An Efficient Mechanism for Product Data Extraction from E-Commerce Websites

Malik Javed Akhtar[1], Zahur Ahmad[1], Rashid Amin[1, *], Sultan H. Almotiri[2], Mohammed A. Al Ghamdi[2], Hamza Aldabbas[3]

Abstract: A large amount of data is present on the web which can be used for useful purposes like a product recommendation, price comparison and demand forecasting for a particular product. Websites are designed for human understanding and not for machines. Therefore, to make data machine-readable, it requires techniques to grab data from web pages. Researchers have addressed the problem using two approaches, i.e., knowledge engineering and machine learning. State of the art knowledge engineering approaches use the structure of documents, visual cues, clustering of attributes of data records and text processing techniques to identify data records on a web page. Machine learning approaches use annotated pages to learn rules. These rules are used to extract data from unseen web pages. The structure of web documents is continuously evolving. Therefore, new techniques are needed to handle the emerging requirements of web data extraction. In this paper, we have presented a novel, simple and efficient technique to extract data from web pages using visual styles and structure of documents. The proposed technique detects Rich Data Region (RDR) using query and correlative words of the query. RDR is then divided into data records using style similarity. Noisy elements are removed using a Common Tag Sequence (CTS) and formatting entropy. The system is implemented using JAVA and runs on the dataset of real-world working websites. The effectiveness of results is evaluated using precision, recall, and F-measure and compared with five existing systems. A comparison of the proposed technique to existing systems has shown encouraging results.

Keywords: Document object model, rich data region, common tag sequence, web data extraction, deep web mining.

1 Introduction

The web is a global resource, accessible widely all over the world. It helps in getting information about products and services without bothering to visit the market and malls. This creates an opportunity for people to compare products to analyze the market, to

[1] University of Engineering and Technology, Taxila, Pakistan.

[2] Computer Science Department, Umm Al-Qura University, Makkah City, Saudi Arabia.

[3] Al-Balqa Applied University, Al-Salt, Jordan.

[*] Corresponding Author: Rashid Amin. Email: rashid.sdn1@gmail.com.

Received: 11 May 2020; Accepted: 25 July 2020.

make an optimal selection of products, competitor analysis and demand forecasting for a product. It also creates a chance for government organizations to monitor the web and find out the details of businesses and currently offered products to verify their legitimacy. They can observe which organizations are selling authorized or unauthorized products.

The web is a large data repository and its manual monitoring and navigation to compare products is a tedious task and requires a huge amount of human resources. Therefore, the need arises for an automatic system and technique to extract data from web documents to do the said tasks efficiently and accurately. There is an emergence of web-based businesses, researchers are working on data extraction problem and a variety of solutions have been proposed which address the problem from different aspects. The system presented by RoadRunner [Crescenzi, Mecca and Merialdo (2001)] is an example of early web data extraction systems and DCADE [Yuliana and Chang (2020)] is an example of current work. A major issue in this regard is data extraction from the web. Because databases of vendor organizations are not accessible publicly due to security, safety and privacy reasons therefore we have to develop techniques to extract data from web pages on client's requests. Web pages are generally semi-structured or poorly structured because they are served for humans and not for machines. Therefore, research-based methods are needed to accomplish the extraction task. Following are some important applications of web data extraction:

1. A system is needed to extract data from the semi-structured web that convert it into some structured form and compare their features because it is not feasible to visit all the sites one by one, navigate products and compare prices and features of required products, as data extraction from such sites is not a trivial task.
2. Due to large and disperse data objects on the web it is not feasible for individuals to get data for their needs like statistical operations and price comparison to discover different trends because internet connectivity may not be available all the time for some people. Data extraction helps in this case and provides an easy solution, one needs to schedule the extraction task based on web availability and the system efficiently do the task.
3. Data extraction techniques serve the purpose of extraction of data objects from web pages and make it possible to compare with products registered in the databases of legal cells of different departments. Zheng et al. [Zheng, Gu and Li (2012)] have proposed a system to detect illegal products from advertising web sites in China.
4. Market intelligence is defined in Kotler [Kotler (2009)] as "information relevant to a company's markets, gathered and analyzed specifically for the purpose of accurate and confident decision-making in determining market opportunity, market penetration strategy, and market development metrics". For surveying existing products of competitors before launching companies can use automatic data extraction tools to scrap web and build their repository to process further for the decision process.
5. Customer feedback about products is very useful for venders to improve the features of their products. Data extraction techniques serve the process of acquiring freely available data from the semi-structured web. Kobayashi et al. [Kobayashi, Inui and Matsumoto (2007); Chang and Lui (2001)] presented systems for mining of customer opinions about products in the web-based environment.

1.1 Challenges in Web data extraction

Web data extraction from all sources of interest is a non-trivial task. Major challenges faced by researchers are that the web is a large resource and the solutions with a high degree of automation are required. To make useful decisions based on extracted data from the web, computationally efficient systems are needed. Data extraction techniques require guaranteeing the privacy of users of web platforms, for example, social media sites. Learning-based data extraction systems require some labeled page on the basis of which they generate rules to extract objects from pages. The web is continuously evolving, and websites are going more and more complex. Changes in the structure of a page can make a wrapper useless if it is not compatible with current structural design. Therefore, researchers have to keep in mind the changes in the structures of web resources. Kaiser et al. [Kaiser and Miksch (2005)] categorized data extraction techniques into two groups, learning techniques, and knowledge engineering techniques. The former technique requires annotated examples of data by domain experts whereas later uses heuristics observations, visual cues and structural similarities to identify and extract data records. Data extraction techniques can also be classified based on the degree of automation i.e., automatic and manual techniques. Manual techniques involve a human effort to extract data from web pages whereas automatic techniques do not need human intervention. The systems PROTEUS [Yangarber and Grishman (1998)], FASTUS [Hobbs, Appelt, Tyson et al. (1992)], GE NLTOOLSET [Krupka, Jacobs, Rau et al. (1992)], and PLUM [Fast] are examples of manual systems. Automatic techniques can be further classified as supervised and unsupervised. Supervised systems require human intervention during execution. For every new type of web page training data is required to generate rules. Therefore, scalability is still an issue because of human intervention. GATE [Cunningham, Maynard, Bontcheva et al. (2002)], RAPIER [Mooney (1999)], WHISK [Soderland (1999)], CRYSTAL [Soderland, Fisher, Aseltine et al. (1995)], LIEP [Huffman (1996)], PALKA [Kim and Moldovan (1995)] and AutoSlog [Riloff (1993)] are examples of supervised learning systems. Unsupervised learning systems require only a small number of annotated examples and expand the training data automatically. Therefore, they need very small human effort and hence are scalable systems. Shopbot [Doorenbos, Etzioni and Weld (1997)], RoadRunner [Crescenzi, Mecca and Merialdo (2001)] and IEPAD [Chang and Lui (2001)] are examples of unsupervised learning systems. Alvarez et al. [Álvarez, Pan, Raposo et al. (2008)] and Sleiman et al. [Kushmerick, Weld and Doorenbos (1997)] are examples of knowledge engineering approaches. Alvarez et al. [Álvarez, Pan, Raposo et al. (2008)] treats web page as a tree structure and HTML elements as nodes of the tree, it creates a model for data types of text that appeared on the web page and maps that model to page. Utilizing Document Object Model, it identifies a list of data records from the page and by editing distance measure and bottom-up clustering, it divides the identified region into data records. TEX [Kushmerick, Weld and Doorenbos (1997)] takes two pages of the same website generated by the same template but with different data values. It treats the pages as strings of text and compares them to eliminate the common portions of pages. The remaining portions in both pages include the data objects. It actually uses the property that two script generated pages served by the same website share the structure but not data elements on a page, e.g., menus, advertisements and layout of the page are the same for every served page but data objects are different for each page.

1.2 Tasks in Web data extraction

Web data extraction includes:

1. Deep Web Crawling
2. Data Extraction and
3. Mining of data after extraction

   i.) Deep Web Crawling: Automation of deep web crawling is a research area and people have addressed it in different ways. Wang et al. [Wang, Lu and Chen (2009); Wang, Lu, Liang et al. (2012)] proposed an algorithm to select a query from a sample database source and a mechanism to select queries to crawl the web efficiently. Alvarez et al. [Álvarez, Pan, Raposo et al. (2008)] presented a technique to crawl the web by executing the JavaScript present on client-side pages. Rivero et al. [Rivero, Frantz, Ruiz et al. (2011)] presented a framework to use high-level structured queries for deep web data integration. Prieto et al. [Prieto, Alvarez, López-García et al. (2012)] presented a scale to classify the web crawling systems.

   ii.) Data Extraction: After web crawling next task is data extraction from crawled pages. Our work falls in this area. People have addressed this problem in different ways, some used text processing method, e.g., in TEX [Sleiman and Corchuelo (2013)], IEPAD [Chang and Lui (2001)], OLERA [Chang and Kuo (2004)] and STALKER [Muslea, Minton and Knoblock (1999)] are an example of text processing and tokenizing of pages. DOM mining/processing is another way to address this problem. We have used DOM mining in our technique.

   iii.) Mining of data after Extraction: As the name suggests it is purely a data mining and a text mining area. When data is extracted it is in raw form and further operations are required to convert it into useful form, e.g., label assignment and numerical data separation. Cleansing is also needed because a lot of noise also occurs in data objects e.g., text elements with an offer to buy a product.

