International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 3 Issue 5, August 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

The Data Records Extraction from Web Pages

Nwe Nwe Hlaing, Thi Thi Soe Nyunt, Myat Thet Nyo

Faculty of Computer Science, University of Computer Studies, Meiktila, Myanmar

How to cite this paper: Nwe Nwe Hlaing | Thi Thi Soe Nyunt | Myat Thet Nyo "The Data Records Extraction from Web Pages" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-5, August 2019, pp.2258-2262, **https://doi.org/10.31142/ijtsrd28010**

IJTSRD28010

Copyright © 2019 by author(s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (http://creativecommons.org/licenses/by/4.0)

ABSTRACT

No other medium has taken a more meaningful place in our life in such a short time than the world-wide largest data network, the World Wide Web. However, when searching for information in the data network, the user is constantly exposed to an ever-growing flood of information. This is both a blessing and a curse at the same time. The explosive growth and popularity of the world-wide web has resulted in a huge number of information sources on the Internet. As web sites are getting more complicated, the construction of web information extraction systems becomes more difficult and time-consuming. So the scalable automatic Web Information Extraction (WIE) is also becoming high demand. There are four levels of information extraction from the World Wide Web such as free-text level, record level, page level and site level. In this paper, the target extraction task is record level extraction.

KEYWORDS: *Information Extraction (IE), Wrapper, Document Object Model DOM*

1. INTRODUCTION

The rapid development of World Wide Web has dramatically changed the way in which information is managed and accessed. The information in Web is increasing at a striking speed. At present, there are more than 1 billion web sites and web information has covered all domains of human activities. This opened the opportunity for users to benefit from the available data. So Web is being concerned more and more.

To retrieve information on the web, people visit web sites or browse a large number of web pages related by key words with the help of search engines. However, manually visiting and searching sites is very time-consuming. Some researchers, therefore, propose to integrate useful data over the whole Internet with uniform schemes, so, people can easily access and query the data with relational database techniques. At the same time, the integrated data can be mined to provide value-added services, such as comparison shopping. As the data sources on the Internet are scattered and heterogeneous, it is very difficult to integrate data from web pages. On the other hand, web pages may present information with embedded structured data, most of which comes from backend relational database systems. Finding a way to extract structured data from semi structured web pages and integrating the data with uniform schemes is necessary.

Web information extraction (WIE) is an important task for information integration. Multiple web pages may present same information using completely different formats or syntaxes, which makes integration of information a challenging task. Structure of current web pages is more complicated than ever and is far different from their layouts on web browsers. Due to the heterogeneity and lack of structure, automated discovery of targeted information becomes a complex task. A typical web page consists of many blocks or areas, e.g., main content areas, navigation areas, advertisements, etc. For a particular application, only part of the information is useful, and the rest are noises. Hence, it is useful to separate these areas automatically. Web information extraction is concerned with the extraction of relevant information from Web pages and transforming it into a form suitable for computerized data-processing applications. Example applications of WIE include: price monitoring, market analysis and portal integration.

This paper is divided into several sections. Section 2 describes the related work on theoretical background and Web information extraction In Section 3 we discuss our proposed methodology in detail. Section 4 discusses the result of our experimental tests while Section 5 concludes this paper.

2. RELATED WORK

Information extraction from web pages is an active research area. The existing works in Web data extraction can be classified according to their automation degree (for a survey, see [5]). There are several approaches [4], [6], [9], [10], [15] for structured data extraction, which is also called wrapper generation. The first approach [9] is to manually write an extraction program for each web site based on observed format patterns of the site. This manual approach is very labor intensive and time consuming. Hence, it does not scale to a large number of sites. The second approach [10] is wrapper induction or wrapper learning, which is currently the main technique. Wrapper learning works as follows: The user first manually labels a set of trained pages. A learning system then generates rules from the training pages. The resulting rules are then applied to extract target items from web pages. These methods either require prior syntactic knowledge or substantial manual efforts.

The third approach [4] is the automatic approach. The structured data objects on a web are normally database records retrieved from underlying web databases and

IJTSRD International Journal of Trend in Scientific Research and Development ISSN: 2456-6470

displayed in web pages with some fixed templates. Automatic methods aim to find patterns/grammars from the web pages and then use them to extract data. Examples of automatic systems are IEPAD [4], ROADRUNNER [6] extracts a template by analyzing a pair of web pages of the same class at a time. It uses one page to derive an initial template and then tries to match the second page with the template. Deriving of the initial template has to be again done manually, which is a major limitation of this approach.

