iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** II **Month of publication:** February 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Extraction of Information from Web Page Using Content Mining Approach

Pranali Gatfane [#1], Rani Tanpure [#2], Anjali Masodkar [#3], Vrushali Patil[#4]
[#] *Computer Science Department, SGBAU*

***Abstract*— Today internet has made the life of human dependent on it. Almost everything and anything can be searched on net. 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. Thus cleaning the web pages before mining becomes critical for improving the mining results. In our work, we focuses on identifying and removing local noises in web pages to improve the performance of mining. The information contained in these non-content blocks can distract the user and also harm web mining So, it is important to separate the informative primary content blocks from non-informative blocks. So, we propose a system that remove various noise patterns from any web page. There are two steps, Web Page Segmentation and Informative Content Extraction, are needed to be carried out for Web Informative Content Extraction. We are going to analyze the web page and by using methods and algorithm we extract topic information requested by user.**

***Keywords*— Web Mining, Web Content Extraction, DOM Tree, Information retrieval, HTML Parser**

I. INTRODUCTION

The Web is perhaps the single largest data source in the world. Web mining aims to extract and mine useful knowledge from the Web. The World Wide Web has rich source of tremendous information which continues to expand in size and complexity. Many Web pages are unstructured and semi-structured, so it consists of noisy information like advertisement, links, headers, footers etc. This noisy information makes extraction of Web content tedious. Many techniques are present for web content extraction. The application of data mining techniques to automatically discover and to extract knowledge from Web data, including Web documents, hyperlinks between documents, usage logs of Web sites, etc, is called Web mining. Some of the data mining techniques applied in Web mining are association rule mining, clustering, classification, frequent item set. Some of the sub tasks of Web mining are finding of relevant resource, selection of information and pre-processing, generalization and analysis. Web content mining is used for extracting useful information from Web pages. Web page content can be structured, unstructured and semi-structured. Structured Web page data are easy to extract when compared with unstructured and semi-structured data. Web Content Extractor normally extracts a whole Web page including links, header, footer, main content and advertisement. During the extraction unwanted data like links, header, footer and advertisement are treated as noisy information. To eliminate the noisy information and extract the useful information is a challenging problem. Many techniques were proposed for eliminating noisy information. Whenever a user query the web using the search engine like Google, Yahoo, AltaVista etc, and the search engine returns thousands of links related to the keyword searched. Now if the first link given by the user has only two lines related to the user query & rest all is uncluttered material then one needs to extract only those two lines and not rest of the things. The current study focuses only on the core content of the web page i.e. the content related to query asked by the user. The title of the web page, Pop up ads, Flashy advertisements, menus, unnecessary images and links are not relevant for a user querying the system for educational purposes.

II. RELATED WORK

This study is proposed to deal with the problem of intra-page redundancy that causes search engines to index redundant contents and retrieve non-relevant results. The problem also affects Web miners since they extract patterns from the whole document rather than the informative content. Thus, we illustrate studies of both fields. In the rest of the paper, for better understanding, we use information retrieval (IR) systems to denote search engines and information extraction (IE) systems to denote Web or text miners. Many IR systems have been implemented to automatically gather, process, index, and analyze the Web documents for serving users information needs. It also parses contents of the page based on HTML or other mark-up language like XML. The former called text mining. jsoup is designed to deal with all varieties of HTML found in the wild; from pristine and

©IJRASET 2015: All Rights are Reserved

validating, to invalid tag-soup; jsoup will create a sensible parse tree.

A. Extraction

Extraction encompasses all the information retrieval programs that are not meant to preserve the source page. This covers uses like: •text extraction, for use as input for text search engine databases for example
•link extraction, for crawling through web pages or harvesting email addresses
•screen scraping, for programmatic data input from web pages
•resource extraction, collecting images or sound
•a browser front end, the preliminary stage of page display
•link checking, ensuring links are valid
•site monitoring, checking for page differences beyond simplistic diffs
There are several facilities in the HTML Parser codebase to help with extraction, including filters, visitors and JavaBeans.

