

Extraction of Flat and Nested Data Records from Web Pages

Siddu P Algur¹ and P S Hiremath²

¹Dept. of Info. Sc. & Engg., SDM College of Engg & Tech, Dharwad, Karnataka, India
siddu_p_algur@hotmail.com

²Dept. of Computer Science, Gulbarga University, Gulbarga, Karnataka, India
hiremathps@yahoo.co.in

Abstract

This paper deals with studies the problem of identification and extraction of flat and nested data records from a given web page. With the explosive growth of information sources available on the World Wide Web, it has become increasingly difficult to identify the relevant pieces of information, since web pages are often cluttered with irrelevant content like advertisements, navigation-panels, copyright notices etc., surrounding the main content of the web page. Hence, it is useful to mine such data regions and data records in order to extract information from such web pages to provide value-added services. Currently available automatic techniques to mine data regions and data records from web pages are still unsatisfactory because of their poor performance. In this paper, we propose a new method to identify and extract the data records from the web pages automatically. Given a page, the proposed technique first identifies the data region based on the visual clue information. It then extracts each record from the data region and identifies it whether it is a flat or nested record based on visual information – the area covered by and the number of data items present in each record. The experimental results show that the proposed technique is effective and better than existing techniques.

Keywords: Web mining, Web data regions, Web data records

1. Introduction

Many companies manage their business and publish their products and services on the Web. Collection and organization of this dynamic information can produce the data for many value-added applications. In order to collate and compare the prices and features of products available from the various Web sites, we need tools to extract attribute descriptions of each product (called data object) within a specific region (called data region) in a pages.

As illustrated in Fig. 1, there are many irrelevant components intertwined with the descriptions of data objects in web pages. These items include advertisement bar, product category, search panel, navigator bar, and copyright statement. In many web pages, there are

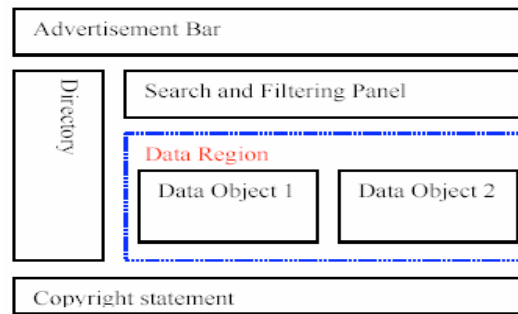


Fig 1: A schematic view of a webpage

normally more than one data object intertwined together in a data region. Furthermore, the raw source of the web page for depicting the objects might be non-contiguous. So it is difficult to discover the attributes for each object.

In real applications, what the users want from complex web pages is the description of individual data object derived from the partitioning of data region. There are several approaches by Hammer, Garcia Molina, Cho, and Crespo (1997), Kushmerick (2000), Chang and Lui (2001), Crescenzi, Mecca, and Merialdo (2001), Zhao, Meng, Wu and Raghavan (2005) proposed in the literature to address the problem of web data extraction, which are called wrapper generation.

The first approach by Hammer, Garcia Molina, Cho, and Crespo (1997) 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 and thus does not scale to a large number of sites.

The second approach Kushmerick (2000) 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. An example of wrapper induction systems is WEIN by Baeza Yates (1989).

The third approach Chang and Lui (2001) is the automatic approach. Since structured data objects on the web are normally database records retrieved from underlying web databases and displayed in web pages with some fixed templates, automatic methods aim to find

Copyright (c) 2006, Australian Computer Society, Inc. This paper appeared at the Australasian Data Mining Conference (AusDM 2006), Sydney, December 2006. Conferences in Research and Practice in Information Technology, Vol. 61. Peter Christen, Paul Kennedy, Jiuyong Li, Simeon Simoff and Graham Williams, Eds. Reproduction for academic, not-for-profit purposes permitted provided this text is included.

patterns/grammars from the web pages and then use them to extract data. Examples of automatic systems – IEPAD by Chang and Lui (2001), ROADRUNNER by Crescenzi, Mecca, and Merialdo (2001).