Another problem with the existing automatic approaches is their assumption that the relevant information of a data record is contained in a contiguous segment of HTML code, which is not always true. MDR [1] basically exploits the regularities in the HTML tag structure directly. MDR works well only for table and form enwrapped records while our method does not have this limitation. MDR algorithm makes use of the HTML tag tree of the web page to extract data records from the page. However, an incorrect tag tree may be constructed due to the misuse of HTML tags, which in turn makes it impossible to extract data records correctly. DEPTA [14] uses visual information (locations on the screen at which the tags are rendered) to find data records. Rather than analyzing the HTML code, the visual information is utilized to infer the structural relationship among tags and to construct a tag tree. But this method of constructing a tag tree has the limitation that, the tag tree can be built correctly only as long as the browser is able to render the page correctly.

Another similar system is Vints[8]. Vints proposes an algorithm to find SRRs (search result records) from returned pages of the search engines. However, our method focuses on list pages of same presentation template. Although some aspects and pieces of web information extraction may be around in various techniques, the important of this paper focus on the some interesting features of web page and block clustering by appearance similarity.

3. SYSTEM OVERVIEW

This section describes the proposed automatic data records extraction from web pages. In this system, there are five steps to extract data record from semi structured web page. The algorithm for the proposed system proposed system is as follows:

```
Algorithm: Data Records Extraction
1. Input :HTML Web page
2. Output :Extracted data table
3. Create DOM tree for input web page P and cleaning
   useless nodes.
4. Segment the web page into several raw chunks/blocks
   Bi={b1,b2…bi}.
5. Filter the noisy blocks based on heuristic rules.
6. Cluster the remaining blocks based on their appearance
   similarity.
7. Labeling the data attributes for extracted data record.
```

Figure1 The Data Records Extraction Algorithm

First of all, input HTML page is changing DOM tree and cleaning useless node as a preprocessing step. Secondly, we tick out several raw chunks as a first round and then filter the noisy block based on noisy features of web pages. Then, thirdly these blocks are clustered by proposed block clustering method. Finally extract information from input web page.

The goal of the system is to offer the extracted data records from web page to the information integration system such as price comparison system and recommendation system.

3.1. Features in Web Pages

Web pages are used to publish information to users, similar to other kinds of media, such as newspaper and TV. The designers often associate different types of information with distinct visual characteristics (such as font, position, etc.) to make the information on Web pages easy to understand. As a result, visual features are important for identifying special information on Web pages.

Position features (PFs).These features indicate the location of the data region on a Web page.

PF1 : Data regions are always centered horizontally.

PF2 : The size of the data region is usually large relative to the area size of the whole page.

Since the data records are the contents in focus on web pages, web page designers always have the region containing the data records centrally and conspicuously placed on pages to capture the user's attention. By investigating a large number of web pages, first, data regions are always located in the centre section horizontally on Web pages. Second, the size of a data region is usually large when there are enough data records in the data region. The actual size of a data region may change greatly because it is not only influenced by the number of data records retrieved, but also by what information is included in each data record.

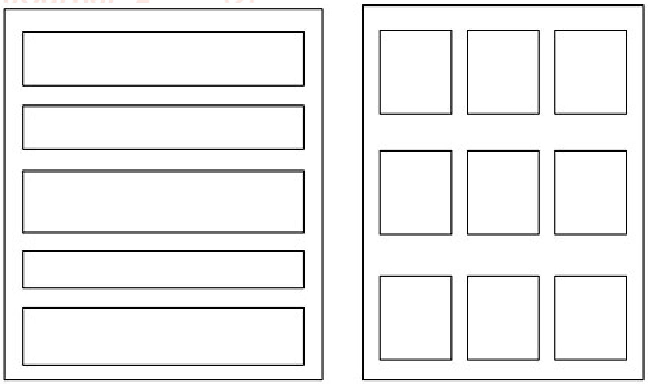

Figure2. Layout models of data records on web pages.

Layout features (LFs). These features indicate how the data records in the data region are typically arranged.

