Mining Contents in Web Page Using Cosine Similarity

Swe Swe Nyein
University of Computer Studies, Mandalay
Mandalay, Myanmar
sweswenyein@gmail.com

Abstract— **Web pages typically contain a large amount of information that is not part of the main contents of the pages, e.g.; banner ads, navigation bars, copy right and privacy notices, advertisements which are not related to the main content (relevant information). In this paper, an algorithm is proposed that extract the main content from the web documents. The algorithm based on Content Structure Tree (CST). Firstly, the proposed system use HTML Parser to construct DOM (Document Object Model) tree from which construct Content Structure Tree (CST) which can easily separate the main content blocks from the other blocks. The proposed system then introduce cosine similarity measure to evaluate which parts of the CST tree represent the less important and which parts represent the more important of the page. The proposed system can define the ranking of the documents using similarity values and also extracts the top ranked documents as more relevant to the query.**

Keywords- DOM tree, CST tree, Cosine Similarity

I. INTRODUCTION

The rapid expansion of the Internet has made Web a popular place for disseminating and collecting information. Apart from the useful information on the web, it usually has such information as navigation panels, copyright notices, banner ads, etc. Although these information item are useful for human viewers and necessary for the Web site owners, they can seriously harm automated information collection and Web data mining, e.g. Web page clustering, Web page classification, and information retrieval. So how to extract the main content blocks become very important. Web pages contain Div block, Table block or other HTML blocks. In this paper, we focus on extracting the relevant documents to the user query. We proposed a new algorithm to extract the informative block from web page based on DOM-based analysis and Content Structure Tree (CST). Then we apply cosine similarity measure to identify and separate the rank of the each block from the page. We also use TF-IDF (Term Frequency and Inverse Document Frequency) scheme for calculating the weight of each node on the CST. Finally, we extract the most relevant information to the query.

In the following section of the paper, we first describe related studies in section 2. Then we illustrate our proposed architecture for extracting the relevant documents from the web page in section 3. We introduce the Document Object Model (DOM) tree in section 4, Content Structure Tree (CST) in section 5 and content estimation in section 6. Finally, we provide conclusion and future work.

978-1-61284-840-2/11/$26.00 ©2011 IEEE

II. RELATED WORK

From time to time, many extraction systems have been developed. In [1], C. Li et al. propose a method to extract informative block from a web page based on the analysis of both the layouts and the semantic information of the web pages. They needed to identify blocks occurring in a web collection based on the Vision-based Page Segmentation algorithm. In [2], L. Yi et al. propose a new tree structure, called Style Tree to capture the actual contents and the common layouts (or presentation styles) of the Web pages in a Web site. Their method can difficult to capture the common presentation for many web pages from different web sites. In [3], Y. Fu et al. propose a method to discover informative content block based on DOM tree. They removed clutters using XPath. They could remove only the web pages with similar layout. In [4], P. S. Hiremath et al. propose an algorithm called VSAP (Visual Structure based Analysis of web Pages) to exact the data region based on the visual clue (location of data region / data records / data items / on the screen at which tag are rendered) information of web pages. In [5] S. H. Lin et al. propose a system, InfoDiscoverer to discover informative content blocks from web documents. It first partitions a web page into several content blocks according to HTML tag <TABLE>. In [6] D. Cai et al. propose a Vision-based Page Segmentation (VIPS) algorithm that segments web pages using DOM tree with a combination of human visual cues, including tag cue, color cue, size cue, and others. In [7], P. M. Joshi propose an approach of combination of HTML DOM analysis and Natural Language Processing (NLP) techniques for automated extractions of main article with associated images form web pages. Their approach did not require prior knowledge of website templates and also extracted not only the text but also associated images based on semantic similarity of image captions to the main text. In [8], Y. Li et al. propose a tree called content structure tree which captured the importance of the blocks. In [9], R, R, Mehta propose a page segmentation algorithm which used both visual and content information to obtain semantically meaningful blocks. The output of the algorithm was a semantic structure tree. In [10], S. Gupta proposes content extraction technique that could remove clutter without destroying webpage layout. It is not only extract information from large logical units but also manipulate smaller units such as specific links within the structure of the DOM tree. Most of the existing approaches based on only DOM tree.

III. PROPOSED ARCHITECTURE

The proposed approach is based on the analysis of actual contents of the Web pages in a given Web site. A web page usually contains main content blocks and noise content blocks. Only the main content blocks represent the informative part that is really we want to know. Thus, in our first task, we will use the HTML Parser to create a DOM tree representation of the original html source. Second, our task is to find relevant information from the web pages in the site. So, we construct a Content Structure Tree (CST) based on the DOM tree. Then we will use cosine similarity method to evaluate each node in the content structure tree and it can easily get the informative block which we want to know. At last, the proposed architecture will be shown in Figure 1.