2 Related work

In this section, we have provided a survey of existing work in the field of web data extraction. Kaiser et al. [Kaiser and Miksch (2005)] categorized data extraction techniques into two groups, learning techniques, and knowledge engineering techniques. Learning techniques require annotated examples of data by domain experts and knowledge engineering techniques use heuristics, observations, visual cues and structural similarities to identify and extract data records.

2.1 Learning techniques

Muslea et al. [Muslea, Minton and Knoblock (1999)] presented STALKER a semi-supervised data extraction system based on Finite Automaton. It accepts many marked pages from the user for training and tries to generate rules to extract data e.g., data of persons is stored in an HTML table. The attributes of a person are name, education, and address. STALKER requires training pages marked by the user. It generates rules and uses these rules to extract data from a semi-structured data source.

Crescenzi et al. [Crescenzi, Mecca and Merialdo (2001)] presented RoadRunner, an automatic wrapper generation system based on training samples. In the beginning, it takes two pages considering one as wrapper and other as the sample. Then it compares the sample with the wrapper and tries to find mismatches. Mismatches are of three types (a) Text mismatches (b) Tag optional mismatches and (c) Tag iterators mismatches. RoadRunner extracts data well until the pages follow a pattern and when they violate the pattern e.g., instead of <table> some page is using <ul> tag for data representation it will not work properly.

Wang et al. [Wang and Lochovsky (2003)] presented Dela (Data Extraction and Label Assignment) a data extraction system which can extract data from a website which provides the user a facility to search from its database using a number of labeled text boxes offered by the same website and also guesses the attributes of tables of the database from which the data is being retrieved using labels on text boxes. Dela consists of the following components:

1. A number of domain-specific keywords are stored in a database. The labels of text boxes are compared with these keywords and if a label and keyword match then it is assigned to the text of that text box and this process is done until all keywords are compared with all text boxes. The preceding step outputs a number of combinations of keywords to send the database of the website. The system sends these combinations as queries one by one and stores the pages served by the web site in response to queries.
2. The extracted pages from the website based on aforesaid combinations of keywords are taken as input in a module to obtain wrapper for the extraction of data records from the said pages. The wrappers are induced based on Wang et al. [Wang and Lochovsky (2002)]. This process is based on the assumption that when more than one data tuples are there on a page then they form a regular repeated pattern.
3. The fetched pages and generated wrappers are passed to data aligner which generates a Finite Automaton (FA) according to wrappers and checks the pages on this FA. The data aligner takes a page as a group of tokens and parses this group using FA. The accepted sequence is extracted from the page and stored in a data table. After saving data in a data table the composite attributes are broken down to simpler ones using visual cues.

2.2 Knowledge engineering techniques

Chang et al. [Chang and Lui (2001)] presented IEPAD gets one page, searches repetitive HTML tag patterns and displays it to the user. The user selects the patterns of interest for data extraction from the page. It is an elementary technique for information extraction because it generates some patterns which become useless when the pattern on a website changes and the user has to choose some patterns manually which has an implicit weakness that generally every website has its own format to display data and user has to run the system for every website, i.e., the sites of interest should be known in advance.

OLERA presented by Chang et al. [Chang and Kuo (2004)] is a semi-supervised technique to extract data from semi-structured data sources like websites. OLERA provides an interface to select the block i.e., data area from a page of interest and after selection by the user it finds out a similar pattern as compared to selected block and outputs the data in a tabular format. User has to browse the web and the proposed system can work only for websites following a pattern therefore the system requires user intervention for every document and is not scalable for a web automated system.

Alvarez et al. [Álvarez, Pan, Raposo et al. (2008)] proposed a technique to extract a list of data records from a given page. They observed that (1) Each data record consists of a set of consecutive sub-trees in the DOM tree structure. (2) The attributes of data records occur in the same order and have the same path from the root in DOM. (3) Data records occur in a list on the result page and all data records are consecutive and make a data region. To identify data region all elements of DOM with keywords are listed and the score of a DOM object is increased by one which is the first common ancestor of a pair of keyword elements. Data region is further divided into data records by utilizing the above-mentioned properties (1) and (2). They generated candidate lists of data records and select an appropriate list using an auto similarity measure. The candidate lists are n in number and calculation of edit distance of their elements and applying string clustering techniques on all HTML elements of a web page, make this technique a CPU intensive task. The technique gives false positives when the number of keywords in data records is less than other elements e.g., menu items because they are using auto similarity of elements but differentiation of menu items from data records are not considered.

Grigalas [Grigalis (2013)] presented ClustVX a clustering-based technique that also utilizes visual features to identify the data region on a page and then divides the extracted data region into data records. ClustVX is an automatic and unsupervised data extraction technique and works well when there is a list of data records on a web page but it will provide false positives when data records are small in number with fewer attributes and web page has a comparatively large menu bar with a larger number of options.

Zheng et al. [Zheng, Gu and Li (2012)] proposed a domain-specific technique DE-SSE to extract product descriptions from web pages. Their main intention is to search detail of illegal medical products being sold online and to provide these details to law enforcement agencies. They developed a technique to discover the boundaries of an HTML element that contains a complete description of the product. Their techniques work in situations when attributes of a product are labeled by appropriate words, e.g., price, name, manufacturer, size, ingredients, and availability. But it will fail to identify any user data if there is no annotation because they are using labels as roles and a metric Structural Semantic Entropy of these roles is calculated.

Sleiman et al. [Sleiman and Corchuelo (2013)] presented TEX. It searches for a shared sequence of tokens between two or more documents with a maximum number of tokens. It uses text-matching which is a CPU intensive task. Another issue with this approach is the requirement of at least two documents generated by the same template because it is based on the search of shared patterns of tokens between two or more documents.

Liu et al. [Liu, Meng and Meng (2009)] proposed ViDE. It utilizes visual information to divide a web page into visual blocks, i.e., rectangles. It uses location, size, and font of HTML elements to divide the page. ViDE is an automatic and unsupervised approach that works well when data records on a web page are presented according to assumptions and observations. But in certain situations, e.g., when data region is less than a menu bar or other regions on a page it will give false positive. In some cases, styles may be missed, or images do not load completely then visual block formation can go wrong and the system may output undesirable results.

Mundluru et al. [Mundluru, Raghavan and Wu (2010)] proposed PIE an automatic data extraction technique based on Document Object Model tree mining. This technique may output menu items as data records if the data region contains a very small number of data records with a smaller number of attributes. It will also ignore a portion of data records if data records are presented in multiple noncontiguous portions.

Hiremath et al. [Hiremath and Algur (2009)] presented a vision-based automatic technique to extract data from web pages. It is a combination of two techniques VSAP (Visual Structure-based Analysis of web pages) and EDIP (Extraction of Data Items from web Pages). Hiremath et al. [Hiremath and Algur (2009)] defined the boundary rectangle of HTML elements which is an area covered by an element when it is rendered on screen. Height and widths are covered by an element when it is rendered on a display device. To get the largest bounded rectangle areas of bounded rectangles of direct children of <body> element is calculated and the rectangle with the greater area is taken as a bounded rectangle. This technique has two issues (i) It is taking direct children of BODY tag and searching the largest rectangle from these elements as a bounded rectangle which may give the wrong data region if a number of data records are small and the page has a comparatively large menu. (ii) It is assuming that data records are presented in an HTML table element but in contemporary web pages, it may not be the case.

Gengxin et al. [Miao, Tatemura, Hsiung et al. (2009)] proposed an automatic clustering-based technique to extract data records from semi-structured web pages. Weng et al. [Weng, Hong and Bell (2011)] proposed a web data extraction technique utilizing visual features involved in the presentation of data in web data documents. It is an automatic technique that requires keywords and some sample pages to learn the ration of the area covered by data region to the area of the whole web page.