B. Transformation

Transformation includes all processing where the input and the output are HTML pages. Some examples are: •URL rewriting, modifying some or all links on a page
•site capture, moving content from the web to local disk
•censorship, removing offending words and phrases from pages
•HTML cleanup, correcting erroneous pages
•ad removal, excising URLs referencing advertising
•conversion to XML, moving existing web pages to XML
During or after reading in a page, operations on the nodes can accomplish many transformation tasks "in place", which can then be output with the to Html method. Depending on the purpose of your application, you will probably want to look into node decorators, visitors, or custom tags in conjunction with the Prototypical Node Factory.

C. DOM Tree Approach

It is the Document Object Model which is a standard for creating and manipulating in memory representation of HTML content. It defines logical structure of document and the way a document is accessed and manipulate. Proposed approach concentrates on web pages where the underlying information is unstructured text. The technique used for information extraction is applied on entire web pages, whereas they actually seek information only from primary content blocks of the web pages. The user specifies his required information to the system. Web crawlers download web pages by starting from one or more seed URLs, downloading each of the associated pages, extracting the hyperlink URLs contained there in, and recursively downloading those pages. Therefore, any web crawler needs to keep track both of the URLs that are to be downloaded, as well as those that have already been downloaded. DOM analyzer defines the concept of blocks in web pages. Most web pages on the internet are still written in HTML. Even dynamically generated pages are mostly written with HTML tags, complying with the SGML format. The layouts of these SGML documents follow the Document Object Model tree structure of the World Wide Web Consortium.2. The relevant pages given out by the web crawler are represented in a form of DOM tree HTML DOM is in a tree structure, usually called an HTML DOM tree. Following Figure illustrates a simple HTML document and its corresponding DOM tree. We are interested only in the <BODY> node and its offspring. In this example, <BODY> node has three children: element nodes <B> and <I>, and text node #and. Element node<B> has a text node child #bold, and element node <I> has a text node #italic. Following the DOM convention, we use <> to indicate element node, and use # to indicate text.

III. ANALYSIS OF PROBLEM

Noisy content makes the problem of information harvesting from web pages much harder. Web pages typically contain non-informative content, noises that could negatively affect the performance of Web Mining. Whenever a user query the web using the search engine like Google, Yahoo, AltaVista etc, and the search engine returns thousands of links related to the keyword searched. Now if the first link given by the user has only two lines related to the user query & rest all is uncluttered material then one needs to extract only those two lines and not rest of the things. Considering that a huge amount of world's information re-sides in web pages, it is becoming increasingly important to analyze and mine information from web pages.

A. Identifying Articles

The first step, determining whether a page contains an article, is a document classification problem. Our evaluation implicitly

©IJRASET 2015: All Rights are Reserved

assumes that such a classifier is provided, since all our testing examples contain articles. No such assumption is made in training, however, and the semi-automatically generated training data may erroneously contain non-articles. To be specific, by "article" we mean a contiguous, coherent work of prose on a single topic or multiple closely related topics that
all one comprises the main informational content of the page—news stories, encyclopedia entries, or a single blog post are considered articles, whereas a collection of headlines with brief summaries, a list of search results, or a set of blog posts are not .For the new domain, a more specific definition is employed, as news websites have many pages that are not commonly considered news articles (such as recipes), but are articles in another domain (such as cookbooks). Thus, in addition to the general requirements for an article, a news article must be a story or report at least two paragraphs and eight total sentences in length. The length requirement serves to exclude those pages that are merely brief summaries (typically with a link to the full article). Finally, a complication arises in determining exactly where an article begins and where it ends. A news article typically follows the pattern "headline → by lines → main text → by lines", where everything other than the main text is optional (by lines after the main text, for example, are less common than those before it, and some articles have neither a headline nor by lines). To resolve this, the true article text is considered to be the main text alone, without the headline or by lines, and this is what the wrappers that generate the training examples (as described in the next section)label as the extraction. However, to be fair in evaluation(especially with regards to VIPS), the evaluation examples are labeled with multiple extractions that correspond with multiple possible interpretations of what constitutes the bounds of an article's text. An extraction may start at (1) the first word of ahead line, (2) the first word of an authorship by line before the main text (e.g. "By John Doe") or the organization by line (e.g."The Associated Press") if no author is given, and (3) the first word of the main text. An extraction may end at (1) the last word of the main text, or (2) the last word of an authorship by line or organization by line appearing after the main text. Articles may thus have up to six possible extractions, while the norm is three. When experimental results are reported, predicted extractions are compared against both the main text alone (the shortest of the possible extractions) and against the "best" extraction, the one that yields the most favorable F1-score for that particular prediction.

