International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-2, Issue-5)

Research Article | **May 2013**

Discovering Informative Block Content from the Web Page

Piyush Rai
IET Dr R M L Avadh University Faizabad
India.

Dr. Suneet Chaudhary
DIT Dehradoon
India.

Abstract:

The rapid growth of World Wide Web has been tremendous in recent years. With the large amount of information on the Internet, web pages have been the potential source of information retrieval and data mining technology such as commercial search engines, web mining applications. However, the web page as the main source of data consists of many parts which are not equally important. Besides the main contents, a web page also comprises of noisy parts that can degrade the performance of information retrieval applications. Most of the previous approaches used heuristic rule sets to locate the main content. Our contribution in this work is mainly to develop the web content extraction module which uses a hybrid approach that consist of machine learning and our own developed heuristic approaches namely Largest Block String, String Length Smoothing, and Table Pattern. Our work differs in the following ways: the operational settings of the content extraction module, the features used, the dataset type, the type of heuristic and the evaluation method.

Keywords:

I. Introduction

Since its birth in the beginning of the 1990s, the World Wide Web (WWW) has been undergoing remarkable growth. Originated as a hypertext system for accessing many forms of documentation at CERN, the WWW rapidly grow as it is accessible for public use through the web browser. Along with its tremendous growth, the web has been experiencing many changes; one of them is related to how its content is presented to the user. Typically, a modern web document comprises of different kinds of content. As illustrated , a news page, for instance, besides the article posting as the main content it also contains other noisy contents such as user comments, navigational menus, headers, footers, links to other news page, advertisements, copyright notices, privacy policies which scatter over the page. Considering the fact that a web document contains various forms of contents, it influences the way human browses the document. When browsing a particular web document, most of the time users typically focus on the main content and ignore the additional contents. For human, this behavior can be done relatively fast and accurate because they can use their knowledge, visual representation and layout of the web pages to distinguish the main content from other parts. On the other hand, since computer software is not as intelligent as human to distinguish between the main content and the noisy content, this becomes the challenge for commercial search engines, web miners and document as a data source. A search engine, for instance, typically indexes the whole text of a web page. As a result, the noisy contents which is useless information remains in the index. The presence of noisy contents may degrade the performance of such Information Retrieval applications for example the quality of the search result, accuracy of information extraction, and the size of the index. In order to alleviate this problem an approach to extract only the main content from web documents during data acquisition (e.g. crawling process) is needed. This task is needed to clean the web document from noisy contents. To the best of our knowledge, there is no commonly agreed terms which describe this task. However, some describe this task as a web content extraction task.

II. Operational Setting

The web content extraction operates on the data acquisition phase of the Information Retrieval system. As for problem domains, in this paper we select the domain for web content extraction namely shopping websites.

Figure 1: Typical Example of Commercial Web Page

III. Objective And Methodology

The objective of the paper is to develop a web content extraction method that, gives an arbitrary HTML document, should be able to extract the main content and discard all the noisy content. In order to accomplish our objective, we conduct a study of existing approaches to web content extraction. The problem of web content extraction has been investigated by researchers and many kinds of content extraction methods have been proposed. In general, there are two issues which can be observed. The first issue is the source base of the content extraction. Typically, a content extraction method uses either the Document Object Model (DOM) representation or the plain HTML source code. The second issue is related to the general approach to do content extraction. Since our operational settings require that our content extraction module runs during data acquisition, we need a relatively lightweight and fast method to do extraction. In general, single document extraction is relatively faster rather than multiple document extraction because it only considers the document at hand during extraction without looking into other documents from the same host. Most of the existing methods in single document extraction operate based on certain heuristic in order to perform content extraction. For instance, by examining certain features such as the number of hyperlinks, the text density, the ratio between HTML tags to text etc. For DOM-based methods, usually the existing methods perform web page segmentation prior to content extraction. Definition :Web page segmentation is a task which breaks down the structure of a web page into smaller segments of certain granularity. The web page segmentation process is needed because there are dozens of DOM nodes in a single web document and we need to focus on DOM nodes in certain granularity. In this paper, we use the DOM tree representation as the base since we can obtain many kinds of features by accessing the DOM nodes. Also, by using DOM nodes, we still have the information of the document structure which can be helpful for some reasons e.g. to detect certain pattern structure in the document, to traverse to the other part of the document etc. The Document Object Model is a cross-platform and language independent convention for representing and interacting with objects in HTML, XML, and XHTML documents.