In light of the above discussion, data extraction techniques can be divided into two broad categories, i.e., machine learning-based techniques and knowledge engineering techniques. Machine learning techniques require sample pages annotated by the user. They generate regular expressions and finite state automata which are used to match similarly structured data on other web pages. This approach is a CPU intensive because regular expressions matching is pattern matching which is $O(n^2)$ and multiple patterns are matching for single data records make its time complexity even higher. Therefore, we have decided to use the knowledge engineering technique to solve the data extraction problem. We developed a Rich Data Region detection technique with less time complexity, i.e., O(n) than the existing technique proposed by Alvarez et al. [Álvarez, Pan, Raposo et al. (2008)]. We have also used a new idea of formatting entropy to differentiate data records from noisy objects. Experiments show that the proposed technique works better than existing systems.

3 Proposed system

3.1 Problem definitions

An automatic and unsupervised technique is required to extract data of products from web pages of e-commerce websites. The technique is required to be able to extract data with high accuracy and efficiency utilizing minimum hardware resources.

3.2 Definitions of propose system

This section explains the fundamentals of the proposed system i.e., definitions and observations about the structure of web pages corresponding to a list of data records.

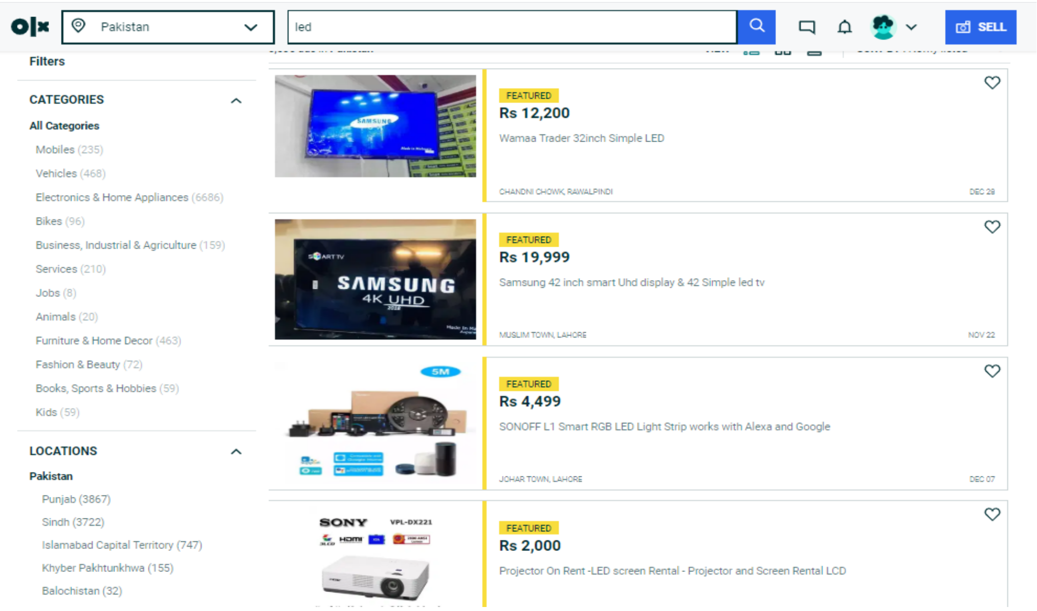
- Text Elements: A text element is a sequence of characters that is human understandable and describes some property of a physical object**.**
- Data Record: A data record is a set of text elements that are logically related and represent a unique entity.
- Depth of an element: Depth of an element in DOM tree taking the depth of body element as zero.
- Frequent Depth: Value of the depth pertaining to the highest number of keywords containing elements.
- Deep Elements: Elements with frequent depth are deep elements.
- Rich Data Region: An element in the DOM tree that contains a list of data records. Most of the keywords occurred in descendant elements of this element.
- Tag String: In-order concatenation of tag names of text elements of a data record is known as Tag String.
- Common Tag Sequence: Common sequence in tags' strings of two data records is known as the common tag sequence.

3.3 Observations

We have made the following observations during the inspection of web pages about products.

OB1*:* On the web, data about entities like products, services, jobs, books, etc., are presented as data records.

OB2*:* Formatting used in a data record is diverse, i.e., each text element is formatted differently with respect to its closest neighbor text elements.



CATEGORIES

All Categories

Mobiles (235)
Vehicles (468)
Electronics & Home Appliances (6686)
Bikes (96)
Business, Industrial & Agriculture (159)
Services (210)
Jobs (8)
Animals (20)
Furniture & Home Decor (463)
Fashion & Beauty (72)
Books, Sports & Hobbies (59)
Kids (59)

LOCATIONS

Pakistan

Punjab (3867)
Sindh (3722)
Islamabad Capital Territory (747)
Khyber Pakhtunkhwa (155)
Balochistan (32)

FEATURED
Rs 12,200
Wamaa Trader 32inch Simple LED
CHANDNI CHOWK, RAWALPINDI DEC 28

FEATURED
Rs 19,999
Samsung 42 inch smart Uhd display & 42 Simple led tv
MUSLIM TOWN, LAHORE NOV 22

FEATURED
Rs 4,499
SONOFF L1 Smart RGB LED Light Strip works with Alexa and Google
JOHAR TOWN, LAHORE DEC 07

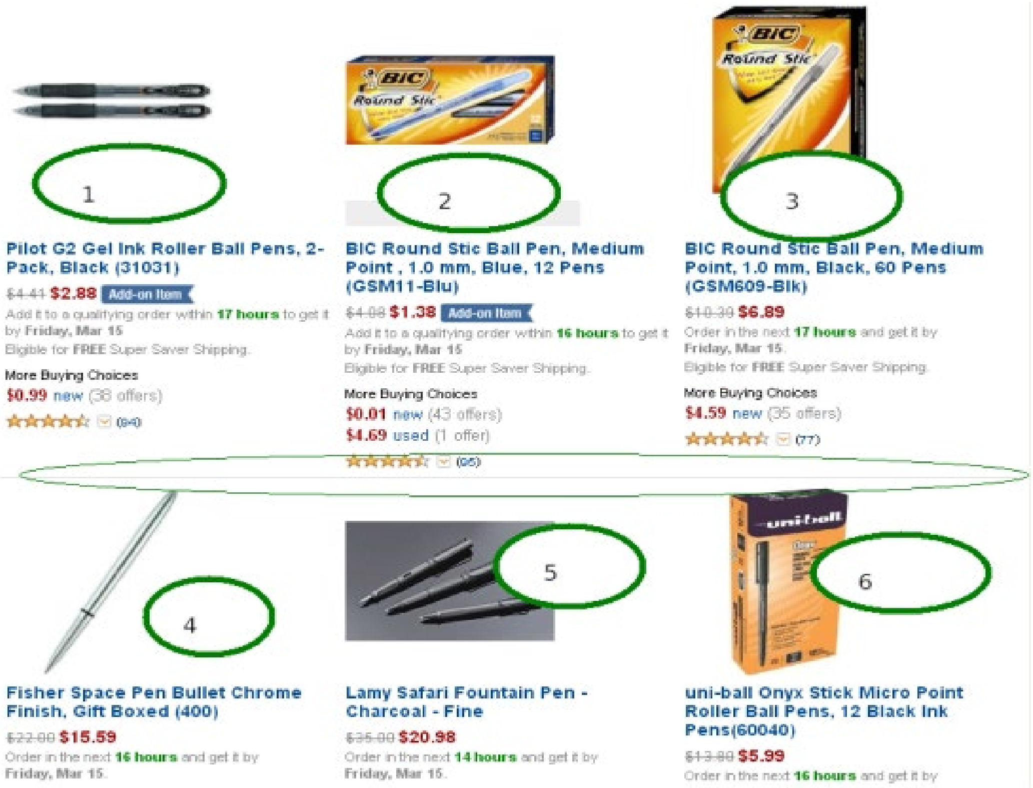
FEATURED
Rs 2,000
Projector On Rent -LED screen Rental - Projector and Screen Rental LCD

Figure 1: A portion of a page generated in result of a query

OB3: In response to a query, issued by the user, using the interface provided by a website, a list of data records is returned. As shown in Fig. 1 a query has been issued using the search interface provided by the web site (olx.com) and a list of items present in the database has been returned by the website. It also shows the validity of Observation OB2 i.e., all text elements in data records are formatted differently to their closest neighbors but data records are externally homogeneous to their neighbors.

OB4: All the data records in a list are formatted symmetrically i.e., all data elements are formatted in the same way and the order of occurrence of text elements of the same type is also the same. As shown in Fig. 1 data records are composed of Headings, category-location, "Price" and "Date Posted". Each two neighboring text elements are formatted in different style but with respect to other data records, all data records share the same formatting scheme and the same order of occurrence of elements (i.e., 1. Heading 2. Price 3. Posting date in first row respectively and 4. Category/location in the second row).

