ISSN:0975-9646

Extracting Content Structure from Web Pages by Applying Vision based Approach

Prof.**K.S.Kadam,** Prof.**A.B.majgave,** Prof.**Y.M.Kamble**

Computer Science and Engineering Department,
Textile and Engineering Institute, Ichalkaranji, India

Abstract: This paper studies the problem of extracting data records on the response pages returned from web databases or search engines. Existing solutions to this problem are based primarily on analyzing the HTML DOM trees and tags of the response pages. While these solutions can achieve good results, they are too heavily dependent on the specifics of HTML.The World Wide Web has several online databases and the number keeps growing every day. The data in the web pages is generally wrapped in the form of data records. Such web pages are generated dynamically. This paper focuses on extracting the data from the web pages. Till today, several techniques are proposed for retrieving information from web pages but all suffer the common problem. The problem is dependence on programming language used to design the web pages. So this paper focuses on utilizing the visual features of web page for extracting the data from the deep web pages. To make the system efficient, it can be combined with non-visual information like the symbols and tags. Approach of this paper is independent of any specific web programming language and hence it can be extended to various web pages which have different underlying architecture.

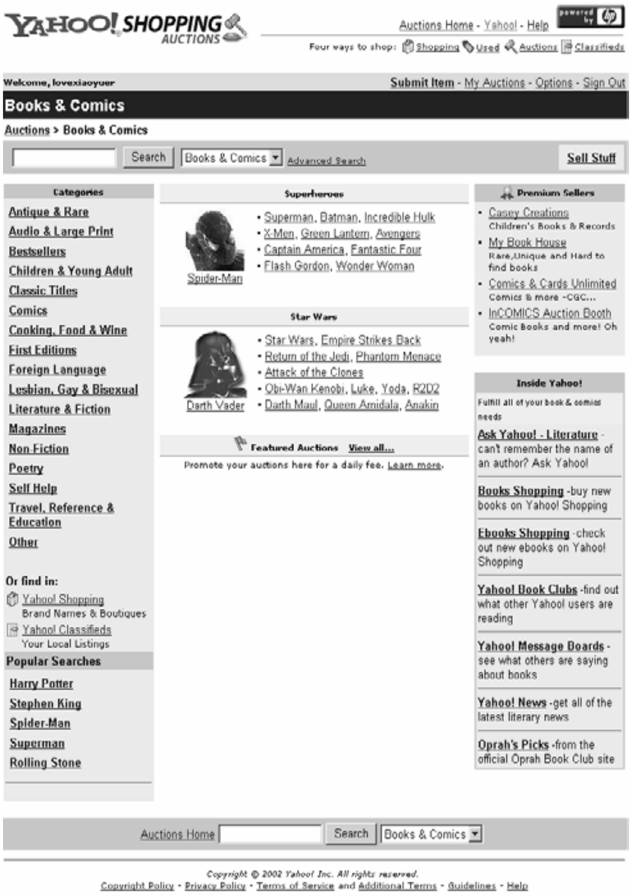
Keywords: Web mining, Web data extraction, visual features of deep Web pages, wrapper generation

I. INTRODUCTION

Today the Web has become the largest information source for people. Most information retrieval systems on the Web consider web pages as the smallest and undividable units, but a web page as a whole may not be appropriate to represent a single semantic. A web page usually contains various contents such as navigation, decoration, interaction and contact information, which are not related to the topic of the web page. Furthermore, a web page often contains multiple topics that are not necessarily relevant to each other. Therefore, detecting the content structure of a web page could potentially improve the performance of web information retrieval.

Many web applications can utilize the content structures of web pages. For example, some researchers have been trying to use database techniques and build wrappers for web documents [3]. If a web page can be divided into semantic related parts, wrappers can be more easily matched and data can be more likely extracted. Link structure analysis can also make good use of the content structures of web pages. Links at different parts of a page usually act as different functions and contribute to the Page- Rank or HITS differently. Recent works on topic distillation [4] and focused crawling [6] show the usefulness of page segmentation on information analysis.
Furthermore, adaptive content delivery on small handheld devices also requires the detection of underlying content structure of a web page to facilitate the browsing of a large page by partitioning it into smaller units.

