Detecting Web Page Structure for Adaptive Viewing on Small From Factor Devices

Authors: Yu Chen, Wai-Ying Ma, Hong-Jing Zhang
Microsoft Research Asia

Paper Code: UA-2
Presenter: Thong Chanchem

Problems

* Most web page today have been designed for desktop computer, so it is often too large to fit into the small screen of a mobile device.
* The user have to manually scroll the window to find the content of interest and position the window properly for reading information.
* This is a time consuming process.

Goal and Solution

Goal: to find a better way to enable easy navigation and browsing of a large web page on a small form factor device.

Solution: Two-level hierarchy with a shrinkable representation

Their Approach

Two Process
1. Page Analysis - analyze the structure of a given web page.
2. Page Splitting - split the web page into a two-level hierarchy.

Page Analysis

Page Analysis: to extract the semantic structure of an existing web page.

Page Analysis:
- Content Structure Extracted by the web server.
- High-Level Content Block Detection.
- Subheader Blocks.
- Detection of Header and Footer.
- Detection of left- and right-side bars.
- Explicit Separator Detection.
- Implicit Separator Detection.
Content Structure Embedded by the web author

To detect the high-level content blocks (headers and footers, side bars and body), and split the content blocks:

1. **High-Level Content Block Detection**

   - Explicit separators are created using content HTML tags.
   - Implicit separators are created by having a blank space between content in a web page.

   ![High-Level Content Block Detection diagram](image1)

   **Example of HTML DOM tree**

   Selecting appropriate nodes and classifying them into one of the five high-level content blocks:

   - Head
   - Body
   - Side
   - Footer
   - Others...

   ![Example of HTML DOM tree](image2)

   **Dynamic threshold for header and footer detection**

   \[ N = \text{head\_threshold} + \text{footer\_threshold} \]

   \[ \text{head\_threshold} = a, \text{footer\_threshold} = b, a \text{ and } b \text{ are constants.} \]

   From their experiment, the best performance can be achieved by setting:

   \[ \text{head\_threshold} = 40 \text{ and } b = 20 \text{ and } c = 5 \]

   ![Dynamic threshold for header and footer detection](image3)

2. **High-Level Content Block Detection (cont.)**

   1. **Detection of Left and Right Side Bar**

   In their experiments, they define the left 1/4 part of the web page to be the left side bar region, and right 1/4 part to be the right side bar region.

   ![High-Level Content Block Detection (cont.)](image4)

   The result of high-level content block detection on the Microsoft Home page

3. **Explicit Separator Detection**

   Explicit separators can be detected by analyzing the properties of the tags.

   The following three types of explicit separators:

   1. `<HR>` represents a horizontal line
   2. `<TABLE>` and `<TD>` and `<TH>` have border properties.
   3. Using an image.

   ![Explicit Separator Detection](image5)
Implicit Separator Detection

Implicit separators are blank areas created intentionally by the author to separate content.

Detecting implicit separators

Implicit Separator Detection (cont.)

The pattern recognition algorithm is used to produce basic blocks for implicit separator detection.

Page Adaptation

These are two methods for splitting a web page:
- Single-subject splitting
- Multi-subject splitting

(a) Single-subject splitting breaks the whole web page into several sub-pages and connects them with user/hack hyperlinks.
(b) Multi-subject splitting generates a new index page in addition to sub-pages. The index page contains hypertext pointing to each sub-page.

Page Adaptation (Cont.)

1. Page Splitting and Sub-page Generation

To solve the problem of style and hyperlink:
- Dealing with Style
  Problem: The sub-page may lose some style information.
  Solution: If the "header" section of a web page into each generated sub-page.
  Example:

```
<head>
  <style type="text/css">";
  #link {color: blue}"
</head>
```

Page Adaptation (Cont.)

- Dealing with Internal Hyperlinks
  Problem: The sub-page loses internal hyperlinks.
  Solution: Change the pointer to sub-page to the proper target sub-page.
  Example:

```
<sub-page1.htm>
  ...<sub-page1.htm>
  ...
  ...<sub-page2.htm>
  ...
```

Page Adaptation (Cont.)

- Dealing with Relative Hyperlink Resolution
  Problem: The sub-page loses absolute hyperlinks or relative hyperlinks.
  Solution: Copy the "header" section into each sub-page.
  Example:

```
Assume that the absolute address for an image is:
Image src="http://www abc.com/images/abc.gif"
```

Specify a base URL for all of the links in a page:
```
<head>
  <base href="http://www abc.com/"></base>
  ...
</head>
```

The relative address:
```
<img src="abc.gif"/>
```
Page Adaptation (Cont.)

2. Index Page Generation
An index page which contains a subhead and hypertext in its sub-pager.

An example of the index page and sub-pages generated from the homepage of MBN news.

Experiment and Results
They selected 30 popular web sites and 20 typical sub-pager as their test data.

<table>
<thead>
<tr>
<th>Page Analysis</th>
<th>Page Splitting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfect</td>
<td>Perfect</td>
</tr>
<tr>
<td>Good</td>
<td>Error</td>
</tr>
<tr>
<td>Error</td>
<td>Error</td>
</tr>
</tbody>
</table>

Problem cause by absolute positioning
Most of the splitting errors in the category of "Good" are related to scripts (including CSS), absolute positioning, or scripts used to display dynamic content.

Processing time for page adaptation algorithm

Auto-positioning instead of splitting
For a sub-page that uses scripts extensively, it is harder not to split it.

Advantage / Disadvantage

Advantage
- Reduce client loading time.
- Reduce the consumption of network bandwidth.
- Reduce the client computation.
- Don’t have to re-design web page for specific client device.
- Don’t destroy the structure of source page.

Disadvantage
- Increase the computation at server side.
Critique

• The paper is well organized.
• The author provided a lot of examples and explanations.
• Some pictures are hard to see in detail.
• There is a lot of information in the top level that is provided in a global view and index to sub-pages. Therefore, it may be difficult for user to find the content of interest in small screen devices.

Quiz

- Why can’t we keep the node <CENTER> in the Yahoo! Homepage as a whole? (see 3.2.1)
- What are the differences between explicit and implicit separator detection?
- What is the difference between Single-subject and Multi-subject splitting?
- What is the problem that causes by CSS in “Page splitting and sub-page generation”? and How to solve it?
- When do we use auto-positioning instead of splitting?