OB5: More than one data records are neighbors in a list of data records i.e., they share the first common parent node if more than one data records match the search criteria. Generally, list of data elements returned by search query to a web site is contiguous but, in some cases, the list does not contiguous and comprises of more than one sub-lists as shown in Fig. 2 This approach is used because the admin of site wants to embed some advertisements in between the list, therefore, data records are grouped in sub-lists and these sub-lists are used as a boundary to add advertisements. As shown in Fig. 2 sub-lists of three data records are formed and a horizontal role separates the said lists.



Pilot G2 Gel Ink Roller Ball Pens, 2-Pack, Black (31031)
~~$4.41~~ $2.88 Add-on Item
Add it to a qualifying order within 17 hours to get it by Friday, Mar 15
Eligible for FREE Super Saver Shipping.
More Buying Choices
$0.99 new (38 offers)
(94)

BIC Round Stic Ball Pen, Medium Point , 1.0 mm, Blue, 12 Pens (GSM11-Blu)
~~$4.08~~ $1.38 Add-on Item
Add it to a qualifying order within 16 hours to get it by Friday, Mar 15
Eligible for FREE Super Saver Shipping.
More Buying Choices
$0.01 new (43 offers)
$4.69 used (1 offer)
(95)

BIC Round Stic Ball Pen, Medium Point, 1.0 mm, Black, 60 Pens (GSM609-Blk)
~~$10.39~~ $6.89
Order in the next 17 hours and get it by Friday, Mar 15.
Eligible for FREE Super Saver Shipping.
More Buying Choices
$4.59 new (35 offers)
(77)

Fisher Space Pen Bullet Chrome Finish, Gift Boxed (400)
~~$22.00~~ $15.59
Order in the next 16 hours and get it by Friday, Mar 15.

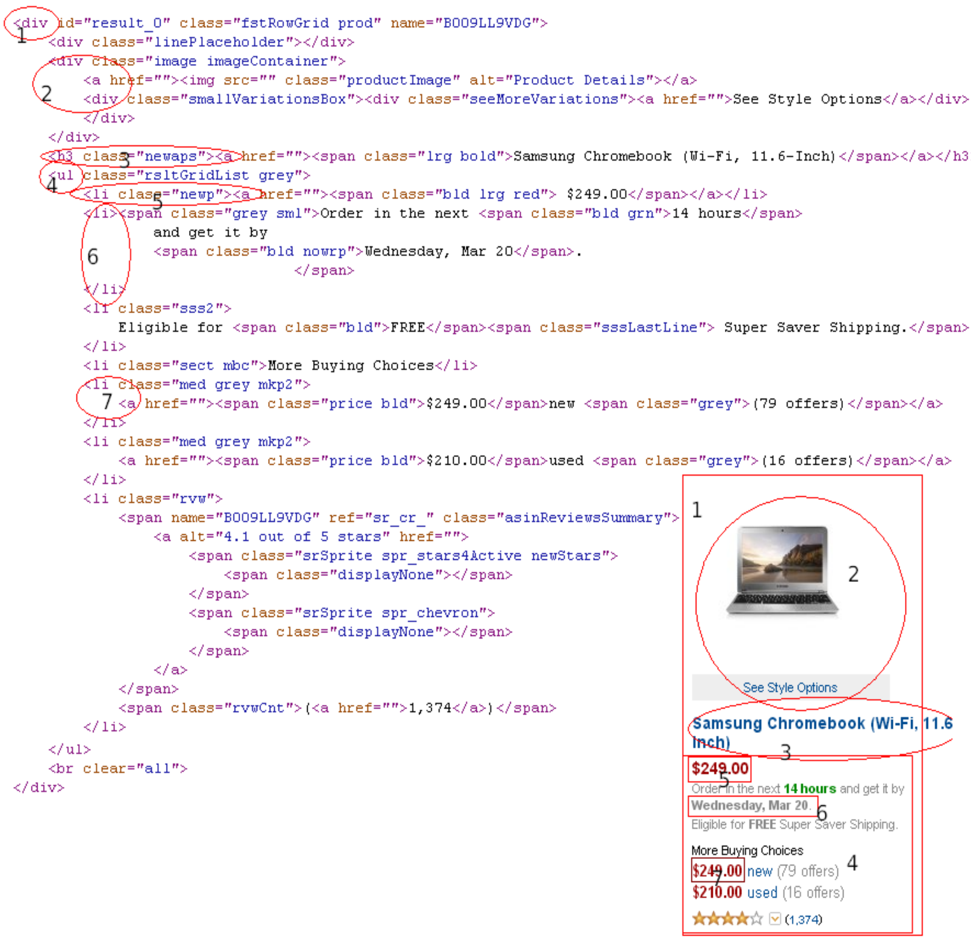
Lamy Safari Fountain Pen - Charcoal - Fine
~~$35.00~~ $20.98
Order in the next 14 hours and get it by Friday, Mar 15.

uni-ball Onyx Stick Micro Point Roller Ball Pens, 12 Black Ink Pens(60040)
~~$13.80~~ $5.99
Order in the next 16 hours and get it by

Figure 2: Sub-lists of data records

OB6: Text elements in a data record are generally grouped in sub-trees but all sub-trees constituting a data element are encapsulated in an HTML element which is shared by all data records. As shown in Fig. 3 a snap of a data element taken from a website (www.amazon.com) and its HTML source is marked for portions of the data element accordingly. The first element i.e., a div encapsulates all sub-trees and then portions 2 to 7 are shown by corresponding sub-trees of HTML tags. We found this observation true in our data set. Developers of websites use some HTML template for rendering the data records because it facilitates the symmetry of design. For the background of a data object, only one CSS (Cascading Style Sheets) class can be used, and visibility of every text element can be controlled using only one CSS file by assigning classes to each element. Therefore, developers have adopted this design strategy as a de facto standard.

OB7: All data records are grouped in a single HTML element which is their least common ancestor. To capture the attention of visitors generally, data records are rendered contiguously with the same background and some common features of the text. Therefore, all data records are embedded in an HTML template which consists of an HTML element along with visual and position properties. Since little effort is needed to customize this template at any time therefore people adopted this design strategy.



```
<div id="result_0" class="fstRowGrid prod" name="B009LL9VDG">
    <div class="linePlaceholder"></div>
    <div class="image imageContainer">
        <a href=""><img src="" class="productImage" alt="Product Details"></a>
        <div class="smallVariationsBox"><div class="seeMoreVariations"><a href="">See Style Options</a></div>
        </div>
    </div>
    <h3 class="newaps"><a href=""><span class="lrg bold">Samsung Chromebook (Wi-Fi, 11.6-Inch)</span></a></h3>
    <ul class="rsltGridList grey">
        <li class="newp"><a href=""><span class="bld lrg red"> $249.00</span></a></li>
        <li><span class="grey sml">Order in the next <span class="bld grn">14 hours</span>
                and get it by
                <span class="bld nowrp">Wednesday, Mar 20</span>.
                                </span>
        </li>
        <li class="sss2">
            Eligible for <span class="bld">FREE</span><span class="sssLastLine"> Super Saver Shipping.</span>
        </li>
        <li class="sect mbc">More Buying Choices</li>
        <li class="med grey mkp2">
            <a href=""><span class="price bld">$249.00</span>new <span class="grey">(79 offers)</span></a>
        </li>
        <li class="med grey mkp2">
            <a href=""><span class="price bld">$210.00</span>used <span class="grey">(16 offers)</span></a>
        </li>
        <li class="rvw">
            <span name="B009LL9VDG" ref="sr_cr_" class="asinReviewsSummary">
                <a alt="4.1 out of 5 stars" href="">
                    <span class="srSprite spr_stars4Active newStars">
                        <span class="displayNone"></span>
                    </span>
                    <span class="srSprite spr_chevron">
                        <span class="displayNone"></span>
                    </span>
                </a>
            </span>
            <span class="rvwCnt">(<a href="">1,374</a>)</span>
        </li>
    </ul>
    <br clear="all">
</div>
```

See Style Options

Samsung Chromebook (Wi-Fi, 11.6 Inch)

$249.00

Order in the next **14 hours** and get it by **Wednesday, Mar 20.**

Eligible for **FREE** Super Saver Shipping.

More Buying Choices

$249.00 new (79 offers)

$210.00 used (16 offers)

★★★★☆ (1,374)

Figure 3: A rendered data record and its HTML source

OB8*:* Maximum number of searched keywords and co-relative terms to the keywords, entered by the user occurs in the rich data region. Users search a product by values of its attributes e.g., name, price, venders, manufacturers, and so on. In response to queries, the underlying system returns data of products whose attributes' values match the query terms. Therefore, the occurrence of searched keywords is higher in the region of a web page containing data records than other regions of the page.