In this paper, we propose VIPS (**Vi**sion-based **P**age **S**egmentation) algorithm to extract the content structure for a web page. The algorithm makes full use of page layout features and tries to partition the page at the semantic level. Each node in the extracted content structure will correspond to a block of coherent content in the original page


YAHOO! SHOPPING AUCTIONS
Auctions Home - Yahoo! - Help
powered by hp
Four ways to shop: Shopping Used Auctions Classifieds
Welcome, lovexiaoyuer
Submit Item - My Auctions - Options - Sign Out
Books & Comics
Auctions > Books & Comics
Search
Books & Comics
Advanced Search
Sell Stuff

Categories

Antique & Rare
Audio & Large Print
Bestsellers
Children & Young Adult
Classic Titles
Comics
Cooking, Food & Wine
First Editions
Foreign Language
Lesbian, Gay & Bisexual
Literature & Fiction
Magazines
Non-Fiction
Poetry
Self Help
Travel, Reference & Education
Other

Or find in:
Yahoo! Shopping
Brand Names & Boutiques
Yahoo! Classifieds
Your Local Listings

Popular Searches

Harry Potter
Stephen King
Spider-Man
Superman
Rolling Stone

Superheroes

Spider-Man
- Superman, Batman, Incredible Hulk
- X-Men, Green Lantern, Avengers
- Captain America, Fantastic Four
- Flash Gordon, Wonder Woman

Star Wars

Darth Vader
- Star Wars, Empire Strikes Back
- Return of the Jedi, Phantom Menace
- Attack of the Clones
- Obi-Wan Kenobi, Luke, Yoda, R2D2
- Darth Maul, Queen Amidala, Anakin

Featured Auctions View all...
Promote your auctions here for a daily fee. Learn more.

Premium Sellers

- Casey Creations
  Children's Books & Records
- My Book House
  Rare,Unique and Hard to find books
- Comics & Cards Unlimited
  Comics & more -CGC...
- InCOMICS Auction Booth
  Comic Books and more! Oh yeah!

Inside Yahoo!

Fulfill all of your book & comics needs

Ask Yahoo! - Literature - can't remember the name of an author? Ask Yahoo!

Books Shopping -buy new books on Yahoo! Shopping

Ebooks Shopping -check out new ebooks on Yahoo! Shopping

Yahoo! Book Clubs -find out what other Yahoo! users are reading

Yahoo! Message Boards - see what others are saying about books

Yahoo! News -get all of the latest literary news

Oprah's Picks -from the official Oprah Book Club site

Copyright © 2002 Yahoo! Inc. All rights reserved.
Copyright Policy - Privacy Policy - Terms of Service and Additional Terms - Guidelines - Help

Fig.1. An example Web page from Yahoo shopping.

II.RELATED WORK

Related works to ours are in the area of wrapper generation. A wrapper is a program that extracts data from a Web site or page and put them in a database.

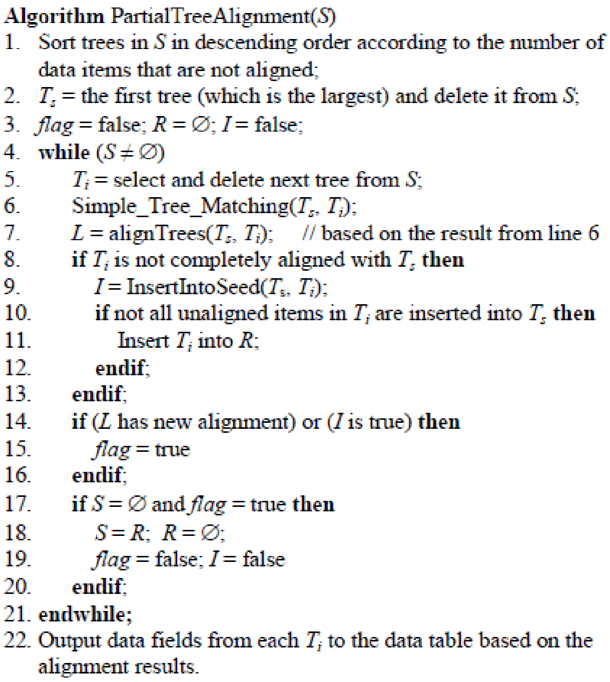
1] The first approach is wrapper induction, which uses supervised learning to learn data extraction rules from a set of manually labeled positive and negative examples. Manual labeling of data is, however, labor intensive and time consuming.