LF1 : The data records are usually aligned flush left in the data region.

LF2 : All data records are adjoining.

LF3 : Adjoining data records do not overlap, and the space between any two adjoining records is the same.

Data records are usually presented in one of the two layout models shown in Fig. 3. In Model 1, the data records are arranged in a single column evenly, though they may be different in width and height. LF1 implies that the data records have the same distance to the left boundary of the data region. In Model 2, data records are arranged in multiple columns, and the data records in the same column have the same distance to the left boundary of the data region. Because most Web pages follow the first model, we only focus on the first model in this paper. In addition, data records do not overlap, which means that the regions of different data records can be separated.

Appearance features (AFs). These features capture the visual features within data records.

AF1 : Data records are very similar in their appearances, and the similarity includes the number of the images they contain and the fonts they use.

AF2 : The data items of the same semantic in different data records have similar presentations with respect to position, size (image data item), and font (text data item).

AF3 : The neighboring text data items of different semantics often use distinguishable fonts.

AF1 describes the visual similarity at the data record level. Generally, there are three types of data contents in data records, i.e., images, plain texts (the texts without hyper links), and link texts (the texts with hyperlinks). AF2 and AF3 describe the visual similarity at the data item level. The text data items of the same semantic always use the same font, and the image data items of the same semantic are often similar in size. AF3 indicates that the neighboring text data items of different semantics often use distinguishable fonts.

3.2. DOM Tree Generation and Clean useless node (Preprocessing)

To begin with, a DOM tree should be generated from html tags of the page. At the same time, features/attributes about the node should be included in these tree nodes, respectively; described in next section.

Pre-processing is necessary in order to clean HTML pages, e.g., to remove header details, scripts, styles, comments, hidden tags, space, tag properties, empty tags, etc. In this step, the white relax function of the jsoup parsing tool for removing cleaning HTML tag. First of all, we need to eliminate these nodes to get clean Html page for further processing.

3.3. Filtering Noisy Block

Web page designers tend to organize their content in a reasonable way: giving prominence to important things and deemphasizing the unimportant parts with proper features such as position, size, color, word, image, link, etc. All of product page features are related to the importance. For example, an advertisement may contain only images but no texts, a contact information bar may contain email, and a navigation bar may contain quite a few hyperlinks. However, these features have to be normalized by the feature values of the whole page to reflect the image of the whole page. For example, the LinkNum of a block should be normalized by the link numbers of the whole page. Then all these features are formulated with equation (1).

$$fi(a) = \frac{\text{number of attributes in block i}}{\text{number of these attributes in whole page}} \quad (1)$$

Firstly, some conclusions are given on product features. All those conclusions are according to the observation of product list page on web site.

1. If a block contains email elements, then it is entirely possible a contact block.
2. If TextLen/LinkTextLen<threshold, then it is quite possible a hub block [7].
3. If <p> is included in a block, then this block is possible authority block[7].
4. If the normalized LinkNum > threshold, then it is quite possible a hub block.

Accordingly, these rules are calculated into equation (2) and then F indicates the possibility of noisy block.

$$F=\sum \infty_i.f_i(b)=\alpha_1.f_{email}(b)+\alpha_2.f_{textlen/linktexlen}(b)+\ldots+\alpha_4.f_{links}(b), \sum\alpha_i=1 \quad (2)$$

Where ∞i is coefficient, we can set different weights on block importance respectively. Additionally, all these parameters can be adjusted to adapt to different conditions. Finally regarding product features, an important block is extracted for further processing. Consequently, filtering noisy blocks can decrease the complexity of web information extraction through narrowing down the processing scope.

3.4. Blocks Clustering for Data Region Identification

The blocks in the data region are clustered based on their appearance similarity. Since there are three kinds of information in data records, i.e., images, plain text and link text, the appearance similarity of blocks is computed from the three aspects. For images, we care about the size; for plain text and link text, we care about the shared fonts. Intuitively, if two blocks are more similar on image size, font, they should be more similar in appearance. The appearance similarity formula between two blocks b1 and b2 is given below:

sim(b1,b2)=Wi * simImg(b1,b2)+Wpt * simPT(b1,b2) +Wlt * simLT(b1,b2) (3)

Where simImg(b1,b2), simPT(b1,b2), and simLT(b1,b2) are the similarity based on image size , plain text , and link text. Wi, Wpt, and Wlt are the weights of these similarities. Table 1 gives the formulas to compute the component similarities and the weights in different cases.