The fourth approach is MDR by Liu, Grossman, and Zhai (2003) which basically exploits the regularities in the HTML tag structure directly. It is often very difficult to derive accurate wrappers entirely based on HTML tags. The MDR algorithm makes use of the HTML tag tree of the web page to extract data records from the page. However, erroneous tags in the HTML source pages may result in building of incorrect trees, which in turn makes it impossible to extract data records correctly. MDR has several other limitations which will be discussed in the latter half of this paper. DEPTA by Zhai, and Liu (2005) uses visual information (locations on the screen at which the tags are rendered) 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 far as the browser is able to render the page correctly. The computation time for constructing the tag tree is also an overhead. Further, this method also fails to identify some of the data records.

NET by Benchalli, Hiremath, Siddu, and Renuka (2005) extracts data from web pages that contain a set of flat or nested data records automatically in two steps. This approach also depends on building of tag tree and post order traversal of the tag tree to identify data records at different levels.

We propose a novel and more effective method to extract data records from web pages that contain a set of flat or nested data records automatically. Our method is called **ENDR (Extraction of Flat and Nested Data Records from Web Pages)**. The experimental results show that the proposed technique is more effective than existing techniques substantially.

2. Related Work

Extraction the regularly structured flat or nested data records from a web page is an important problem. So far, some attempts have been made to deal with the problem. For automatic extraction, in Crescenzi, Mecca, and Merialdo (2001), Zhao, Meng, Wu and Raghavan (2005), Lerman, Getoor, Minton, and Knoblock (2004), it is proposed to find patterns or grammars from multiple pages containing similar data records. They require an initial set of pages containing similar data records which is, however, a limitation. In Lerman, Getoor, Minton and Knoblock, (2004), it proposes a method that tries to explore the detail information pages behind the current page to segment the data records. The need for such detail pages is a drawback because many data records do not have such pages or perhaps such pages are hard to find. In Chang and Lui (2001), string matching method is studied. However, it could not find nested data records. A similar method is proposed in Wang, Lochovsky (2003). Liu, Grossman, and Zhai (2003) and Zhao, Meng, Wu and Raghavan (2005), it some algorithms are proposed to identify data records, but they do not extract data items from the data records and do not handle nested data records. DEPTA by Zhai and Liu (2005) is able to align

and extract data items from the data records but does not handle nested data records.

The NET by Benchalli, Hiremath, Siddu and Renuka (2005) (Nested data Extraction using Tree matching) works in two main steps:

(i) Building a tag tree of the page: Due to numerous tags and unbalanced tags in the HTML code of the page, building a correct tag tree is a complex task. A Visual based method is used to deal with this problem.

(ii) Identifying data records and extracting data from them: The algorithm performs a post order traversal of tag tree to identify data records at different levels. This ensures that nested data records are found. The tree edit distance algorithm and visual clues are used to perform these task.

Though the technique is able to extract the flat or nested data records, construction of tag tree and its post order traversal is consider to be an overhead.

The above automatic methods are inaccurate, tag dependant, incorporate time-consuming tag tree construction, and are based many assumptions which do not always hold good for all web pages. The proposed method does not make such assumptions and can scale well for almost all web pages. It is also independent of the type of tags and dispenses with the time-consuming tag tree construction procedure.

3. Data Region Extraction

We now start to present our proposed technique .This section focuses on the extraction of Data Region from the web page. The extraction of the data records and extraction of the data items in the data records will be the topic of the next section. Since this step is an improvement of our previous technique VSAP by Benchalli, Hiremath, Siddu, and Renuka (2005), we give a brief overview of the VSAP algorithm.