IV. Feature Extractor (Fe)

The Feature Extractor (FE) algorithm by Debnath et al [18] is a content extraction algorithm which based on DOM block structures. The algorithm segments a web document into blocks and selects certain blocks to be extracted. A block here corresponds to the DOM sub tree nodes. The algorithm will start working from the root node and recursively splitting the document into blocks. They defined a set of HTML tags which denotes a block namely table, tr, hr, and ul. FE uses the feature such as the presence of nested blocks, texts, images, applets, or contained JavaScript code. FE will extract the blocks which is dominant in certain features. In the context of content extraction, for example, we can set the feature we need it's the text properties. As a result the blocks that will be extracted will be those which is rich with text. The K-Feature Extractor, the

variant of FE, works a little bit different, instead of simply choosing one single winning block the blocks which are able to pass the first iteration are clustered using a k-means clustering algorithm. Afterwards, the cluster with the highest probability for the desired feature is chosen as the winner. FE allows many kinds of features to be incorporated in the content extraction process however we have to select the features manually and define the proper threshold values.

```
Input: D: DOM node (root node), T: set of block defining HTML elements
Output: B: Set of blocks
B ←D
for all t ∈ T do
for all b ∈ B do
if b hasChildNode (t) then
BN ← getBlocks (b, t)
B ← (B- b) ∪ BN
end if
end for
end for
return B
function getBlocks (b, t);
B ← Empty
C ← descendants (b)
for all m ∈ C do
if elementType (m) = t then
B ← B ∪ {m}
end if
end for
```

Feature Extractor Algorithm

V. Web Contet Extraction Approaches

In this paper we present our approach in web content extraction. Our approach is different in the following ways. First, it subdivides a webpage into smaller semantically-homogeneous components based on their content. We refer to such components as blocks. A block is a portion of webpage enclosed within an open tag and its matching closed tag.
Second, after getting all the blocks we may find the desired block using the class property of a tag.

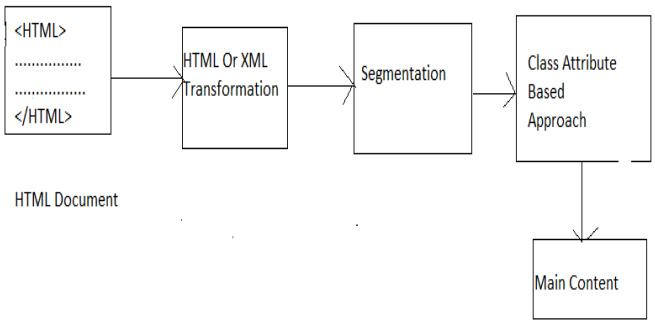

Fig 2 Overall Approach for Main Content Extraction

3.1 HTML-XML TRANSFORMATION.

World Wide Web consortium has warmly recommended usage of stricter standard Markup Languages, such as XML in order to reduce errors resulted in the process of parsing various web pages created by disobeying the basic rules. Despite this recommendation, there still remains a huge quantity of web pages that do not respect the new standards, and with whom, the parser of search engine must cope. We transform each HTML page into a well-formed XML page. Its goal is to replace the HTML language in the future and obtain cleaner documents.

Documents must be well-formed: all elements must be nested inside on unique root element **<html>**, any element can have children elements; children elements must be correctly **closed** and **properly nested**.

```
<html>
  <head>...</head>
  <body>...</body>
</html>
```

tag names must be in lowercase

```
<body>
   <p> This is the paragraph </p>
</body>
```

all XML elements must be closed

Wrong: My text

Correct: My text

XHTML elements must be properly nested

Wrong: Bold, italic, bold, italic

Correct: Bold, italic, bold, italic

attribute names must be in lower case

Wrong: <table WIDTH="100%">

Correct: <table width="100%">

attribute values must be quoted

Wrong: <table width=100%>

Correct: <table width="100%">

attribute minimization is forbidden

Wrong: <frame noresize>

Correct: <frame noresize="noresize" />

the id attribute replaces the name attribute

Wrong:

Correct:

The lang attribute applies to almost every XML element. It specifies the language of the content within an element.

