International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 04 | Apr-2016 www.irjet.net p-ISSN: 2395-0072

An Effective Approach to Extract Information from web pages

R.Manjula[1] and A.Chilambuchelvan[2]

[1] *Research Scholar, Department of CSE, R.M.K Engineering College Chennai, India*
[2] *Professor , Department of CSE, R.M.K Engineering College, Chennai, India*

Abstract: Web mining is one of the data mining techniques to extract relevant information from the web and to analyze users' web browsing patterns to identify users' web interests and their needs. Web mining as a data mining technique to extract knowledge from World Wide Web pages and services. Web is very large, diverse, dynamic, and mostly unstructured data storage, it raises the difficulty to deal with the information from different perspectives. The users would get the relevant documents they want from search results with fast response time. The web service provider needs to watch the web usage of users to identify their interests. All of them need techniques and methods to facilitate the extraction of web contents, and conclude the appropriate knowledge in easy and accurate way. For all the previous reasons web mining became very active and important research area. Web contents are the primary information of web document, which usually include different types of data such as texts, images, hyperlinks. This unwanted web contents are called noise information and should be cleaned before mining web contents process begins. This system extends and advances to automatically generate and extract and mine structured web contents from different web pages based on similarity matching, and stores the extracted information in historical data warehouse. similarity matching of DOM tree tag nodes for identifying data blocks and data regions to be used , for generating web site content extraction, which contain similar data.

Keywords: Web content mining, Web data extraction, Wrapper, DOM, VIPS

1. Introduction

World Wide Web (WWW) now represents a huge repository of data source and Number of web pages is growing very fast every day in the world[1]. Web coverage of information is very wide and diverse, where users find some difficulties to retrieve all the information they want[2][3]. So, web data analysis and web content mining have become very important research area. Web content mining aims to extract useful knowledge from the contents of web and conclude future decision based on this knowledge. For example, web content mining can extract potential useful information about products or individual item from different web sites such as prices, titles, products series, etc. Web contents are heterogeneous in nature and can be in different formats, e.g., structured tables, texts, images, links, multimedia data, etc. So far there is no complete automatic extraction model catches the full diversity of web contents. Web pages contain different types of data

which are embedded under different complex structures. The available approaches for extracting data contents from the web are manual wrappers, supervised wrapper induction, or automatic data extraction. This system is using an automatic extraction system that tries to extract diverse heterogeneous web contents. With the advent of Internet more and more information is available, and the need to search and query relevant data has become a difficult task for both humans and machines.

One of the main goals of Information Extraction and the creation of Wrapping Systems is to generate Structured Information based on Web Pages in a way that can be automatically processed and transformed[5]. Due to the nature of the Information available in the Web, either by its origin, inconstancy or frequency of updatability, it is necessary to have advanced tools and algorithms in order to fill forms and iterate through pages in a dynamic and consistent way. wrapping are used There are and will be a huge number of legacy HTML applications,. Web wrappers are thus essential in interoperability efforts and used in selectively extracting structured content from semi-structured web sources [3].Data are valuable and very important because they represent the main theme of their websites and a list of such information represents a list of similar items, e.g., list of products, books, services, etc. Mining data is very useful because it allows different information from different web pages to be integrated together in one database, which add more web services like shopping comparisons, e-commerce, and web search. Web Sources allow access to deep web and underlying database

in HTML or semi-structured format, which makes it difficult task on any software to extract data objects and their related attributes from web pages.

2. Proposed system

One of the stages in content gathering is *crawling*. Crawling is an automated process of collecting data typically in hyperlinked documents [1]. A crawler systematically follows the hyperlinks between documents and stores the local copies in the database. The behavior of this crawler is defined in a configuration, which enables to set up the parameters that we need such as the depth, maximum pages to fetch, stop condition, time to re-crawl etc.