B. Cleaning Extracted Blocks

Maximum subsequence segmentation addresses the second part of the problem, identifying the block of HTML containing the article text, but further cleaning may be required to remove extraneous words from embedded ads, lists of links to other stories, images with captions, eye-catching reiterations of quotes appearing in the article, etc. A key observation is that, once the article's HTML block has been identified, removing whatever "junk" remains becomes much simpler. Indeed, after inspecting our evaluation sets we found this could be done using just a few rules with little error. Starting with an extracted block of HTML that starts at the first word and ends at the last word of the main text of the article:

1. Remove the contents of any <IFRAME> or <TABLE>tag pair
2. Remove the contents of any <DIV> tag pair that
contains a hyperlink (<A>), <IFRAME>, <TABLE>,
<IMG>, <EMBED>, <APPLET> or <OBJECT>

This heuristic works well because the text of most news articles very rarely contains links, and embeds are intended to either show you something (e.g. a picture) or get you to go somewhere (e.g. a link). The few mistakes we did observe were mostly trivial (such as leaving in the word "Advertisement"), though one article would lose most of its text because it contained links and was inside a<DIV>. Outside the news domain we will not always be so fortunate: encyclopedia articles, for example, tend to contain many links, and would require a more sophisticated approach. In practice, however, cleaning the extracted block is far less important than identifying it correctly, since web designers typically place the largest amount of junk text, such as navigation bars or user comments, around the article, not inside it, regardless of domain. This is demonstrated by our very strong cross-domain performance cleaning general articles from the Clean Eval task despite making no effort to clean the blocks selected by MSS.

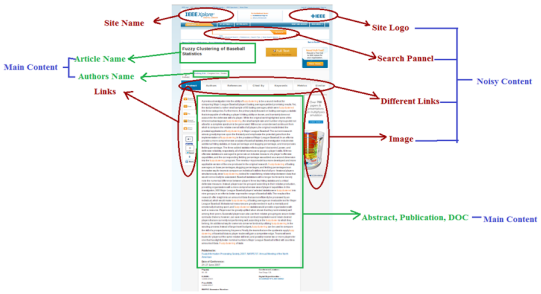


Fig.1 web page of IEEE explore article

IV. PROPOSED WORK

Proposed approach concentrates on web pages where the underlying information is unstructured text. The technique used for information extraction is applied on entire web pages, whereas they actually seek information only from primary content blocks of the web pages. The user specifies his required information to the system.

Input: The Web Documents (web page of IEEE explore article).

Output: The web document containing only informative contents such as abstract of the paper, title of the paper, date of publication, pages and authors name particular paper.

Procedure: Take the input web page for content extraction .After that pass the webpage through HTML parser that converts into HTML code .Now Create Document Object Model (DOM) tree for above HTML code.

Apply various algorithm on DOM tree for extracting informative content. Finally we get desired output requested by user.