2] The second approach is automatic extraction. In [1], a study is made to automatically identify data record boundaries. The method is based on a set of heuristic rules, e.g. highest-count tags, repeating-tags and ontology-matching. [2] proposes a few more heuristics to perform the task without using domain ontology. However, [3] shows that these methods produce poor results. In addition, these methods do not extract data from data records.

3] Paper [5] proposes another method for data extraction. Its main idea is to utilize the detailed data in the page behind the current page to identify data records. It is common that a page with multiple data records does not contain the complete information of each data record. Instead, a link is normally used to point to the page with complete details. For example, a product record normally has a link pointing to the page that contains the detailed description of the product.

4] Another approach is the MDR algorithm which only identifies data records but does not align or extract data items from the data records. Thus it only performs the first step of our task.

5] Another one approach is to extract web page data by applying Partial Tree Alignment technique. This approach aligns multiple tag trees by progressively growing a seed (tag) tree. The seed tree, denoted by Ts, is initially picked to be the tree with the maximum number of data fields.
Note that the seed tree is similar to the center tree but without the O (k2) pair-wise tree matching to choose it. The reason for choosing this seed tree is clear as it is more likely for this tree to have a good alignment with data fields in other data records. Then for each Ti (i ≠ s), the algorithm tries to find for each node in Ti a matching node in Ts. When a match is found for node ni, a link is created from ni to ns to indicate its match in the seed tree. If no match can be found for node ni, then the algorithm attempts to expand the seed tree by inserting ni into Ts. The expanded seed tree Ts is then used in subsequent matching. Note that data items in the tag tree nodes are not used during matching or alignment.



```
Algorithm PartialTreeAlignment($S$)
1.  Sort trees in $S$ in descending order according to the number of
    data items that are not aligned;
2.  $T_s$ = the first tree (which is the largest) and delete it from $S$;
3.  $flag$ = false; $R = \varnothing$; $I$ = false;
4.  while ($S \neq \varnothing$)
5.      $T_i$ = select and delete next tree from $S$;
6.      Simple_Tree_Matching($T_s$, $T_i$);
7.      $L$ = alignTrees($T_s$, $T_i$);    // based on the result from line 6
8.      if $T_i$ is not completely aligned with $T_s$ then
9.          $I$ = InsertIntoSeed($T_s$, $T_i$);
10.         if not all unaligned items in $T_i$ are inserted into $T_s$ then
11.             Insert $T_i$ into $R$;
12.         endif;
13.     endif;
14.     if ($L$ has new alignment) or ($I$ is true) then
15.         $flag$ = true
16.     endif;
17.     if $S = \varnothing$ and $flag$ = true then
18.         $S = R$;  $R = \varnothing$;
19.         $flag$ = false; $I$ = false
20.     endif;
21. endwhile;
22. Output data fields from each $T_i$ to the data table based on the
    alignment results.
```

Fig.2. The partial tree alignment Algorithm.

III.PROBLEM STATEMENT

The proposed Web-page programming-language-independent vision-based approach is highly effective for deep Web data extraction. This approach primarily utilizes the visual features on the deep Web pages to implement deep Web data extraction, including data record extraction and data item extraction This approach focuses on extracting regularly arranged data records and data items from deep Web pages.