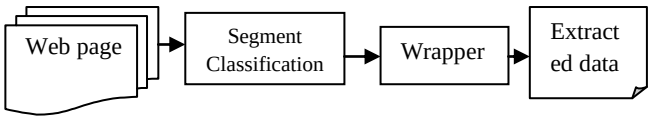

Fig.1 web content extraction approach

2.1 Segment Classification

The goal of segment classification is to break down the structure of a web document into smaller segments with certain granularity. our segmentation process is doing this segmentation from the Document Object Model (DOM) of the web document. this section describes about the DOM structure and shows how we use the DOM structure to perform segmentation of a web document. According to W3C specification [9], DOM is an application programming interface (API) for valid HTML and well-formed XML documents. It defines the logical structure of documents and the way how a document is accessed and manipulated. DOMis used to construct document, navigate through the structures of the document, and perform operations such as add, update, delete the properties of the elements. DOM is designed to be programming language independent in standard programming interface,

2.1.1 DOM tree construction

HTML CODE

```
<html>
 <head>
  <title> Search Engines </title>
 </head>
<body
 <ul>
  <li>
    <a      href=www.yahoo.com>
YAHOO </a>
  </li>
  <li>
   <a      href=www.google.com>
GOOGLE </a>
  </li>
 </ul>
</body>
</html>
```

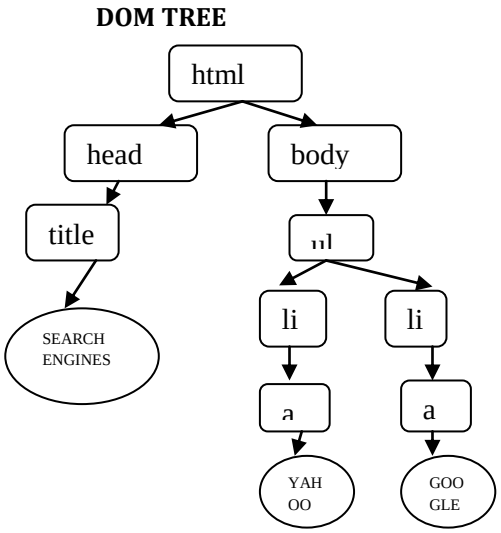

Fig.2 Example of DOM tree construction

In the DOM,[4] the logical structure of a document is represented as a tree structure. An example of a DOM tree representation is shown in Figure 2. From the illustration we can see the DOM tree of the respective web document rendered on the web browser. In HTML elements are rendered in the browser correspond with one node in the DOM tree. For example, the DOM node which is indicated by the arrow sign in Figure 2 corresponds to the side menu in the rendered web document. 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. 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). We traverse the nodes in the DOM tree in a depth first manner and for each node that we encounter; we check whether a node is a structural elements. If it is a structural element we need to check whether it is a good segment. The meaning of good in this context, is related to the content uniformity of the segments.

An HTML document is first retrieved from the Web according to one or more retrieval rules. Once retrieved, it is fed to an HTML parser that constructs a parse tree following the Document Object Model. Extraction rules are then applied on the parse tree and the extracted information is stored in internal format . Instead of using raw HTML text, it uses the DOM tree representation of a web document. Receiving input of a HTML page, this system will parse the HTML string, construct the DOM, traverse the nodes recursively and filter out the non-informative content behind. Each of the filters can be turned on or off and customized to certain degree. Algorithm 1 illustrates the main filters of this system. Generally there are two sets of filter are used. The first set simply ignores specific HTML tag such as styles, links, and images. The second set consists of the advertisement remover, the link list remover, the empty table remover, and the removed link retainer. The advertisement remover maintains a list of advertisement server addresses to detect whether a DOM node contains advertisement elements. The link list remover employs a filtering technique by calculating the ratio between the number of links and non-linked words. This ratio will be compared to certain threshold which can be customized. The empty table remover simply removes table element which doesn't have substantial information inside it by looking at the number of characters. The removed link retainer adds link information back at the end of the document to keep the page browseable. After performing a series of filtering pipeline the final output can be customized as a transformed HTML document or plain text format.

Filtering Algorithm I

Input: N :DOM nodes

1. Get the node type of N
2. Find the parent of node N
3. If node type is an element node then
    a. Get the node name
4. If node name is DIV element then
    a. Remove the node N
5. If node name is TD and the table is empty
    a. Remove the node N
6. Otherwise add node to list.