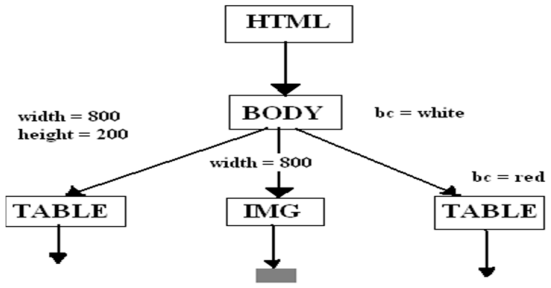


Fig 2: DOM Tree

```
<HTML>
  <BODY bgcolor=WHITE>
    <TABLE width=800 height=200 >
      ...
    </TABLE>
    <IMG src="image.gif" width=800>
    <TABLE bgcolor=RED>
      ...
    </TABLE>
  </BODY>
</HTML>
```

Fig 3: HTML code

A. *Jsoup*

jsoup is a Java library for working with real-world HTML. It provides a very convenient API for extracting and manipulating data, using the best of DOM, CSS, and jquery-like methods. jsoup implements the WHATWG HTML5 specification, and parses HTML to the same DOM as modern browsers do.

- scrape and parse HTML from a URL, file, or string
- find and extract data, using DOM traversal or CSS selectors
- manipulate the HTML elements, attributes, and text
- clean user-submitted content against a safe white-list, to prevent XSS attacks
- output tidy HTML

V. CONCLUSIONS

This paper proposed a novel task for finding local noise in web pages. Using DOM tree approach contents of the web pages are extracted by filtering out non informative content. With the Document Object Model, programmers can build documents, navigate their structure, and add, modify, or delete elements and content. With this features it becomes easier to extract the useful content from a large number of web pages. In future this approach will be used in information retrieval, automatic text

categorization , topic tracking, machine translation, abstract summary. It can provide conceptual views of document collections and has important applications in the real world.

VI. ACKNOWLEDGMENT

We take this opportunity to express our gratitude and indebtedness to our guide **Prof. P. G. Gondse** as well as H.O.D. **Prof. A. B. Raut**, Computer Science and Engineering department, who is a constant source of guidance and inspiration in preparing this work. Her constant help and encouragement helped us to complete our work.

We are grateful to Principal **Dr. A. B. Marathe,** for his encouragement and support.

We are also thankful to all the staff members of Computer Science and Engineering department, whose suggestions helped us to complete the work and those who have directly and indirectly helped for completion of the work

REFERENCES

[1] S. Baluja, "Browsing on smalls screens: Recasting Web-page segmentation in to an efficient machine learning framework", *Proceedings of the 15th International Conference on World Wide Web*, pp. 33–42, 2006.

[2] M. Baroni, F. Chantree, A. Kilgarri, S. Sharoff, "Cleaneval: A competition for cleaning Web pages", *Proceedings of the sixth International Conference on Language Resources and Evaluation*, 2008

[3] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. Vips: vision-based page segmentation algorithm. Technical report, Microsoft Research, 2003.

[4] A. H. F. Laender, B. A. Ribeiro-Neto, A. S. da Silva,and J. S. Teixeira. A brief survey of web data extraction tools. SIGMOD Rec., 31(2):84-93, 2002.

[5] Y. Yesilada, ―Web Page Segmentation: A Review,‖ eMINE Technical Report Deliverable 0 (D0), 2011.

[6]. Y. Yesilada, ―Heuristics for Visual Elements of Web Pages,‖ eMINE Technical Report Deliverable 1 (D1), 2011.