3.1. The Basic Idea of VSAP

An effective method to mine the data region in a web page automatically is the VSAP by Benchalli, Hiremath, Siddu and Renuka (2005). The visual information (i.e., the locations on the screen at which tags are rendered) helps the system to identify gaps that separate data records, and, thus, helps to segment data records correctly, because the gap within a data record (if any) is typically smaller than that in between data records. Also, by the visual structure analysis of the web pages, it can be observed that the relevant data region seems to occupy the major central portion of the web page.

The VSAP technique works as follows in two steps:

Step 1) Determination of the co-ordinates of all bounding rectangles in the web page

The first step of the VSAP technique determines the co-ordinates of all the bounding rectangles in the web page. The rendering engine of the browser produces the boundary coordinates. A bounding rectangle is constructed by obtaining the co-ordinate of the top-left corner of the tag, the height and the width of that tag. The left and top co-ordinates of the tag are obtained from the

offsetLeft and offsetTop properties of the HTMLObjectElement.



Fig 2 A sample web page of a product related website

Step 2) Data Region Identification

The second step of the VSAP technique is to identify the data regions of the web page. There are 3 steps involved in identifying the data region:

a) Identify the largest rectangle.

Based on the height and width of bounding rectangles obtained in the previous step, the area of the bounding rectangles of each of the children of the BODY tag are determined. Then the largest rectangle amongst these bounding rectangles is found. The reason for doing this is due to the observation that the largest bounding rectangle will always contain the most relevant data in that web page. Thus, by determining the largest rectangle, a superset of the data region is obtained.

b) Identify the container within the largest rectangle.

Once the largest rectangle is obtained, a set of all the bounding rectangles whose area is more than half the area of the largest rectangle is formed. The rationale behind this is that the most important data of a web page must occupy a significant portion of the web page. Then the bounding rectangle having the smallest area in this set is found. The reason for determining the smallest rectangle within this set is that the smallest rectangle will only contain data records. Thus a *container* is obtained, which contains the data region and some irrelevant data.

c) Identify the data region containing the data records within this container

To filter the irrelevant data from the container, a *filter* is used. The filter determines the average height of children within the container. Those children whose heights are less than the average height are identified as irrelevant data and are filtered off. The outcome of the filter is a data region. The data regions of the web page in Fig 2 are shown in the Fig 3.

4. The Proposed Technique

We propose a more effective method to extract flat or nested data records from a given web page automatically. The method is called ENDR (Extraction of Flat and Nested Data Records from Web Pages). Before presenting the method, we discuss three observations about data records in web pages, which simplify the extraction task. These observations were made in Benchalli, Hiremath, Siddu, and Renuka (2005).



Fig 3. Filtered Data Region

- A group of data records, that contains the descriptions of a set of similar observations, is typically presented in contiguous region of a page.
- The area covered by rectangle that bounds the data region is more than the area covered by rectangles bounding other regions. eg., advertisements and links.
- The height of irrelevant data records within a collection of data records is less than the average height of relevant data records within that region.

The experimental results show that these observations are true.

Definition 1: A *flat data* record is defined as a collection of data items that together represents a single meaningful entity.

eg., the product having single size, look, price etc.,

Definition 2: A *nested data record* is defined as one that provides multiple description of the same entity.

eg., the same type of products but different sizes, looks, prices etc.,

The Fig.4 illustrates an example, which is a segment of a web page that shows flat and nested data records.

The system model of the **ENDR (Extraction of Flat and Nested Data Records)** technique is shown in Fig.5.

When a web page having description of products is given to VSAP, it identifies and extracts the data region. All the noises of a given web page are eliminated using filter. The filtered data region corresponding to the figure is shown in Fig.6.