2.1.2 VISION-BASED PAGE SEGMENTATION (VIPS) ALGORITHM

VIsion-based Page Segmentation is used to find all of the regions of which a document is to be composed. The different web designers provide visual cues that help people to recognize the different regions of a document. We use VIPS algorithm to separate the web pages into number of blocks based on visual features such as horizontal or vertical rules, tables, fonts, or images [4][6].

VIPS Algorithm

Input: DOM tree and visual information
Output: web page with different blocks

1. The DOM tree of the input document is constructed , and it is enriched with information about visual features, e.g., position, background and foreground color, font, or background image
2. Traverse the DOM tree and identify the sub regions.
   - Initially, the algorithm assumes that the whole document is a big region; it then traverses the DOM tree level after level and analyses each node to determine if it can be considered a sub region.
   - if a parent node has a child node of type hr, then that node must be divided into two sub regions;
   - if the background colour or a node is different from the background colour of one of its children, then that child is a sub region;
   - if a table cell does not have any sub regions, then the next table cell should not have any sub regions, either; and so on.
3. Identify the separators that should not intersect the sub regions.
   - After discovering the sub regions in each level of the DOM tree, the algorithm calculates a collection of separators and its visual boxes that do not intersect with any of the sub regions.
   - In other words, a separator is an empty region within a document.
   - Each separator is assigned a weight that is related to the visual difference between the regions that it separates. The authors devised a number of heuristic to calculate the weight of a separator building on how similar the blocks it separates are, what colours they have, if a horizontal rule overlaps the separator, and so on. Adjacent regions that have a separator whose weight is smaller than a predefined threshold are merged.

4. Once all of the regions have been identified, they are organized into a tree and is returned by VIPS.

Then separators are identified between the blocks in order to give the semantic structure of the web page. <table> tag divides the web page into number of partitions . It consists of different subtree for both content and non content blocks. It is used in web pages such as yahoo which are developed based on multiple semantics. Fig 3 shows the block representation of Yahoo page. Yahoo page consists of News, Advertisement, search box, footers, related links, and images. Each and Everything in yahoo page is considered as blocks.

