Document Visualization on Small Displays

Hoi Ka Kit
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
algerhoi@excite.com

Dik-Lun Lee
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
dlee@cs.ust.hk

ABSTRACT

Limited display size and resolution on mobile devices is one of the main obstacles for wide-spread use of web applications in a wireless environment. Web pages are often too large for a PDA (Personal Digital Assistant) screen to present. The same problem exists for devices with low resolution such as WebTV. Manual reconstruction of web pages for these devices would ease the problem; however, the large variation of display capabilities will greatly increase the burden of web page designers since they have to customize a web page for each possible display device. An automatic system is needed to solve the problem.

A Document Segmentation and Presentation System is proposed in this thesis to solve the problem. The system automatically divides a web document into a number of logical segments based on the screen size and the structure and content of the document. Additional information such as overviews and summaries is also extracted to facilitate navigation. The system presents the segments and structural information of a web document to make full use of the screen for information finding.

Keywords

Document Model HTML Transcoding Document Visualization

1. INTRODUCTION

Mobile computing has drawn much attention in these few years due to the advances in technology. However, viewing documents on a mobile device is not an easy task because of the limited display size, CPU power and bandwidth. This is an obstacle for wireless web browsing as there is no automatic optimal transformation of HTML documents designed for display on large screens to small PDA screens. [1]

The same problem appears in devices for web access (or web-like access) on television. Although the size of TVs generally is larger than most of the ordinary computer monitors, the relatively low resolution and long viewing distance reduce the amount of information they can display. In this thesis, we use the term "small displays" to refer to devices which can only display a small amount of information for comfortable user viewing, either because the screens are physically small or have very low resolution.

Small displays increase the burden of web page authors and application designers. They need to consider all the possibilities of the display ability of user's devices when the web pages are created.

One of the solutions to the problem is to use a scroll bar to navigate a web page when the page is too large for the screen. Obviously, this approach requires many user interactions before the desired information can be reached. Another approach is to break down the web page into small segments in a top-down left-right order and display them one by one. However, users would still have difficulties in locating relevant information if no prior knowledge of the web page is known.

An automatic document segmentation and presentation system (DSPS) is presented to solve the problem. The system has three primary functions. Firstly, it automatically divides a web document into different logical segments based on the display size of the devices, and the hierarchical structure and content of the documents. Secondly, it extracts the summary and overview information from the logical segments to help users locate relevant information. Thirdly, an interface for clear and user-friendly presentation of the segments is created for rapid access to the desired information.

2. HEURISTIC CONVERSION TO CONTENT TREE

In order to obtain high-level structural meaning from an HTML document, a sequence of algorithms and heuristic methods is proposed to produce an approximate structured document of the document structure from an HTML document. All the algorithms were built around a document model called Content Tree, which was proposed by Lim and Ng [2, 3].

An HTML document is first converted and normalized into an intermediate Content Tree by an HTML parser. The intermediate CT is then converted into an initial CT using the Extended Linear Dependence algorithm. Then the table refinement algorithm transforms table structures in the HTML document into different subtrees. The Segment Breakdown algorithm further refines the CT structure by analyzing the formatting style of each node. Some irrelevant subtree segments are removed from the CT by the Irrelevant Segment Removal algorithm. The resulting CT is then transformed into a form that can be used for visualization on the screen. Fig. 1 shows the overview of the sequence of algorithms used to approximate an HTML document into a CT.

2.1 Extended Linear Dependence Algorithm

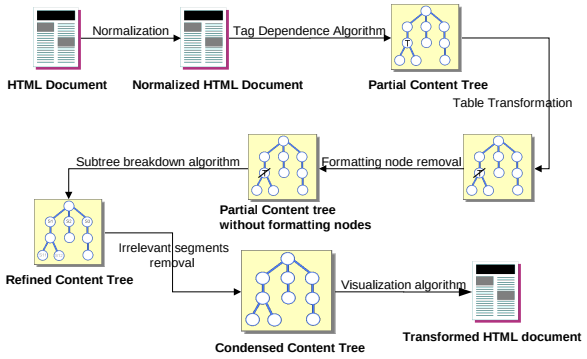

Figure 1: Algorithm overview of the heuristic conversion from HTML document to Content Tree

The Extended Linear Dependence algorithm is based on the construction of CT proposed by Lim and Ng [2, 3]. The original construction of CT is based on the tag object dependence. The linear dependence defines a sequence of the tag objects (`H1 > H2 > H3 .....`) which indicates the conceptual, structural and scope meaning of the tags. Any access path from the root node to any leaf node of the CT should follow the linear dependence.