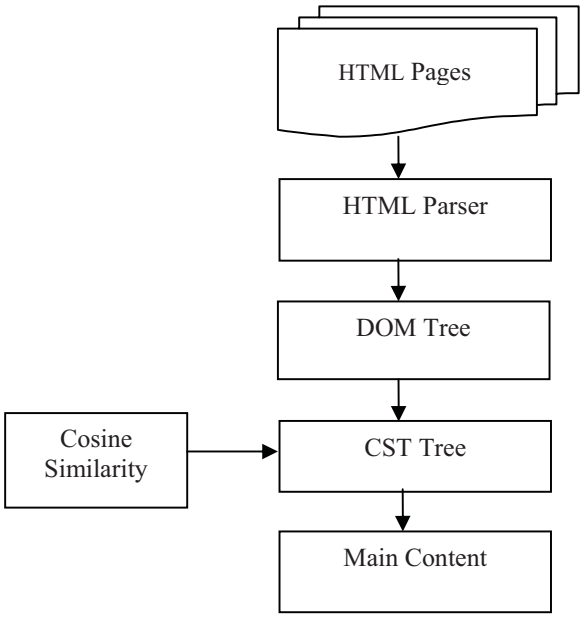

Figure 1. System Architecture

IV. DOM TREE

We will use open source HTML Parser that builds a DOM tree from a page using its HTML code. HTML documents contain HTML tags and plain text. HTML lets format text, add graphics, create link, input forms, frames and tables, etc. In a DOM tree, tags are internal nodes and the detailed texts, images or hyperlinks are the leaf nodes. Figure 2 shows some html segments and its corresponding DOM tree. In the DOM tree, we need to tidy some unnecessary nodes, such as script, style or other customized nodes. HTML Web pages begin from the BODY tag since all the viewable parts are within the scope of BODY. If we need to extract useful informative block from the web pages, we need a more powerful structure tree called Content Structure Tree (CST). Then we can find which content block is more important in the CST tree in Figure 3.

```
<body>
   <div id = “wrapper”>
      <a href=”#”></a>
      <img src=”img”/ >
      <span>text</span>
  </div>
<img src=”img”/>
  <a href=”#”></a>
<table>
…..
</table>
<span>text</span>
</body>
```

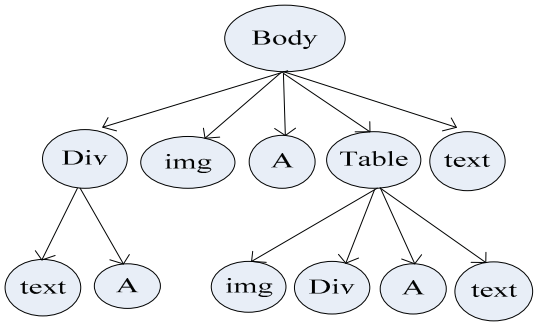

Figure 2. Segment of HTML code and its DOM tree

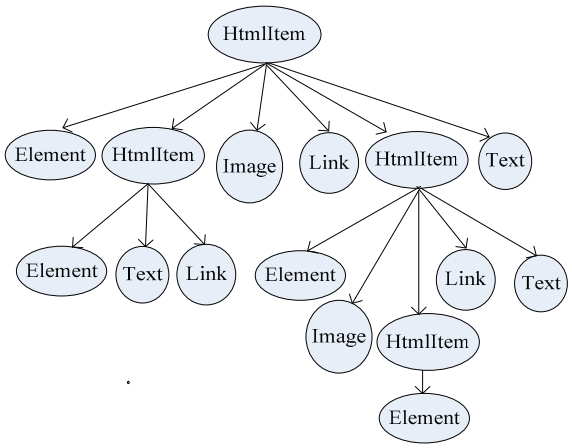

Figure 3. Content structure tree

V. CONTENT STRUCTURE TREE

A Page Content Structure Tree consists of two types of nodes: HtmlItem node and content node that contains text node, image node, link node, etc. An HtmlItem node represents a block which is generated by body tag, div tag and table tag. A content node represents the actual content

of the web page such as text, image and link. The root of the DOM tree is a Body tag node and its corresponding is HtmlItem node in CST tree. We will scan the DOM tree, if we find the Div node, Table node, and then we will construct the new HtmlItem node. The first node of HtmlItem node is Element node which is used to compute the value of its parent HtmlItem node. And then we can compute the similarity with the value of each Element node. Img, A and text nodes in DOM tree will be transformed to Image, Link and Text nodes and be added to their corresponding HtmlItem node in the CST tree.