IV. OUTLINE OF PROPOSED STRUCTURE

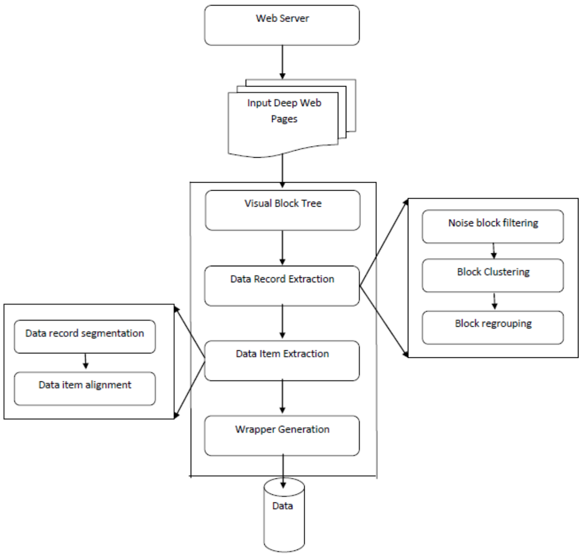


Fig.3.Outline of proposed work

V.Proposed Structure

Web pages are used to publish information for humans to browse and read, the desired information we want extracted must be visible, so the visual features of web pages can be very helpful for web information extraction. Currently, some works are proposed to process web pages based on their visual representation. For example, a web page segmentation algorithm VIPs is proposed, which simulates how a user understands web layout structure based on his/her visual perception. Our approach is implemented based on VIPs.

a) VISUAL BLOCK TREE CONSTRUCTION

The Vision-based Page Segmentation (VIPs) algorithm aims to extract the content structure of a web page based on its visual presentation. Such content structure is a tree structure, and each node in the tree corresponds to a rectangular region on a web page. The leaf blocks are the blocks that cannot be segmented further, and they represent the minimum semantic units, such as continuous texts or images. We call this tree structure Visual Block tree in this paper. In our implementation we adopt the VIPS algorithm to build a Visual Block tree for each response page. Figure 4(a) shows the content structure of the response page shown in Figure 1 and Figure 4(b) gives its corresponding Visual Block tree.

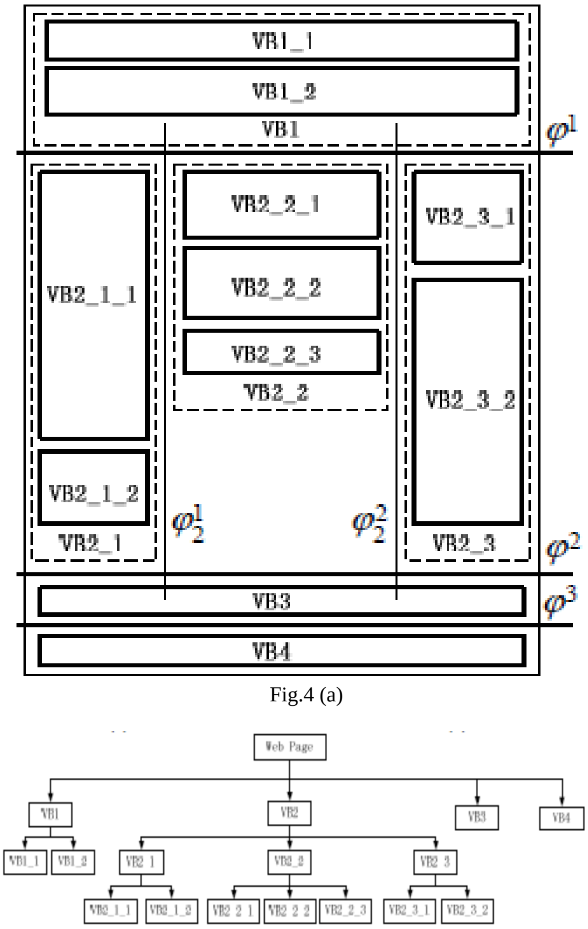


Fig.4 (a)