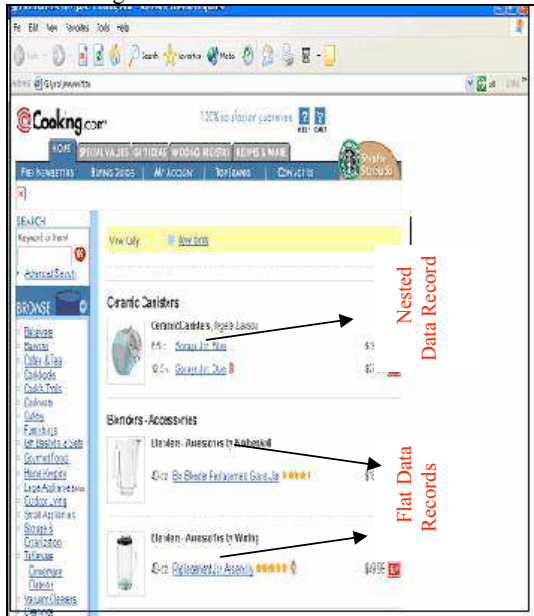


Fig. 4 An example of flat and nested data records

The filtered data region is given as the input to our system which extracts the flat and nested data records from the given data region and extracts data fields from the identified records.

4.1 Extraction of data records

Extraction of data records is based on visual clues. In the first step of the proposed technique, we determine the height of all the data records. This approach uses the MSHTML parsing and rendering engine that gives the height of each data record. The height of the data record is obtained from the offsetHeight property of the HTMLObjectElement. Next, the average height of the records is calculated. The average height of all the records provides the approximate height of each record. The height of each data record is compared with the average height. If the height of the child is greater than or equal to the average height, then the data record is extracted.

The procedure Extract Data Record extracts the flat and nested records from given data region. It is as follows.

```

Procedure
ExtractDataRecord(dataRegion)
{
    THeight=0
    For each child of dataRegion
    BEGIN

```

```

        THeight += height of the bounding
                    rectangle of child
    END
    AHeight = THeight/no of children of
dataRegion
    For each child of dataRegion
    BEGIN
        If height of child's bounding
rectangle > AHeight
        BEGIN
            dataRecord=child
        END
    END
}

```

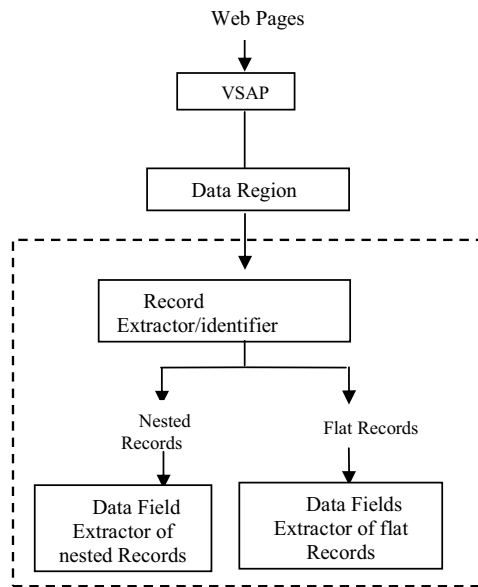


Fig. 5 System Model

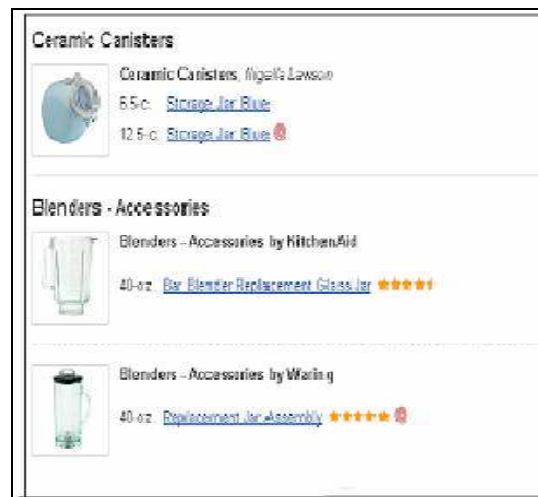


Fig. 6 Filtered data region

The Fig. 7 shows extracted data records from the data region shown in the Fig.6.