OB9: All text elements in a menu item have the same formatting, i.e., each text element, within a menu sub-list of options, is formatted in the same way. Fig. 4 shows snaps of menus and data object lists from four different websites. Portions encircled by red circles are menus and portions encircled by green circles are data records and it can be seen that text elements in menus are formatted in the same way but in data records diverse formatting is used. Designers of web pages use similar styles for menu items because they want to tell the visitor that this group of elements is of similar type and every element has the same purpose i.e., to bring the user to another page.

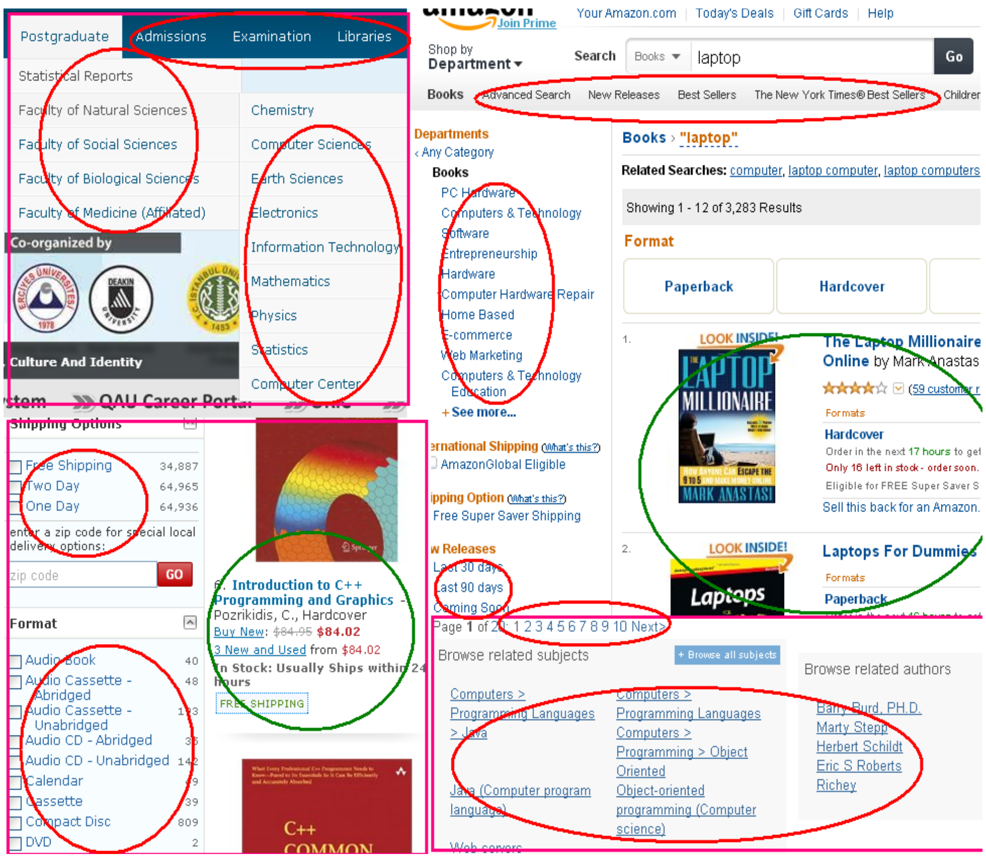

Postgraduate
Admissions
Examination
Libraries
Statistical Reports
Faculty of Natural Sciences
Faculty of Social Sciences
Faculty of Biological Sciences
Faculty of Medicine (Affiliated)
Co-organized by
Culture And Identity
QAU Career Portal
Chemistry
Computer Sciences
Earth Sciences
Electronics
Information Technology
Mathematics
Physics
Statistics
Computer Center
Join Prime
Your Amazon.com | Today's Deals | Gift Cards | Help
Shop by Department
Search
Books
laptop
Go
Books
Advanced Search
New Releases
Best Sellers
The New York Times® Best Sellers
Children
Departments
‹ Any Category
Books
PC Hardware
Computers & Technology
Software
Entrepreneurship
Hardware
Computer Hardware Repair
Home Based
E-commerce
Web Marketing
Computers & Technology Education
+ See more...
Books › "laptop"
Related Searches: computer, laptop computer, laptop computers
Showing 1 - 12 of 3,283 Results
Format
Paperback
Hardcover
LOOK INSIDE!
The Laptop Millionaire Online by Mark Anastas
(59 customer r
Formats
Hardcover
Order in the next 17 hours to get
Only 16 left in stock - order soon.
Eligible for FREE Super Saver S
Sell this back for an Amazon.
LOOK INSIDE!
Laptops For Dummies
Formats
Paperback
ernational Shipping (What's this?)
AmazonGlobal Eligible
ipping Option (What's this?)
Free Super Saver Shipping
w Releases
Last 30 days
Last 90 days
Coming Soon
Page 1 of 20 1 2 3 4 5 6 7 8 9 10 Next›
Browse related subjects
+ Browse all subjects
Computers > Programming Languages > Java
Java (Computer program language)
Computers > Programming Languages
Computers > Programming > Object Oriented
Object-oriented programming (Computer science)
Web servers
Browse related authors
Barry Burd, PH.D.
Marty Stepp
Herbert Schildt
Eric S Roberts
Richey
Shipping Options
Free Shipping 34,887
Two Day 64,965
One Day 64,936
enter a zip code for special local delivery options:
zip code
GO
Format
Audio Book 40
Audio Cassette - Abridged 48
Audio Cassette - Unabridged 173
Audio CD - Abridged 35
Audio CD - Unabridged 142
Calendar 9
Cassette 39
Compact Disc 809
DVD 2
6. Introduction to C++ Programming and Graphics -
Pozrikidis, C., Hardcover
Buy New: $84.95 $84.02
3 New and Used from $84.02
In Stock: Usually Ships within 24 hours
FREE SHIPPING
C++ COMMON

Figure 4: Data records (green circles) and menus (red circles)

OB10: The first text element in data records is mandatory. Data records are rendered symmetrically to create an impression that these elements are semantically similar to each other. Therefore, the most common attribute among all data records (e.g., title) is rendered at the start of each data record. In some cases, the serial number is the first element. This is obviously occurring in all data records as the first element we inspected all pages in data sets and a large number of other web sites and found this observation valid in all cases. As shown in Fig. 1 through Fig. 4 the first element in the data record is visually different from others and generally it is the title of the product.

3.4 Overview of system

This section provides an overview of the proposed solution and web pages that contain data of interest. Resulting pages in the response of a query, issued by a user, contain data records matching the query criteria along with menus and banners according to the template used by the web designer. The data records are useful information for us whereas menus and banners are useless. Our task is to identify the region containing data records and to extract the list of data records in such a way that noise i.e., information other than data records, is excluded and data extracted contains only useful information. For this purpose, we first identify the rich data region which will exclude most of the unnecessary elements e.g., menu bars but some elements like banners and links to the next pages in case of multiple page results which is removed by further refinement process preceded by rich data region detection.

3.4.1 Rich data region detection:

The rich data region is the element in the page that contains all data records. A web page is parsed into the DOM tree and all elements containing some text are annotated as text elements. Algorithm 1 takes the DOM tree and in the first step searches keywords from the text of each DOM object. When a keyword is found in the text of the DOM object it is labeled as a keyword element. In the second step, a number of keywords for a DOM object are calculated including the sum of keywords in its children. In the third step, the frequent depth (i.e., the depth at which the maximum number of keywords lie) is found. In Step # 4 the element at frequent depth are identified and finally, the lowest common ancestor of all deep element is identified in the fifth step. Fig. 5 shows a pictorial diagram of the Rich Data Region detection process.

Algorithm 1: Rich data region detection

```
1. Identify searched keywords by inspecting texts contained in all elements on the page.
2. Calculate the number of keywords for all DOM objects including their descendant elements in HTML tree.
   $numberOfKeyWords = \sum_{i}^{n} \mathrm{KeyWords}(\mathrm{Child}_i)$
       Where n is the number of direct children of a node of a DOM object.
       $Child_i$ = ith child of an Element
3. Find frequent depth.
4. Calculate deep elements i.e., the elements with frequent depth.
5. Find the lowest common ancestor of deep elements i.e., the element containing a bigger chunk of deep elements.
```

The lowest common ancestor found in Algorithm 1 is the required rich data region because it will contain most of the deep elements, i.e., the elements with text containing searched keywords and are frequent in a depth in HTML tree and according to Observation OB8, we have observed that most of the keywords entered by user occur in rich data region because search process from underlying databases uses them as search criteria. After rich data region detection, the next step is a division of the region into data records which is done using Algorithm 2.