Fig.4 (b)

Fig.4. (a) The presentation structure and (b) its Visual Block tree

Visual features for web pages

Web pages are used to publish information on the Web.
To make the information on web pages easier to understand, web page designers often associate different types of information with distinct visual characteristics (such as font, color, layout, etc.). As a result, visual features are important for identifying special information on Web pages.
Response pages are special web pages that contain data records retrieved from Web information sources, and the data records contained in them also have some interesting distinct visual features according to our observation. Below we describe the main visual features our approach uses.

Position Features (PF): These features indicate the location of the data region on a response 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.

Though web pages are designed by different people, these designers all have the common consideration in placing the data region: the data records are the contents in focus on response pages, and they are always centered and conspicuous on web pages to catch the user's attention. By investigating a large number of response pages, we found two interesting facts.
First, data regions are always located in the middle section horizontally on response pages.
Second, the size of a data region is usually large when there are enough data records in the data region.

Layout Features (LF): 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.

The designers of web pages always arrange the data records in some format in order to make them visually regular. The regularity can be presented by one of the two layout models. In Model 1, The data records are arrayed 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 Model2, 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.

Appearance Features (AF): These features capture the visual features within data records.

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

AF2: Data contents of the same type in different data records have similar presentations in three aspects: size of image, font of plain text and font of link text.
Data records usually contain three types of data contents, i.e., images, plain texts (the texts without hyperlinks) and link texts (the texts with hyperlinks.

Content Feature (CF): These features hint the regularity of the contents in data records.
CF1: All data records have mandatory contents and some may have optional contents.
CF2: The presentation of contents in a data record follows a fixed order.

The data records are the entities in real world, and they consist of data units with different semantic concepts. The data units can be classified into two kinds: mandatory and optional. Mandatory units are those that must appear in each data record. For example, if every book data record must have a title, then titles are mandatory data units.

b) WEB DATA RECORD EXTRACTION

Based on the visual features, we propose a vision-based approach to extract data records from response pages. Our approach consists of three main steps. First, use the VIPs [9] algorithm to construct the Visual Block tree for each response page. Second, locate the data region in the Visual Block tree based on the PF features. Third, extract the data records from the data region based on the LF and AF features.

Data region discovery

PF1 and PF2 indicate that the data records are the primary content on the response pages and the data region is centrally located on these pages. The data region corresponds to a block in the Visual Block tree (in this paper we only consider response pages that have only a single data region). We locate the data region by finding the block that satisfies the two PF features. Each feature can be considered a rule or a requirement. The first rule can be applied directly, while the second rule can be represented by $(area_b/area_{responsepage}) \geq T_{dataregion}$, where $area_b$ is the area of block b, $area_{responsepage}$ is the area of the response page, and $T_{dataregion}$ is the threshold used to judge whether b is sufficiently large relative to $area_{responsepage}$. The threshold is trained from sample response pages collected from different real web sites. For the blocks that satisfy both rules, we select the block at the lowest level in the Visual Block tree.

Data records extraction from data region

In order to extract data records from the data region accurately, two facts must be considered. First, there may be blocks that do not belong to any data record, such as the statistical information and annotation about data records. These blocks are called noise blocks here.