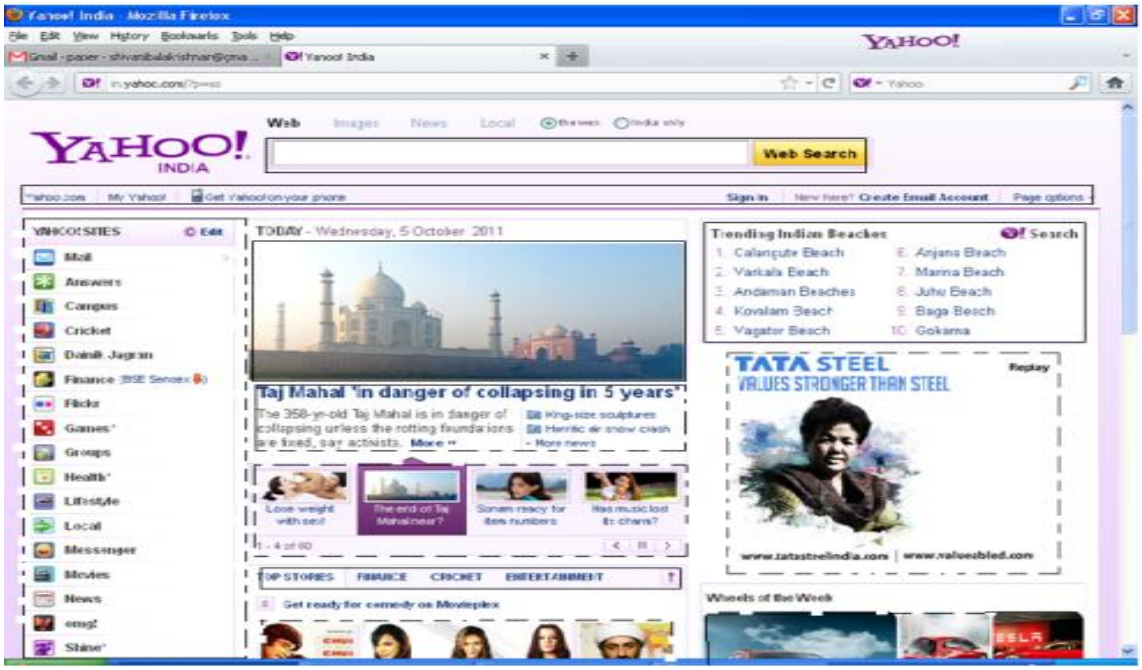

Fig **3** Block representation of Yahoo Home Page

2.2 Feature Extractor

The Feature Extractor (FE) algorithm 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 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 complete algorithm of FE is shown in Algorithm 3. The 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 s 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.

2.2.1. Informative Page Blocks Recognition

Recognising Informative Page Blocks, relies on VIPS and is intended to identify the largest data region in a document. This algorithm is supervised, which implies that the user must provide a few examples of data records. The algorithm then tries to find the regions that contain structures that are similar to these records.

Algorithm 3:

1) It first finds a matching between the DOM trees that represent the data records that the user must provide. This matching builds on the algorithm proposed by the information extractor . The tree, which can be interpreted as a rule that allows to identify the trees in a document that have a data record that is similar to one of the sample data records provided by the user.
2) It then uses VIPS to segment the input document into a collection of candidate regions.
3) It then applies a clustering algorithm to the previous candidate regions. The similarity function used is based on a distance.
4) Next, each region in a cluster must be compared to the tree and the result is a score that is based on the distance. The total score of a cluster is the sum of the partial scores of its regions.
5) The algorithm selects the cluster with the highest score and returns it as the data region in the input document.

2.3 WRAPPER Generation

A Web wrapper is used to convert an HTML document into a form that intended software can understand. A Web wrapper is a [6] program which extracts data of interest from an HTML document and outputs the data in a structured form. Various wrapper construction tools [1] have been proposed for users to easily construct new Web wrappers for arbitrary Web documents. Wrapper construction from given extraction examples is one of the most popular approaches for such tools. In this approach, a user gives a set of portions of an HTML document as examples. Then the system constructs a new wrapper that extracts portions similar to given examples from the HTML document. The extraction manner of the wrapper depends on the given examples. Therefore, a user needs to make many trials with various combinations of examples until an intended wrapper is generated. Various interfaces for Web wrapper construction are proposed in previous studies. However, there are few studies that focus on the improving of this trial-and-error task.

The first class is called manual extraction, where a user or developer manually labels targeted items inside a web page and writes the extraction rules to extract such items. The manual approach suffers from many of problems, it is considered time consuming, requires a lot of human efforts to write extraction rules and update them. The second class is called wrapper induction, where a set of manually labeled pages are given and machine-learning techniques are applied to identify specific patterns and build extraction rules from large initial training web pages. The extraction rules are applied for further manipulation and extraction of data from subsequent pages that contain important information similar to those pages in training collections. Wrapper induction suffers from several problems, where manual labelling is still labor intensive and time consuming. In addition, wrapper needs regular maintenance by experts to accommodate the frequent changes and updates of websites to keep the extraction rules valid. The third class is automatic extraction, where a set of training pages are given and the extraction rules are built automatically. Automatic extraction system is able to extract web contents even if only one training page is given. Many of researchers consider current automatic web content extraction methods as inaccurate and make many assumptions about web pages which need to be extracted. Many of the important information on the web are contained in regularly structured format such as list of online electronic products objects. Such objects represent structured database records generated from underlying database of website and displayed in web page in a regular structured format.

.