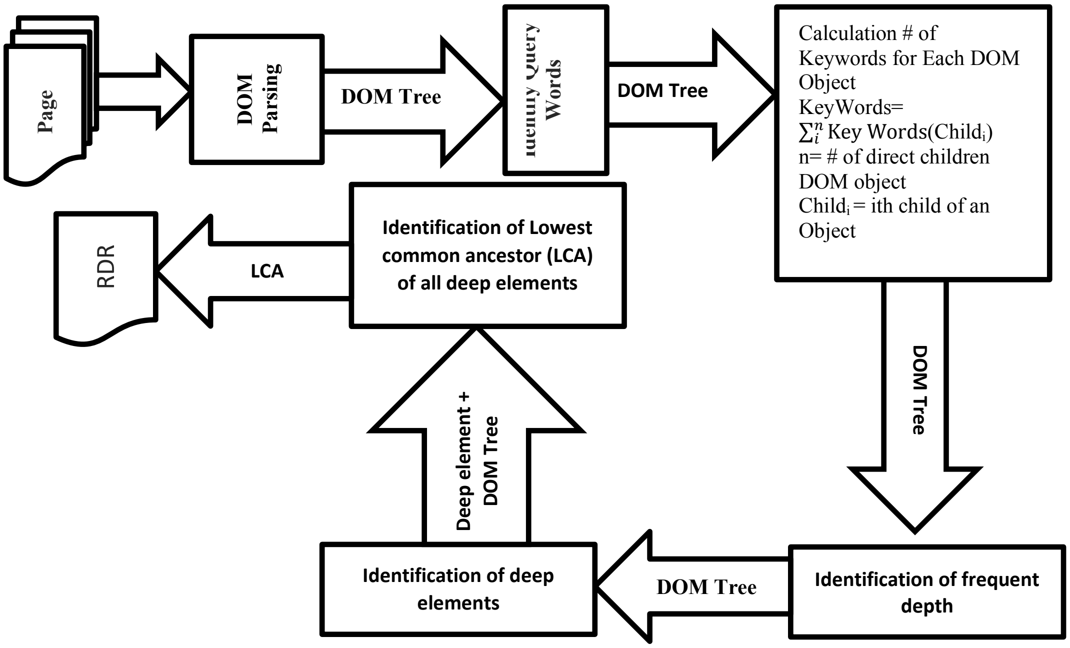

Page
DOM Parsing
DOM Tree
Identify Query Words
DOM Tree
Calculation # of Keywords for Each DOM Object
KeyWords= $\sum_{i}^{n}$ Key Words(Child$_i$)
n= # of direct children DOM object
Child$_i$ = ith child of an Object
DOM Tree
Identification of frequent depth
DOM Tree
Identification of deep elements
Deep element + DOM Tree
Identification of Lowest common ancestor (LCA) of all deep elements
LCA
RDR

Figure 5: Rich data region detection procedure

3.4.2 Lists detection and division

The candidate list is a DOM object having two or more child elements. Algorithm 2 divides the rich data region into data records:

Algorithm 2: Data records list detection

1. *All text elements in rich data region are annotated by adding an attribute "istextelement = true" to each DOM object.*
2. *Number of text elements including those of its children are calculated for a DOM object and an attribute "textElements = num_text_elements" is added. num_text_elementsis calculated as follows:*

   Number of text elements including those of its children are calculated for a DOM object and an attribute "textElements = num_text_elements" is added. num_text_elementsis calculated as follows:

$$num\_text\_elements = own\_text\_elements + \sum_{i=1}^{n} textElementsOfChild(i)$$

   where n is the number of direct child nodes of a DOM object

3. *Traverse the DOM Tree of a web page in breadth-first order and do the following:*
   a. *Extract all its immediate children of root element i.e, Rich Data Region*
   b. *For each child do step (a) recursively if it is divisible (An element is divisible if it has more than one keywords and text elements greater than*

```
        a threshold value).
    c.  For each non-divisible group of child elements, the longest common
        sequence in tag sequences of elements is calculated.
    d.  The group of elements is extracted as a candidate list if its elements'
        common tag sequence is relatively higher.
4.  Repeat steps 3b to 3d for all children of root element recursively.
```

Algorithm 2 outputs a list of candidate lists of data records. The next task is to detect the correct list of data records. Algorithm 2 takes Rich Data Region, calculates the number of text elements for each HTML DOM object, including a number of text elements of its descendant objects. Now all DOM objects are taken one by one and if it is not divisible then its tag sequence is generated. An object with more than one nondivisible children is considered a group of data records. Now common tag sequence is calculated for each group, with a value of CTS higher than a threshold value are output as a candidate list of data records. Fig. 6 shows a pictorial diagram of the data records list detection and division process.

3.4.3 Data records' list detection

Web pages contain menu bars to navigate other pages, lists of some featured products and banners. As the web site grows and more services are offered more pages are added and more options are included in the menus which increase their sizes. For styling purpose menus are encapsulated in other HTML objects e.g. lists, div, etc. hence the chance of detection of a menu as an object list is high, therefore, we have to devise a mechanism to remove false positives from candidate lists.

According to Observation OB9 "all data elements in a menu are formatted same way", therefore we picked detected candidate lists in above section one by one and inspect formatting of text elements contained in so-called data contained in that list and if a relatively larger contiguous chunk of text elements is formatted in similar styles then we reject the candidate list, for this purpose, we calculated Formatting Entropy Eq. (1) for a data object, which is defined as follows.

3.4.4 Formatting entropy

Zheng et al. [Zheng, Gu and Li (2012)] used structural-semantic entropy to identify a portion of the web page which they called "rich data node" using keywords known as "roles". They defined Structural Semantic Entropy as "summation of the product of the proportion of a role (to other text elements in a node) and log of the said proportion, with the negative sign". Higher the value of entropy, the higher the chance of a rich data node. We have defined formatting entropy Eq. (1) to identify data records from menu items. Formatting entropy Eq. (1) is defined on visual styles and the appearance of text elements in an HTML element.

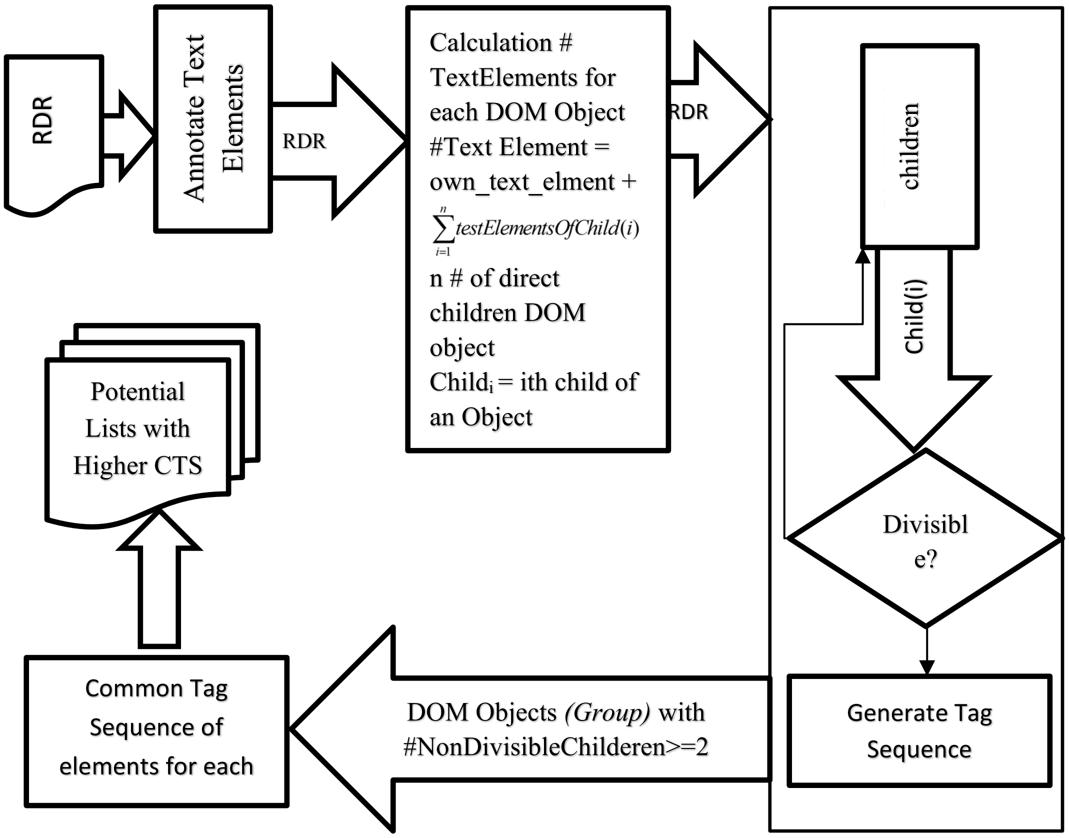

RDR
Annotate Text Elements
RDR
Calculation # TextElements for each DOM Object
#Text Element = own_text_elment + $\sum_{i=1}^{n} testElementsOfChild(i)$
n # of direct children DOM object
Child$_i$ = ith child of an Object
RDR
children
Child(i)
Divisibl e?
Generate Tag Sequence
DOM Objects *(Group)* with #NonDivisibleChilderen>=2
Common Tag Sequence of elements for each
Potential Lists with Higher CTS

Figure 6: List detection and division procedure

Text elements within a data record are formatted in a way to differentiate them from each other. Generally, two consecutive elements have a different visual appearance. We have used this property to differentiate data records from menu bars particularly drop-down menus because they resemble data records in terms of similarity and symmetry of visibility. Text elements in menu items are formatted in a similar fashion because all text elements in a menu bar represent semantically similar objects (i.e., navigation links to other web pages).