However, there is a serious limitation of the Lim and Ng's algorithm when it is applied to real-world web pages. Their algorithm only considers some simple HTML tags like `H1`, `TITLE` and `BODY`. Tables and frames, which are heavily used in formatting the web pages, are excluded.

In order to cope with the problem, Lim and Ng's algorithm has been extended to create an initial CT for a real-world web page. The main element of the extension is to use different strategies for different types of tags. Different algorithms are applied to the structural, block and formatting tags.

2.2 Heuristic Transformation

After performing the Extended Linear Dependence algorithm and Table Transformation algorithm, the resulting CT captures most of the structural information provided by the structural tags and table structures. However, not every web page authors or web page authoring tools use structural tags or table structures to indicate the structural information of the web pages.

If the HTML fragment does not contain structural tags or tables, heuristic can be applied to fill up the gap. In the following section, the use of formatting styles of the web pages to extract hidden structural meanings from the web pages is discussed. The idea behind the heuristic is that if a small piece of text contains a lot of formatting, it is more important and structurally meaningful than a plain text is.

Different formatting tags have different degrees of emphasizing effects. For example, `B` has a stronger emphasizing effect than `I` and `SMALL`. Each formatting tag is assigned with a value, a higher value means a higher degree of emphasizing effect.

Formatting Nodes Removal algorithm removes all the formatting nodes by using a numeric value to replace a collection of formatting nodes. The numeric value reflects how heavy the text is formatted. The numeric value is calculated by summing up all the formatting values of the formatting tags applied to the text.

Once the format values is calculated, the Segment Breakdown algorithm is used to segment an HTML segment by identifying titles within the segment and using them as the cutting points to create sub-segments. The Segment Breakdown algorithm can be applied to each HTML segment until the size of a sub-segment is small enough for display.

The Segment Breakdown algorithm identifies the highly formatted text nodes and promote them to a higher hierarchical level of the CT. Less formatted texts then follow the highly formatted texts as dependent nodes.

The Segment Breakdown algorithm can break a flat tree structure into a hierarchical one by analyzing the formatting styles. This operation is essential in DSPS for visualizing the document on small display devices as the document fragment represented by the flat tree structure can be replaced by a number of logically separated segments. Also, an overview, which consists of highly formatted texts, of these segments is also generated.

3. CONCLUSION

The Document Segmentation and Presentation System is proposed to transform an HTML document into a data structure called a "Content Tree", which represents the logical structure of the HTML document. The Content Tree structure is used to provide various views that are suitable for display on various small displays. The DSPS is targeted for real world commercial HTML documents that contain many screen-oriented formatting styles like heavily nested tables and complicated text formatting structures.

The Content Tree structure facilitates both information extraction and document visualization. As each subtree in the Content Tree is a logical segment of the document, logical segments can be extracted by identifying the corresponding subtrees while irrelevant segments can be removed by purging the subtrees from the Content Tree.

The central part of the DSPS is the heuristic approach for approximating an HTML document into a structured document. The heuristic approach incorporates the information provided by the HTML tags and the formatting styles to accomplish the transformation.

4. REFERENCES

[1] D. Raggett. HTML 3.2 Reference Specification. World-Wide Web Consortium, 1997. http://www.w3c.org/TR/REC-html32.

[2] Seung Jin Lim and Yiu Kai Ng. Constructing Hierarchical Information Structures of Sub-Page Level HTML Documents. *Proceedings of the 5th International Conference of Foundations of Data Organization (FODO'98), Kobe, Japan*, 1998, 66-75.

[3] Seung Jin Lim and Yiu Kai Ng. Extracting Structures of HTML Documents. *Proceedings of the 12th International Conference on Information Networking (ICOIN-12), Tokyo, Japan*, 1998, 420-426.