Table1. The formulas of block appearance similarity and the weights in different cases

Formulas	Descriptions
$simImg(b1,b2) = \frac{Min\{sa_i(b_1),sa_i(b_2)\}}{Max\{sa_i(b_1),sa_i(b_2)\}}$	sai(b)is total number of images in block b.
$W_i = \frac{sa_i(b_1)+sa_i(b_2)}{sa_b(b_1)+sa_b(b_2)}$	sab(b) is the total number of block b.
$simPT(b1,b2) = \frac{Min\{fn_{pt}(b_1),fn_{pt}(b_2)\}}{Max\{fn_{pt}(b_1),fn_{pt}(b_2)\}}$	fnpt(b) is the total number of fonts of the plain texts in block b.
$W_{pt} = \frac{sa_{pt}(b_1)+sa_{pt}(b_2)}{sa_b(b_1)+sa_b(b_2)}$	sapt(b) is the total number of the plain texts in block b.
$simLT(b1,b2) = \frac{Min\{fn_{lt}(b_1),fn_{lt}(b_2)\}}{Max\{fn_{lt}(b_1),fn_{lt}(b_2)\}}$	fnlt(b) is the total number of fonts of the link texts in block b.
$W_{lt} = \frac{sa_{lt}(b_1)+sa_{lt}(b_2)}{sa_b(b_1)+sa_b(b_2)}$	salt(b) is the total number of the link text in block b.

International Journal of Trend in Scientific Research and Development IJTSRD ISSN: 2456-6470

Our block clustering method consists of two steps: The first one is to build clusters by computing the similarity among blocks. The similarity sim(b1,b2) between two blocks bi and bj is computed by the equation (3).The second one is to merge the resulting clusters. The threshold is trained from sample page. So the cluster building procedure is simplified as follows:

```
Procedure BlockClustering

Put all the blocks bi into the pool;
FOR(every block bi in pool){
compute the appearance similarity sim(bi,bj) bet: two blocks
IF(sim(bi,bj) >threshold){
group bi and bj into a new cluster;
delete bi and bj from the pool;
 }
ELSE{
create a new cluster for bi;
delete bi from the pool;
}
}
```

The second step is to merge clusters. To determine if two clusters must be merged, we define the cluster similarity simCkl between two clusters Ck and Cl as the maximum value of sim(bi,bj), for every two blocks bi∈Ck and bj∈Cl.

```
Procedure BlockMerging
FOR(every cluster Ck)
{
compute the simCkl with other clusters;
IF(simCkl >threshold){
clusters Ck and Cl are merged;
}
}
```

4. EXPERIMENTAL RESULTS

Our experiments were testing using commercial book store web sites collected from different web site in Table 2. The system takes as input raw HTML pages containing multiple data records. The measure of our method are based on three factors, the number of actual data records to be extracted, the number of extracted data records from the list page, and the number of correct data records extracted from the list page. Based on these three values, precision and recall are calculated according to the formulas:

Recall=Correct/Actual*100

Precision=Correct/Extracted*100

According to above measurement, we tested web pages from various book store web sites and check each page by manually.

Table2. Results for selected Web Site

URL	Precision	Recall
gobookshopping.com	100	98
yahoo.com	98	97
allbooks4less.com	100	98
amazon.com	92.4	87.5
barnes&nobels.com	98	90
Average	97.68	94.1