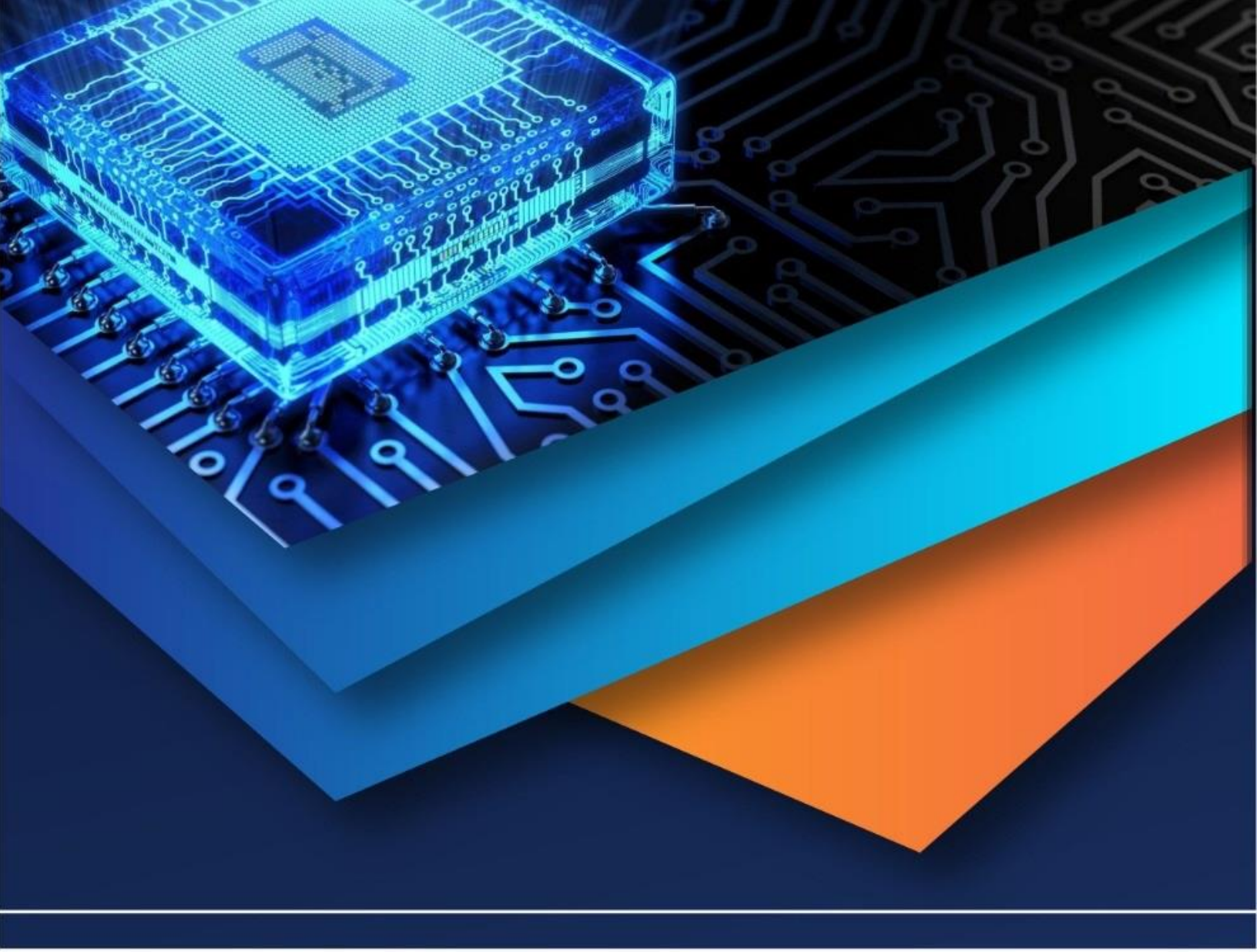
[7] Zhao Xin-xin, Suo Hong-guang, Liu Yu-shu. Web Content Information Extraction Method Based on Tag Window. Application Research of Computers. 2007,24(3).-144-145,180.

[8] Pan Donghua, Qiu Shaogang. Web Page Content Extraction Method Based on Link Density and Statistic. The 4Th International Conference.

[9] A. F. R. Rahman, H. Alam and R. Hartono "Content Extraction from HTML Documents"

[10] Wolfgang Reichl, Bob Carpenter, Jennifer Chu-Carroll, Wu Chou "Language Modeling for Content Extraction in Human- Computer Dialogues".

©IJRASET 2015: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)