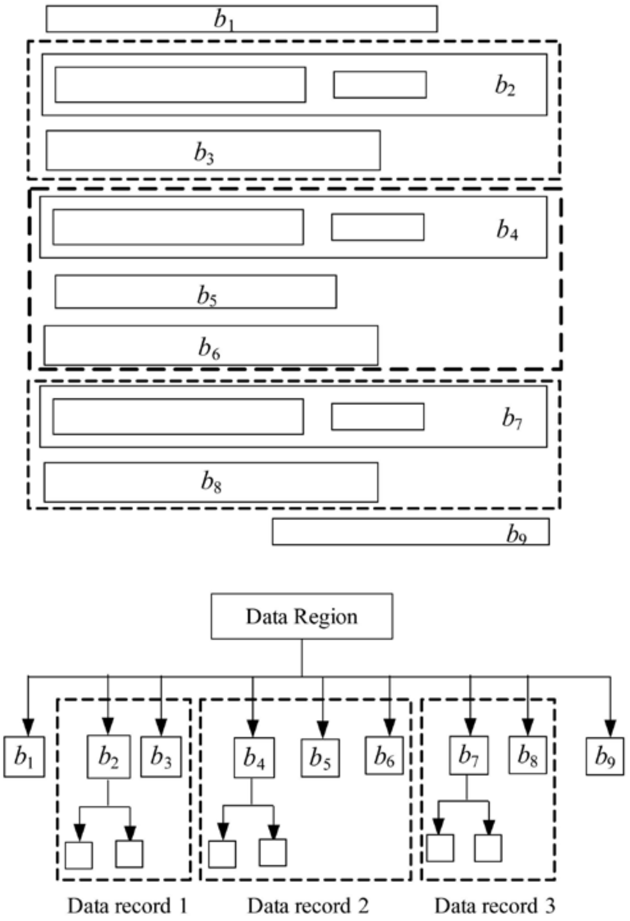


Fig.5. A general case of data region.

According to LF2, noise blocks cannot appear between data records and they can only appear at the top or the bottom of the data region. Second, one data record may correspond to one or more blocks in the Visual Block tree, and the total number of blocks one data record contains is not fixed. Figure 5 shows an example of a data region that has the above problems: Block B1 (statistical information) and B9 (annotation) are noise blocks; there are three data records (B2 and B3 form data record 1; B4, B5 and B6 form data record 2; B7 and B8 form data record 3), and the dashed boxes are the boundaries of data records. This step is to discover the boundary of data records based on the LF and AF features.

The VIPS Algorithm

In the VIPS algorithm, the vision-based content structure of a page is deduced by combining the DOM structure and the visual cues.

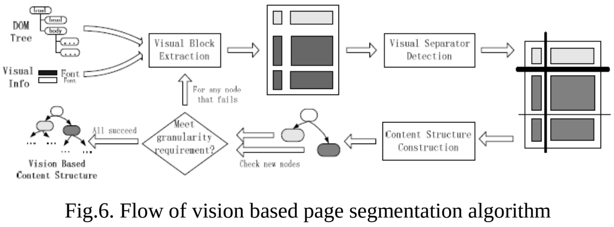

DOM Tree
Visual Info
Font
Visual Block Extraction
For any node that fails
Visual Separator Detection
Content Structure Construction
Check new nodes
Meet granularity requirement?
All succeed
Vision Based Content Structure

Fig.6. Flow of vision based page segmentation algorithm

.

The segmentation process is illustrated in Fig. 6. First, DOM structure and visual information, such as position, background color, font size, font weight, etc., are obtained from a web browser. Then, from the root node, the visual block extraction process is started to extract visual blocks of the current level from the DOM tree based on visual cues Every DOM node is checked to judge whether it forms a single block or not. If not, its children will be processed in the same way. When all blocks of the current level are extracted, they are put into a pool. Visual separators among these blocks are identified and the weight of a separator is set based on properties of its neighboring blocks. After constructing the layout hierarchy of the current level, each newly produced visual blocks is checked to see whether or not it meets the granularity requirement. If no, this block will be further partitioned. After all blocks are processed, the final vision-based content structure for the web page is outputted.

c) WEB DATA ITEM EXTRACTION

The extraction of data item process focuses on the leaf nodes of the visual block tree. The three types of data items in the data record are: mandatory, optional and static data items. The mandatory data items are always appear in all data records. The optional data items may be missed in some data records. The static data items are the annotations to data. Fixed static texts refer the text appear in every data record. The position of data items in respective to their data record is classified as: absolute position and relative position. The absolute position says that the positions of the data item of certain semantics are fixed in the line they belonged. The relative position says that the position of the data item relative to the data record ahead of it.

