both for successful hierarchy and heading identification with metrics suitably designed for this purpose. The results show that our algorithm obtains acceptable results for heading-based hierarchy identification for un-restricted domain of Web documents.

We also proposed an application of the head-ing-based hierarchy of HTML documents; namely the summarization of these documents for Web search tasks. We believe that, such summaries can be very helpful to Web users in assessing the relevancy of the documents in the results of the search engines in response to their queries. As future work, we will evaluate the effectiveness of such summaries in Web search tasks.

REFERENCES

[1] Document Object Model (DOM), 2005. http://www.w3.org/DOM/.
[2] S. Gupta, G. E. Kaiser, P. Grimm, M. F. Chiang, and J. Starren, "Automating content extraction of HTML documents," *World Wide Web*, vol. 8, 2005, pp. 179 – 224.
[3] C. Y. Chung, M. Gertz, and N. Sundarsan, "Reverse engineering for Web data: From visual to semantic structures," *Proceedings of the 18th International Conference on Data Engineering*, 2002.
[4] Y. Yang and H.-J. Zhang, "HTML page analysis based on visual cues," *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, 2001.
[5] S. Mukherjee, G. Yang, W. Tan, and I. V. Ramakrishnan, "Automatic discovery of semantic structures in HTML documents," *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 2003.
[6] S. Mukherjee, G. Yang, and I. V. Ramakrishnan, "Automatic annotation of content-rich HTML documents: Structural and semantic analysis," *Proceedings of the International Semantic Web Conference*, 2003.
[7] Y. Chen, W.-Y. Ma, and H.-J. Zhang, "Detecting Web page structure for adaptive viewing on small form factor devices," *Proceedings of the 12th International World Wide Web Conference*, 2003.
[8] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke, "Accordion summarization for end-game browsing on PDAs and cellular phones," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2001
[9] Y. Xue, Y. Hu, G. Xin, R. Song, S. Shi, Y. Cao, C.-Y. Lin, and H. Li., "Web page title extraction and its application," *Information Processing and Management*, vol. 43, 2007, pp. 1332-1347.
[10] GATE, A General Architecture for Text Engineering, 2007. http://gate.ac.uk/.
[11] D. Maynard, K. Bontcheva, H. Saggion, H. Cunningham, and O. Hamza, "Using a text engineering framework to build an extendable and portable IE-based summarisation system," *Proceedings of the ACL Workshop on Text Summarisation*, 2002.
[12] TREC, 2004. http://www.trec.org.
[13] Google, 2007. http://www.google.com.

[14] R. W. White, J. M. Jose, and I. Ruthven, "A task-oriented study on the influencing effects of query-biased summarization in Web searching," Information Processing and Management, vol. 39, 2003, pp. 707-733.

Scaling Back to Essentials: Scaffolding Summarization With Fishbone Mapping

...

Overview

...

The lesson includes explicit instruction on how to use repeated references as a strategy for determining important information in a text and how to generalize main ideas from related details.

...

From Theory to Practice

...

Summarization, as an organizational study strategy, promotes "deep processing."

...

Student Objectives

...

Apply the repeated reference cue strategy in cooperative learning groups to identify important information in a text

...

Derive generalizations by examining clusters of text details for commonalities

...

Instructional Plan

...

Preparation

...

Complete a fishbone map in advance to serve as practice for the process.

...

Instruction and Activities

...

To identify main ideas that are important to the author of a text by recognizing repeated references

...

To use the fishbone map as the basis for writing a summary of the passage in their own words

Figure 6. An example summary of the proposed system.