The XML DTD defines mandatory elements: the 'html', 'head' and 'body' elements must be present, and the 'title' must be present inside the head element; all XML documents must have a DOCTYPE declaration

```
<!DOCTYPE  html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Title </title>
</head>
<body> Continue </body>
</html>
```

VI. Segmentation Of Main Content

The goal of segmentation is to break down the structure of a web document into smaller segments with certain granularity. In order to do this we based our segmentation process from the Document Object Model (DOM) of the web document. In this section we start by describing about the DOM structure and next we show how we use the DOM structure to perform

segmentation of a web document By utilizing DOM, one can construct document, navigate through the structures of the document, and perform operations such as add, update, delete the properties of the elements. As a standard programming interface, DOM is designed to be programming language independent. There are numerous languages binding for DOM such as Java Essentially, every HTML elements rendered in the browser correspond with one node in the DOM tree.

The DOM based segmentation process starts from the body of the document and skips the HTML version information and header section since those two parts most likely will only contain scripting and style sheet declaration and they are not visible elements. In the body part of the document, there can be dozens of HTML elements inside. In order to focus our segmentation process we only interested to HTML elements which define structural style in the document. These elements basically are used to define sections in the document. According to the HTML 4.01 specification there are several HTML elements which define structural element namely block level elements (div, span tags), list elements (ul, ol, li tags), table elements (table, tr, td tags).

Example- we can illustrate the segmentation using following example of a simple web page

```
<div> Hello
<table><tr><td>INDIA></td></tr>
<tr><td>USA</td></tr></table>
<div>SMS</div>World
</div>
```

Div	Table	Tr	Td	Tr	Td	Div
Hello World			India		USA	SMS

Fig 3 : Segmentation of a Web Page

VII. Conclusion

In general, according to our paper we have shown that our approach namely the combination of segmentation and our own developed rule based approach is competitive compared to the existing heuristic methods. Essentially, the main results and contributions of this paper fall in two categories:

Segmentation for web content extraction and the rule based approach which consists of class property associated with a tag for main content extraction.

REFERENCES

[1] R. Hartono A.F.R. Rahman, H. Alam, Content extraction from html documents, In 1st Int. Workshop on Web Document Analysis (2001).

[2] Mohsen Asfia, Mir Mohsen Pedram, Amir Masoud Rahmani: Main Content Extraction From Detailed web Pages. In: International Journal Of computer Applications(0975-8887) Volume 4- No. 11, August 2010.

[3] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Computation, vol. 10, 1998, pp. 1895-1924

[4] Mirel Cosulschi, Nicolae Constantinescu, and Mihai Gabroveanu: Annals of University of Craiova, Math. Comp. Sci. Ser. Volume 31, 2004, Pages 109_121 ISSN: 1223-6934.

[5] Thomas Gottron, Content extraction: Identifying the main content in html documents, Ph.D. thesis, Johannes Gutenberg-University Mainz, 2008.

[6] D.S. Hirschberg, A linear space algorithm for computing maximal common subsequences, Communications of ACM Volume 18 Number 6, 1975, p. 342.

[7] P.S Hiremath, Siddu P. Algur: Extraction Of Flat And Nested Data Records From Web Pages. In: P.S. Hiremath et al / International Journal on Computer Science and Engineering Vol.2(1), 2010, 36-45.

[8] Dongdong Hu and Xiaofeng Meng: School of Information Renmin University of China: Automatic Data Extraction from Data-rich Web Pages.

[9] http://www.w3.org/TR/2000/WD DOM-Level-1-20000929/introduction.html.

[10] Deng Cai, Shipeng Yu, Ji-Rong Wen, Wei-Ying Ma: VIPS- a Vision-based Page