The extraction of data item process is carried out in two phases:

Segmentation of data record:--

The data record segmentation is carried out by collecting the leaf nodes in the data record of the visual block tree in left to right order. Leaf node also correspond each composite data item

Aligning data item:--

Data item aligning focuses on how the data items of same semantic together are aligned and it should maintain the order of data items in the data record.

d) WRAPPER GENERATION

The visual wrappers are the set of extraction rules that are generated by using the extracted data record and the data item. These are programs which performs the data record and data item extraction with the set of parameters obtained from the sample web pages. The visual information is used to generate the visual wrappers.

VI.EXPERIMENT

For experiment, the data set is collected from completeplanet.com site, which is currently the largest deep web data repository with more than 70,000 entries of web databases. These web databases are classified in 42 categories covering most domains in the real world. The experiment of web data extraction is carried out on General Data Set (GDS) which is collected from completeplanet.com. This web database is classified into 42 categories and contains more than 70,000 entries of deep web pages. From the available deep web pages, here more than 3,400 deep web pages are processed and the visual features of these all pages are determined and shown in tabular form. The average precision and recall for all processed deep web pages are 92.62 and revision is 10.70. For each page from the web database, more than 85% data is properly extracted.

	precision	*recall*	*revision*
ViDRE	$\frac{DR_c}{DR_e}$	$\frac{DR_c}{DR_r}$	$\frac{WDB_t - WDB_c}{WDB_t}$
ViDIE	$\frac{DI_c}{DI_e}$	$\frac{DI_c}{DI_r}$	

Table 6.1 performance measure used in the Evaluation of VIDE

VII.PERFORMANCE MEASURE

Three measures, precision, recall and revision, are widely used to measure the performance of data record extraction. Precision is the percentage of correctly extracted records among all extracted records. Recall is the percentage of correctly extracted records among all records that exist on response pages and revision is the percentage of the Web databases whose data records or data items are not perfectly extracted[10].

In table 6.1, DRc is the total number of correctly extracted data records, DRr is the total number of data records, DRe is the total number of data records extracted, DIc is the total number of correctly extracted data items, DIr is the total number of data items, and DIe is the total number of data items extracted; WDBc is the total number of Web databases whose precision and recall are both 100 percent and WDBt is the total number of Web databases processed. With the proposed system, more than 3000 web pages are processed from dataset which is collected from completeplanet.com. The PRECISION, RECALL and REVISION for each page is calculated separately, and finally same experiment is carried out on more than 3000 web pages. Finally the average is considered as the performance of proposed system. ViDE technique shows great result as compare to MDR and DEPTA. The overall percentage of PRECISION, REVISION and RECALL for ViDE is great as compare to MDR and DEPTA.

Technique	**data set**	**Precision**	**Recall**	**Revision**
MDR	GDS	85.3%	53.2%	55.2%
ViDRE		92.61%	92.61%	10.70%

Table 7.1 Comparison Results between MDR, and ViDRE

Technique	**data set**	**Precision**	**Recall**	**Revision**
DEPTA	GDS	75.3%	71.6%	32.8%
ViDIE		92.60%	92.61%	10.70%