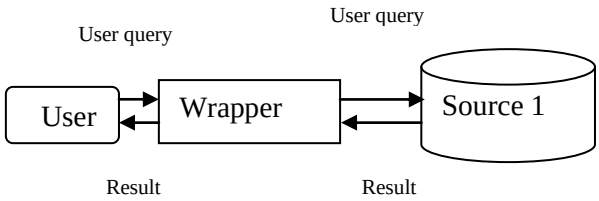

Fig 4. Wrapper for extracting data from source

The initial job of the Web wrapper is to retrieve the to-be-processed Web document. The retrieval layer is in charge of issuing an HTTP request to a remote server and fetching the corresponding HTML page as any Web browser would do. For the wrapper, it is completely transparent, all the job being done in the background by the retrieval layer: creation of the HTTP request, management of connections, handling of redirections and authorizations, etc. The retrieval layer is described by a set of retrieval rules that look like an interface definition: the name of the rule is followed by the list of parameters it takes, the type of the method (GET or POST) and the corresponding url [9].The url might contain some variables to be replaced by their string value in order to offer parameterization. For the POST method, parameters can be specified using the PARAM keyword. Some other information to be included in the HTTP request can also be specified. There are three main classes of data extraction from web.

3. Conclusion

Data mining for Web information extraction will be an important research in Web technology. To makes it possible to fully use the immense information available on the Web one must overcome many mining challenges before we can make the Web a richer, friendlier, and more intelligent resource that we can all share and explore. Many promising data mining methods can help achieve effective Web mining. But using data mining to find a user's profile patterns can further enhance these services. Although a personalized Web service based on a user's history could help recommend appropriate services, a system usually cannot collect enough information about a particular individual to warrant a quality recommendation. Either the traversal history has too little historical information about that person, or the possible spectrum of recommendations is too broad to set up a history for any one individual. So, customizing service to a particular individual requires tracing that person's Web history to build a profile, then providing intelligent, personalized Webservices based on that information. Collaborative filtering can be effective because it does not rely on a particular individual's past

experience but on the collective recommendations of the people who share patterns similar to the individual being examined. This approach generates quality recommendations by evaluating collective effort rather than basing recommendations on only one person's past experience. Indeed, collective filtering has been used as a data mining method for Web Data Mining and effective result presentation in future

References:

[1] Managing semantic content for web by Amit Sheth, Clemens Bertram,David Avant, Brian Hammond,Krysztof Kochut, and Yashodhan Warke ,on JULY • IEEE IEEE INTERNET COMPUTING, AUGUST 2002

[2] Mining Generalized Associations of Semantic Relations from Textual Web Content ,by Tao Jiang, Ah-Hwee Tan, Senior Member, IEEE, and Ke Wang

[3] An Overview of Web Data Extraction Techniques, Devika K1, Subu Surendran nternational Journal of Scientific Engineering and Technology (ISSN : 2277-1581) Volume 2 Issue 4, pp : 278-287 1 April 2013

[4] DATA EXTRACTION FROM DYNAMIC WEB PAGES BASED ON VISUAL FEATURES by,Prof. Sachin Bojewar, Prof.Varsha Bhosale, Shuveta Chanchlani International Journal of Advanced Engineering Research and Studies E-ISSN2249–8974 IJAERS/Vol. I/ Issue IV/July-Sept., 2012/91-94

[5] A Study: Web Data Mining Challenges and Application for Information Extraction T. Sunil kumar1, Dr. K. Suvarchala2 IOSR Journal of Computer Engineering (IOSRJCE)ISSN: 2278-0661, ISBN: 2278-8727Volume 7, Issue 3 (Nov. - Dec. 2012), PP 24-29

[6] A Survey on Region Extractors From Web Documents Hassan A. Sleiman and Rafael Corchuelo on IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING 1

AUTHORS

R.MANJULA received B.E Degree in 2002 from Bharathidasan University and M.E degree in Computer Science and Engineering from Anna University in 2008. She is currently research scholar under the guidance of Dr .A. Chilambuchelvan, Professor in the Department of Computer Science and Engineering in R.M.K Engineering College. Her research interests include various aspects of Knowledge and Data Discovery.

Dr.A. CHILAMBUCHELVAN obtained B.E. Degree in 1989 from Mepco Schlenk Engineering College, Sivakasi and M.E. Degree in 1994 from Coimbatore Institute of Technology, Coimbatore. He did his PhD from College of Engineering, Guindy, Anna University, Chennai in 2008. He is in the teaching profession for the past 22 years and his areas of interest are Knowledge and Data Discovery, soft computing and bio medical engineering. He published 24 papers in International journals and conferences.