Entropy for HTML element is calculated as follows:

$$Formatting\ Entropy = -\sum_{1}^{n} S_i \log S_i \tag{1}$$

n=the total number of different styles used for text elements enclosed by an HTML element.

S$_i$ is the ratio of the style *i* to total the number of different styles used for text elements in an HTML element. Entropy Eq. (1) is a measure to calculate the diversity of formatting in an element. As discussed above text elements in data records are rendered with multiple visual styles whereas text elements in menu items are rendered with similar styles. Therefore, the value of entropy for data records will be higher than that of menu items.

3.4.5 Removal of noise

In a list of data objects, there are other irrelevant objects like advertisements and a block of links to the next results to show because when results are greater in number to show on a page they are divided into multiple pages. For this purpose, we used the Common Tag Sequence which is explained as under:

According to Observation OB4, in each data object, text elements are placed in the same order therefore in-order concatenation of tag names of elements of each data record will produce strings with a larger common sequence of tag names. Since all data records are of the same type, share the same template and have the same order of occurrence of text elements, therefore, they will have a greater common tag sequence and all other objects will have smaller ones. In view of the above, we have calculated the common tag sequence of each data object to all other data records and then calculated its length. To calculate the length of the common tag sequence we have considered HTML tag as a unit and not characters because a tag consists of more than one character. Finally, we have calculated the avg_CTS (average common tag sequence) Eq. (2) and normalized_avg_CTS (normalized average common tag sequence) Eq. (3) as follows:

$$avg\_CTS = \frac{\sum_{i=1}^{n-1} CTSi}{n} \tag{2}$$

where

n = total number of objects in a list

CTSi= common tag sequence of the current object to the object i.

$$normalized\_avg\_CTS = \frac{avg\_CTS}{LCTS} \tag{3}$$

where LCTS is the largest common tag sequence observed in the data object list. The CTS of the data object is higher than that of the noise objects because noise objects will not share structure with data records therefore their CTS will be less. We have dropped objects with smaller normalized_avg_CTS Eq. (3) than a threshold value.

4 Experimental design

In this Section, the experimental design and evaluation criteria are discussed. A prototype of a proposed system is developed using Java. JSOUP API (Application Programming Interface) [JSOUP] is used for parsing and feature extraction from HTML elements. JSOUP converts a web page into a well-formed W3C DOM model compliant document. To run the prototype a 64-bit Core-i5 machine with clock frequency 2.0 GHz and 4 GB RAM with Windows 10 Home and Oracle's Java Development Kit 1.8 was used. For development, Netbeans IDE 8.2 is used.

Dataset: To evaluate the system a dataset is selected that is used in DCADE [Yuliana and Chang (2020)]. DCADE is a recently published proposal in the domain of web data extraction (i.e., 2019). Although primarily proposed system addresses business domain its successful application on diverse types of dataset shows that it is applicable for other domains as well with slightly different parameters.

We have sent different queries to websites and saved the resultant web pages. The numbers of original data records present on each page are also recorded. The saved pages are input to the system to extract data records. The numbers of correctly extracted data records are compared with the original number of data records present on the web pages. The result shows that the efficiency of the proposed system is higher than the existing systems. Tab. 1 shows the properties of the dataset for our experiments. The first column shows the category of web documents, second column is a list of websites from where we have obtained web documents. The third column shows the size of each document in kilobytes. Finally, in the fourth column number of data records is provided against each document we observed manually.

Table 1: Summary of the dataset

Category	URL	Doc Size (KBs)	Data Records
Books	www.abebooks.com	193.00	30
	www.awesomebooks.com	75.00	10
	www.betterworldbooks.com	135.00	10
	www.manybooks.net	29.00	11
	www.waterstones.com	85.00	10
Cars	www.autotrader.com	696.00	22
	www.carmax.com	81.00	10
	www.carzone.i.e.	122.00	10
	www.classiccarsforsale.co.uk	182.00	13
	www.internetautoguide.com	170.00	15
Events	www.allconferences.com	164.00	20
	www.mbendi.com	98.00	20
	www.rdlearning.org.uk	16.00	10
Doctors	doctor.webmd.com	89.00	13
	extapps.ama-assn.org	47.00	10
	www.drscore.com	34.00	10
	www.steadyhealth.com	56.00	10
Jobs	careers.insightintodiversity.com	49.00	50
	www.4jobs.com	229.00	15
	www.6figurejobs.com	74.00	10
	www.careerbuilder.com	221.00	25
	www.jobofmine.com	31.00	10
Movies	www.allmovie.com	49.00	10
	www.citwf.com	53.00	100
	www.disneymovieslist.com	52.00	10
	www.imdb.com	135.00	200
	www.soulfilms.com	78.00	20
Real estate	realestate.yahoo.com	132.00	6
	www.haart.co.uk	128.00	16
	www.homes.com	147.00	16
	www.remax.com	279.00	9

	www.trulia.com	313.00	18
Sports	en.uefa.com	96.00	15
	www.atpworldtour.com	94.00	10
	www.nfl.com	206.00	25
	www.soccerbase.com	180.00	80
EXALG	teams.uefa.com	96.00	15
	www.ausopen.com	68.00	15
	www.ebay.com	200.00	50
	www.majorleaguebaseball.com	123.00	47
	www.rpmfind.net	6.00	8

4.1 Evaluation

In the research process evaluation is a very critical and essential phase. Donnelly et al. defined evaluation as "while evaluating a system or technique its worth, consistency, efficiency and importance is judged, using some predetermined criteria, in a systematic way" [Donnelly and Trochim (2007)].

4.2 The objective of evaluation

The major objective of the evaluation is to test the validity of techniques, observations, and heuristics presented in the paper. There are mainly three contributions of our work: (i) Rich data region (i.e., region on the page where data of interest lie) detection (ii) Division of data region into data records (iii) Removal of noisy elements from data records. The main purpose of the evaluation is to provide evidence that proposed techniques perform the above-mentioned tasks efficiently and correctly.

4.3 Selection of evaluation technique

We have selected a system-centric based empirical approach to evaluating our system due to the reason that the system we are going to evaluate is an un-supervised web data extraction system. Therefore, we have decided to evaluate its efficiency and correctness of results. A comparison of time consumption of Rich Data Region detection with the said task proposed by Alvarez et al. [Álvarez, Pan, Raposo et al. (2008)] is made. In literatures [Zheng, Gu and Li (2012)] and [Grigalis (2013)] has used precision-based data extraction, while [Liu, Meng and Meng (2009); Weng, Hong and Bell (2011)] discuss the precision and recall to measure the effectiveness of data extraction. Precision , recall and F-measure are explained as follows:

4.3.1 Precision

Precision Eq. (4) shows the correctness of system output. It is calculated as follows:

$$Precision = \frac{CERs}{TERs} \quad (4)$$

where

CERs=number of *Correctly Extracted data Records*
TERs=number of *Total Extracted data Records*

4.3.2 Recall

Recall Eq. (5) shows the ratio of correctly extracted results to the number of total data records present on the page. It is calculated as follows:

$$Recall = \frac{CERs}{TDRs} \tag{5}$$

where
CERs=number of Correctly Extracted Results
TDRs=number of Total Data Records present on the page

4.3.3 F-measure

Precision Eq. (4) shows the percentage of truly extracted data, from extracted results, with respect to total extracted data records which is actually the quality of extracted results i.e., how much noise is present in extracted results, e.g., 0.98 precision means 2% noise and 98% original data in extracted results but it does not consider the ratio of truly extracted data records to total data records present in the page. Recall Eq. (5) shows the percentage of truly extracted data records, from extracted results, with respect to total data records present on the page which actually shows that how much data records are extracted from total records present on the page, e.g., 0.98 recall means 98% of data records presents on the page have been extracted successfully and 2% could not be extracted. The recall does not consider the noise in the extracted data. Therefore, another metric F-measure Eq. (6) is used to approximate precision and recall both which is a harmonic mean of precision and recall and can be calculated as follows:

$$F\text{ - }measure = 2 \times \frac{precision \times recall}{precision + recall} \tag{6}$$

A higher value of F-measure Eq. (6) means higher values of both precision and recall, i.e., the higher ratio of data extraction with less noise.