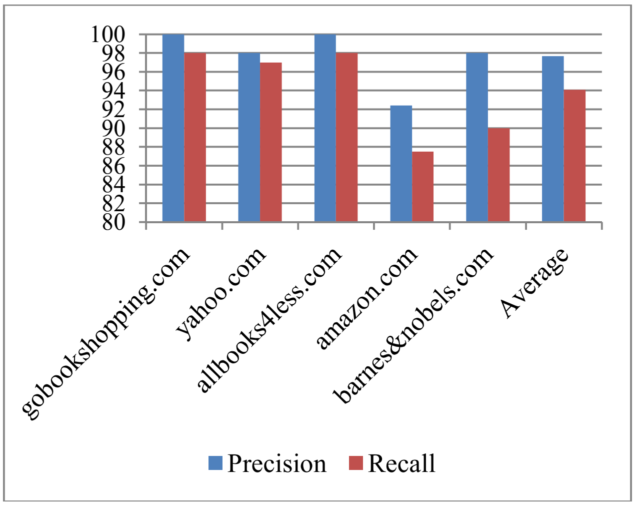

Figure3. Result chart for selected web sites

5. CONCLUSION

In this paper, we have presented extraction information content from semantic structure of HTML documents. It relays on the observation that the appearance similarity of data record in web page. Firstly, we segment a web page into several raw chunks. Second, filter the noisy block. Then proposed block clustering method groups remaining blocks with their appearance similarity for data region identification. Our method is automatic and it generates a reliable and accurate wrapper for web data integration purpose. In this case, neither prior knowledge of the input HTML page nor any training set is required. We experiment on multiple web sites to evaluate our method and the results prove the approach to be promising.

Acknowledgment

I would like to greatly thank my supervisor, Dr. Thi Thi Soe Nyunt, Professor of the University of Computer Studies, Yangon, for her valuable advice, helpful comments, her precious time and pointing me in the right direction. I also want to thank Daw Aye Aye Khaing, Associate Professor and Head of English Department and Daw Yu Yu Hlaing, Associate Professor of English Department.

References

[1]. B Liu, R. Grossman and Y. Zhai, “Mining Data Records in Web Pages”, ACM SIGKDD Conference, 2003.

[2]. B Liu and Y. Zhai, “NET – A System for Extracting Web Data from Flat and Nested Data Records”, WISE Conference, 2005.

[3]. Cai, D., Yu, S., Wen, J.-R. and Ma, W.-Y., VIPS: a vision-based page segmentation algorithm, Microsoft Technical Report.

[4]. Chang, C-H., Lui, S-L. “IEPAD: Information Extraction Based on Pattern Discovery”, WWW-01, 2001.

[5]. Chang, C.-H., Kayed, M., Girgis, M., and Shaalan, K. (2006). “A survey of web information extraction systems”, IEEE Transactions on Knowledge and Data Engineering, 18(10):1411–1428.

[6]. Crescenzi, V. and Mecca, G. “Automatic information extraction from large websites”, Journal of the ACM, 2004, 51(5):731–779.

[7]. D. Cai, H. Xiaofei, W. Ji-Rong, and M. Wei-Ying, “Block-level Link Analysis”, SIGIR'04, July 25-29, 2004.

[8]. H. Zhao, W. Meng, Z. Wu, V. Raghavan, C. Yu, "Fully Automatic Wrapper Generation for Search Engines", WWW Conference, 2005.

[9]. J. Hammer, H. Garcia Molina, J. Cho, and A. Crespo, "Extracting semi-structured information from the web", In Proceeding of the Workshop on the Management of Semi-structured Data, 1997.

[10]. Kushmerick, N, "Wrapper Induction: Efficiency and Expressiveness. Artificial Intelligence", 118:15-68, 2000.

[11]. M. Kayed, C.-H. Chang, "FiVaTech: Page-Level Web Data Extraction from Template Pages", IEEE TKDE, vol. 22, no. 2, pp. 249-263, Feb. 2010.

[12]. Shian-Hua Lin, Jan-Ming Ho, "Discovering Informative Content Blocks from Web Documents", IEEE Transactions on Knowledge and Data Engineering, page 41-45, Jan, 2004.

[13]. Yang, Y. and Zhang, H. "HTML page analysis based on visual cues", Z Niu, LiuLing Dai,YuMing Zhao, "Extraction of Informative Blocks from web pages", in the Proceedings of International Conference on Advanced Language Processing and Web Information Technology, 2008.

[14]. YuJuan Cao, ZhenDong Niu, LiuLing Dai,YuMing Zhao, "Extraction of Informative Blocks from web pages", in the Proceedings of International Conference on Advanced Language Processing and Web Information Technology, 2008.

[15]. Y. Zhai, and B. Liu, "Web Data Extraction Based on Partial Tree Alignment", WWW Conference, 2005.CA: University Science, 1989.

International Journal of Trend in Scientific Research and Development
IJTSRD
International Journal of Trend in Scientific Research and Development
ISSN: 2456-6470