Fig.7 Extracted data records

4.2 Identification of data records

Identification of data records, as flat or nested, is essential in order to simplify the task of extracting the data items, which is very useful for various applications as mentioned earlier.

This technique determines the data fields for each data record within the data region. Various tags such as <TD>, <TR>, <A>, , represent the data fields. By counting these tags as they are encountered, the number of fields is obtained. The flat record gives description of a single entity, whereas the nested data record gives multiple description of a single entity, so the data fields in flat records are less as compared to that of nested records. Experimental observations have shown that the number of fields in the nested data records is at least 40% (approx) more than that of the flat records. The number of fields in the first record is compared with the number of fields in the next record. If the number of fields is more than 40%, then it is a nested record else it is a flat record. Suppose a condition is encountered where the number of fields is equal then in both cases. Then the record is compared with the third record and so on until the condition is satisfied.

The procedure IdentifyNestedData identifies whether the record is flat or nested based on the number of data items present in the data record. It is as follows:

Procedure

```

IdentifyNestedData (dataRecord[I],
dataRecord[I+1])
{
    noofField[I]=0
    For I 1 to no of records
        BEGIN
            noofFields [I]=
                noofFields[I]+noofFields
                in the record[I]
        END
    DO
        For I 1 to no of records
            BEGIN

```

```

For dataRecord [I], dataRecord[I+1]
    IF the no of fields in the
    [I+1]th record>=40% of the no of
    fields in the [I]th record
    Then [I+1]th record is a
        nested data record
    ELSE
        The [I]th record is a nested
        data record
    END
WHILE (EOF)
}

```



Fig. 8

- (a) Identified nested data record, No. of data fields=12
(b) Identified flat data record, No. of data fields = 7

The Fig. 8 shows the identified nested and flat data records. In Fig.8 (a), the number of data fields is 12 and in Fig.8 (b) the number of data fields is 7. The number of data fields in Fig 8(a) is 58.3% more than the number of data fields in Fig 8(b).

5. Empirical evaluation and experimental results

In this section, we evaluate the proposed ENDR (Extraction of Flat and Nested Data Records from Web Pages) technique. We compare it with the state-of-the-art existing system NET by Bing and Yanhong (2005). We do not compare it with DEPTA by Zhai, and Liu (2005) here as it is shown that NET is better than DEPTA. For flat nested data records, the proposed method performs very well. The experimental results are given in Table 1.

Column 1 lists the site of each test page. Due to the space limitations, we have not listed all the URL's considered for experimentation. We have not considered many erroneous pages for testing because such pages are relatively rare and quite difficult to find.

Column 2 and 4 give the number of data items extracted wrongly (Wr) by NET and proposed method from each page respectively. In x/y, x is the number of extracted results that are incorrect and y is the number of results that are not extracted. Columns 3 and 5 give the numbers

of correct (Corr) data items extracted by NET and proposed method from each page respectively. Here, in x/y , x is the number of correct items extracted and y is number of items in the page. From the table, we observe that, for flat and nested data records, the proposed method performs better than the other. The precision and recalls are computed based on extraction performed on all test pages.

| URL | NET | | ENDR | |
|-------------------------------|---------------|-------|---------------|-------|
| | Wr. | Corr. | Wr. | Corr. |
| Without Nesting | | | | |
| http://www.bookpool.com | 0/0 | 15/15 | 0/0 | 15/15 |
| http://www.amazon.com | 0/0 | 22/22 | 0/0 | 22/22 |
| http://www.shopping.com | 0/0 | 20/20 | 0/0 | 20/20 |
| http://www.barnesandnoble.com | 0/0 | 10/10 | 0/0 | 10/10 |
| http://www.cooking.com | 0/0 | 28/29 | 0/0 | 29/29 |
| http://tigerdirect.com | 0/0 | 12/14 | 0/0 | 13/14 |
| http://www.lmart.com | 0/0 | 70/70 | 0/0 | 70/70 |
| Recall | 97.15% | | 98.99% | |
| Precision | 99.3% | | 98.92% | |
| With Nesting | | | | |
| http://www.amazon.com | 0/0 | 22/25 | 0/0 | 25/25 |
| http://lmart.com | 1/0 | 42/43 | 0/0 | 43/43 |
| http://www.cooking.com | 1/0 | 62/63 | 0/0 | 62/63 |
| Recall | 98.63% | | 100% | |
| Precision | 99% | | 100% | |