VI. CONTENT ESTIMATION

We estimate the similarity of content in the web pages using cosine similarity measure which is the cosine of the angle between the query vector q and the document vector d_j. Then we will calculate the weight of each node (term) in CST tree such as Text node, Image node, and Link node. We use TF-IDF scheme (Term Frequency and Inverse Document Frequency) to calculate the weight. The weight of a term t_i in document d_j is the number of times that appears in document. In this scheme, an arbitrary normalized w_{ij} is defined as follows;

$$w_{ij} = c(t_i, d_j) = \frac{tf_{ij} \log(N / n_i)}{\sqrt{\sum_{i=1}^{n} (tf_{ij})^2 [\log(N / n_i)]^2}}$$

where,

$t_i = i^{th}$ term in document d_j

tf_{ij} = frequency of the word t_k in document d_j

$idf_i = \log\left(\frac{N}{n_i}\right)$ inverse document frequency of word t_i in entire dataset

n_i = number of documents containing the word t_i

N = total number of document in the dataset

Then we will get the weight of content node,

ContentWeight = TextWeight+ ImageWeight + LinkWeight

The content value which is computed by the children nodes of HtmlItem node will be added to element which is the first child of HtmlItem node. Before adding to the parent node, the proposed system checks the HtmlItem nodes whether there are same level nodes. If so, we will calculate similarity of these nodes using their weights and then eliminate all HtmlItem nodes except the highest similarity node. Ranking of the documents is done using their similarity values. The top ranked documents are regarded as more relevant to the query.

$$\text{Cosine } (d_j, q) = \frac{< d_j \bullet q >}{\|d_j\| \times \|q\|}$$

$$= \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}\, 2} \times \sqrt{\sum_{i=1}^{V} w_{iq}\, 2}} \qquad \text{eq:(1)}$$

where, w_{ij}= term weight in the document
w_{iq}= term weight in the query

```
Function: Generate CST
Input: DOM tree
Output: Content Structure Tree (CST)
1: add htmlnode to the CST
2: for each node n in the DOM tree do
3:     if (ContentNode (n) = = true) then
4:         E = buildContentNode(n);
5:         addCST(E)
6:     else if (HtmlNode (n) = = true) then
7:         E = buildHtmlNode(n);
8:         GenerateCST (child (n));
9:         addCST (E)
10:    end if
11: end for
```

Figure 4. Generate CST tree algorithm

```
Function: Extract MainBlocks
Input  : Content Structure Tree (CST),
Output: Main Blocks
1: for each node n in CST do
2:     if (ConentNode(n) = = true) then
3:         weight= computeContentNode(n);
4:         addCurrentElementNode(weight);
5:     end if
6: end for
7: visit bottom_up in CST
8:  if (HtmlNode (n) = = true) then
9:     if (siblings) then
10:     compute the similarity of weight of each
          HtmlNode using eq: (1)
11:     eliminate HtmlNode which is less
          similarity than other in the same floor
12: add the weight of the highest similarity node
        to its parent
13:   else add weight of HtmlNode (n) to its parent
14:   end if
15: end if
```

Figure 5. Extract main content algorithm

VII. CONCLUSION AND FUTURE WORK

Many researchers have been developed several approaches for extract main content from web pages. Most of the approaches based on the only DOM tree. In this paper, the proposed system based on the CST tree generated by the DOM tree and also could extract the relevant documents from the web pages using cosine similarity measure. A commercial Web page typically contained many information blocks. Apart from the main content blocks, it usually has such blocks as navigation panels, copyright and privacy notices, and advertisements which are called noisy blocks. These noisy blocks can seriously harm Web data mining. In our future work, we will detect and remove noises on web pages and also extract the main content.

REFERENCES

[1] C. Li, J. Dong, and J. Chen, "Extraction of Informative Blocks from Web Pages Based on VIPS", 1553-9105/ Copyright January 2010.

[2] L. Yi, B. Liu, and X. Li, "Eliminating Noisy Information in Web Pages for Data Mining", in Proc. ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2003).

[3] Y. Fu, D. Yang, and S. Tang,"Using XPath to Discover Informative Content Blocks of Web Pages", IEEE. DOI 10.1109/SKG, 2007.

[4] P. S. Hiremath, S. S. Benchalli, S. P. Algur, and R. V. Udapudi, "Mining Data Regions from Web Pages", International Conference on Management of Data COMAD, India, December 2005.

[5] S. H. Lin and J. M .Ho, "Discovering Informative Content Blocks from Web Documents", in Pro. ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp.588-593, July 2002.

[6] D. Cai, S. Yu, J. R. Wen, and W. Y. Ma, "VIPS: a Vision- based Page Segmentation Algorithm", Technical Report, MSR-TR, Nov. 1, 2003.

[7] P. M. Joshi, and S. Liu, " Web Document Text and Images Extraction using DOM Analysis and Natural Language Processing", ACM, DocEng, 2009.

[8] Y. Li and J. Yang, "A Novel Method to Extract Informative Blocks from Web Pages", IEEE. DOI 10. 1109/ *JCAI*, 2009.

[9] R. R. Mehta, P. Mitra, and H. Kamick, "Extracting Semantic Structure of Web Documents Using Content and Visual Information", ACM, Chiba, Japan, May 2005.

[10] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm,"DOM-based Content Extraction of HTML Documents", *Pro.* 12 th International Conference on WWW, ISBN: 1-58113-680-3, 2003.