Table 7.2 Comparison Results between DEPTA, and ViDIE

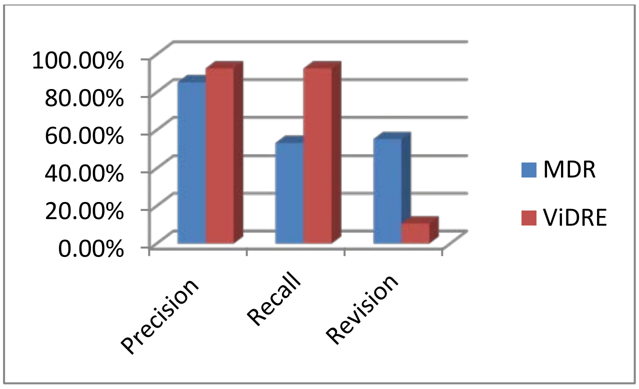


Fig 7.1 Comparison Results between MDR and ViDRE

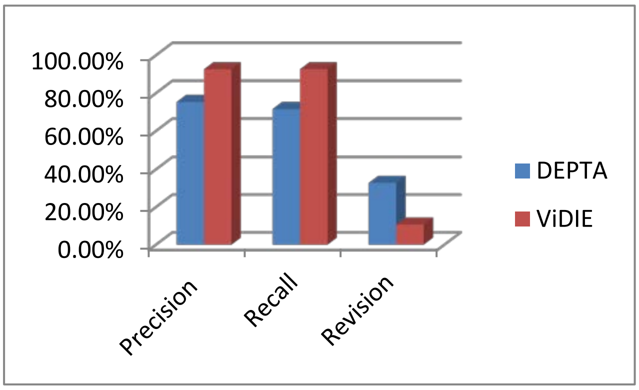


Fig 7.2 Comparison Results between DEPTA and ViDIE

VIII.CONCLUSION

In this paper, we presented a fully automated technique to extract search result data records from response pages dynamically generated by search engines. Our technique utilizes only the visual content features on the response page, which is html language or any other language independent. This differentiates our technique from other competing techniques for similar applications.

IX.FUTURE WORK

This dissertation has proven that use of ViDE system improves accuracy of web data extraction process. Although the results achieved in this dissertation were good, there are several directions to be explored in future research, which are listed below:

- ViDE can only process deep Web pages containing one data region, while there is significant number of multi data region deep Web pages.
- The efficiency of ViDE can be improved. In the current ViDE, the visual information of Web pages is obtained by calling the programming APIs of IE, which is a time-consuming process.

REFERENCES

[1]. Embley, D., Jiang, Y and Ng, Y. "Record-boundary discovery in Web documents." SIGMOD-99, 1999

[2]. Buttler, D., Liu, L., Pu, C. A fully automated extraction system for the World Wide Web. IEEE ICDCS-21, 2001. [3]. Liu, B., Grossman, R. and Zhai, Y. "Mining data records from Web pages." KDD-03, 2003.

[4]. Chakrabarti, S., Integrating the Document Object Model with hyperlinks for enhanced topic distillation and information extraction, In the 10th International World Wide Web Conference, 2001

[5]. Lerman, K., Getoor L., Minton, S. and Knoblock, C. "Using the Structure of Web Sites for Automatic Segmentation of Tables." SIGMOD-04, 2004

[6]. Chakrabarti, S., Punera, K., and Subramanyam, M., Accelerated focused crawling through online relevance feedback, In Proceedings of the eleventh international conference on World Wide Web (WWW2002), 2002, pp. 148-159.

[7]. D. Cai, S. Yu, J. Wen, W. Ma. Extracting Content Structure for Web Pages Based on Visual Representation. In APWeb, pages 406-417, 2003.

[8]. Buneman, P., Davidson, S., Fernandez, M., and Suciu, D., Adding Structure to Unstructured Data, In Proceedings of the 6th International Conference on Database Theory (ICDT'97), 1997, pp. 336-350.

[9]. D.Cai, X. He, J.-R. Wen and W.-Y. Ma, *"Block-Level Link Analysis,"* Proc. SIGIR, pp. 440-447, 2004.

[10]. *"Extraction from the Web,"* Information Systems, vol. 23, no. 8, pp. 521-538, 1998.

[11]. A. Laender, B. Ribeiro-Neto, A. da Silva, and J. Teixeira, *"A Brief Survey of Web Data Extraction Tools,"* SIGMOD Record, vol. 31, no. 2, pp. 84-93, 2002.

[12]. B. Liu, R.L. Grossman, and Y. Zhai, *"Mining Data Records in Web Pages,"* Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 601-606, 2003.