Table 1: Experimental Results

6. Conclusion

In this paper, we have proposed a more effective technique to perform the automatic extraction of flat and nested data records from the web pages. Given a web page, the proposed method first identifies correct data region based on visual clue information. It counts the number of the data items in each record and then identifies the record as either flat or nested. Although the problem has been studied by several researchers, existing techniques are either inaccurate or make many strong assumptions. The proposed method is a pure visual clue based extraction of flat and nested data records. Experimental results show that the method performs data extraction more effectively.

7. References

- A. Arasu, H. Garcia-Molina, (2003): Extracting structured data from web pages, ACM SIGMOD 2003,
- Baeza Yates(1989): R. Algorithms for string matching: A survey. ACM SIGIR Forum, 23(3-4):34—58.
- Benchalli, Hiremath, Siddu and Renuka 2005): "Mining Data Regions from Web Pages" , COMAD2005b, DEC.
- Bing Liu , Kevin chen-chuan chang(2002): Editorial: Special issue on web content mining, WWW 02.
- Bing Liu and Yanhong Zhai(2005): "NET - A System for Extracting Web Data from Flat and Nested Data Records." Proceedings of 6th International Conference on Web Information Systems Engineering (WISE-05).
- Cai, D., Yu, S., Wen, J.-R. and Ma, W.-Y. (2003). Extracting Content Structure for Web Pages based on Visual Representation, Asia Pacific Web Conference (APWeb 2003), pp. 406417.
- Chang, C-H., Lui, S-L(2001): IEPAD Information Extraction Based on Pattern Discovery. WWW-01.
- Crescenzi, V., Mecca, G. and Meriardo, P, (2001): ROADRUNNER: Towards Automatic Data Extraction from Large Web Sites. VLDB-01.
- D. Buttler, L. Liu, C. Pu. (2001): A Fully Automated Object Extraction System for the World Wide Web. International Conference on Distributed Computing Systems (ICDCS 2001).
- D. Embley, Y. Jiang, and Y. K . Ng(1999): Record-boundary discovery in Web documents. ACM SIGMOD Conference.
- Eying, H. Zhang, (2001): HTML Page Analysis based on Visual Cues. 6th International Conference on Document Analysis and Recognition.
- H. Zhao, W. Meng, Z. Wu, Raghavan (2005): Clement Yu. Fully Automatic Wrapper Generation For Search Engines, International WWW conference 2005, May 10-14, Japan. ACM 1-59593-046-9/05/005
- J. Hammer, H. Garcia Molina, J. Cho, and A. Crespo (1997): Extracting semi-structured information from the web.In Proc.of the Workshop on the Management of Semi-structured Data.
- J. Wang, F. H Lochovsky (2003): Data Extraction and Label Assignment for Web Databases.WWW conference,.
- Kushmerick, N(2000): Wrapper Induction Efficiency and Expressiveness. Artificial Intelligence, 118:15-68, Clustering-based Approach to Integrating Source Query]
- Lerman, K., Getoor L., Minton, S. and Knoblock, C(2004): Using the Structure of Web Sites for Automatic Segmentation of Tables. SIGMOD'04.
- Liu, B., Grossman, R. and Zhai, Y(2003): Mining Data Records in Web Pages. KDD-03.
- Zhai, Y., Liu, B(2005): Web Data Extraction Based on Partial Tree Alignment , WWW-05, 2005, May 10-14, , Chiba, Japan. ACM 1-59593-046-9/05/00