5 Results and analysis

In Section 4, we have presented details of the experimental setup to implement and evaluate the system. We have also described precision, recall and F-measure, the metrics used for evaluation of results of the proposed system. In this chapter results of experiments are presented and the efficiency of the proposed system is analyzed based on predefined metrics

5.1 Effectiveness analysis

In this section, we will discuss the effectiveness of the proposed system as compared to other systems. As described in Section 4, precision Eq. (4) , recall Eq. (5) and F-measure Eq. (6) are used to test the effectiveness of data extraction systems. We borrowed results of precision, recall and F-measure from DCADE and experiments are run on the same dataset.

5.1.1 Precision

Precision Eq. (4) is defined as a measure of the quality of the extracted data. A higher value of precision means the extracted data contains less noise and low precision means extracted data contains greater noise. We compared the average precision of the proposed technique with DCADE [Yuliana and Chang (2020)], DCA [Yuliana and Chang (2018)], UWIDE [Chang, Chen, Chen et al. (2016)], TEX [Sleiman and Corchuelo (2013)] and RoadRunner [Crescenzi, Mecca and Merialdo (2001)] as shown in Fig. 7. Fig. 7 shows that the proposed technique extracts data with high quality as compared to other techniques.

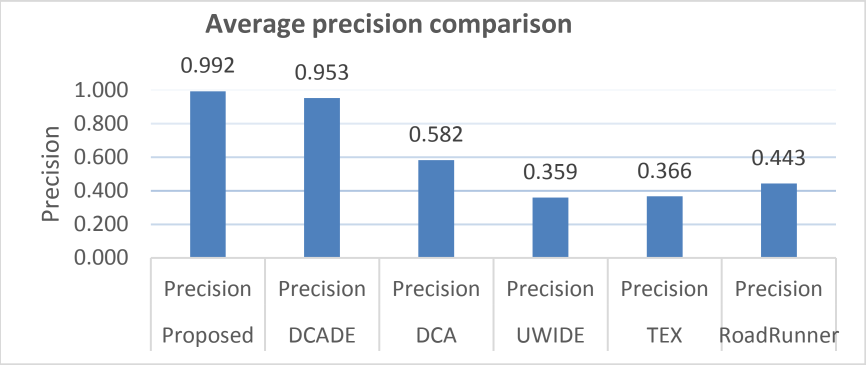


Figure 7: Comparison average of precision of the proposed technique with DCADE, DCA, UWIDE, TEX and RoadRunner

5.1.2 Recall

Recall Eq. (5) measures how much data on the page is extracted. A higher value of recall means most of the data on a web page is extracted. We have compared average recall of proposed technique with DCADE [Yuliana and Chang (2020)], DCA [Yuliana and Chang (2018)], UWIDE [Chang, Chen, Chen et al. (2016)], TEX [Sleiman and Corchuelo (2013)] and RoadRunner [Crescenzi, Mecca and Merialdo (2001)] in Fig. 8. On the vertical axis, recall is shown from 0.0 to 1.0 and on horizontal axis average recall of all systems is shown. Fig. 8 shows that the proposed technique extracts almost all the data present on the page whereas other techniques except DCADE have a very low ratio of data extraction.

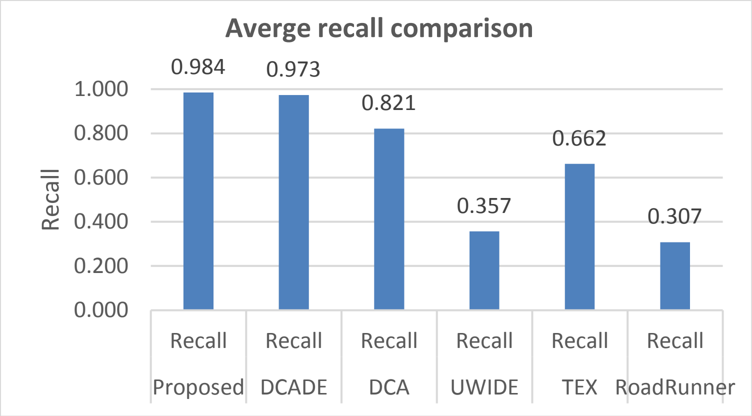


Figure 8: Comparison average of recall of the proposed technique with DCADE, DCA, UWIDE, TEX and RoadRunner

5.1.3 F-measure

F-measure Eq. (6) is a harmonic mean of precision and recall. Recall takes into account both, the quality of data and ratio of extraction to the total data present on the page. A higher value of recall means both precision and recall are higher. We have compared average F-measure of the proposed technique with DCADE [Yuliana and Chang (2020)], DCA [Yuliana and Chang (2018)], UWIDE [Chang, Chen, Chen et al. (2016)], TEX [Sleiman and Corchuelo (2013)] and RoadRunner [Crescenzi, Mecca and Merialdo (2001)] in Fig. 9. F-measure is shown on the vertical axis from 0.0 to 1.0 and on horizontal axis average F-measure of all systems. Fig. 9 shows that the proposed technique extracts almost all the data present on the page with minimum noise whereas other techniques except DCADE have a very low ratio of data extraction and high ratio of noise.

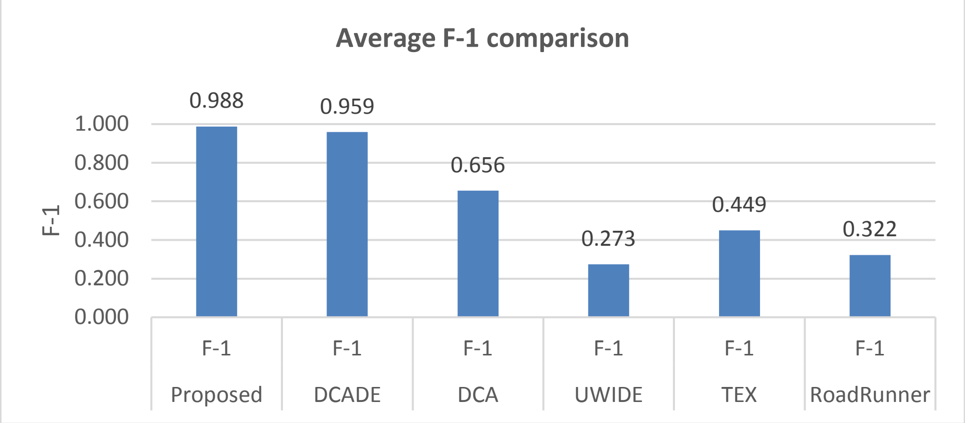


Figure 9: Comparison average of F-1 of the proposed technique with DCADE, DCA, UWIDE, TEX and RoadRunner

6 Conclusion

In this paper, we have presented a novel technique to extract data records from web pages served by web servers in response to queries issued by the user**.** We have used visual cues, observations, and heuristics to identify data records on a web page. We have observed that text elements in data records are formatted in such a way that they look different from each other. It helps the visitors to differentiate different attributes of data records like price, size, and vender. We have also observed that menu elements of pages share formatting. Therefore, we have used formatting entropy to identify the data records. The value of entropy for data records is higher than a menu item because links in menus are formatted similarly because they represent a similar type of information. The proposed technique takes a single page as input, parses it into HTML elements, and applies above-discussed techniques to Detect Rich Data Region. The Rich Data Region is divided into potential data record lists. Potential data records are validated using a common tag sequence, style similarity, and formatting entropy.

Funding Statement: This research work is supported by Data and Artificial Intelligence Scientific Chair at Umm Al-Qura University, Makkah City, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Álvarez, M.; Pan, A.; Raposo, J.; Bellas, F.; Cacheda, F. (2008): Extracting lists of data records from semi-structured web pages. *Data & Knowledge Engineering*, vol. 64, no. 2, pp. 491-509.

Appelt, D.; Hobbs, J.; Bear, J.; Israel, D. J.; Tyson, M. (1993): FASTUS: a finite-state processor for information extraction from real-world text. *Proceedings of IJCAI-93*, Chambery, France.

Chang, C. H.; Chen, T. S.; Chen, M. C.; Ding, J. L. (2016): Efficient page-level data extraction via schema induction and verification. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 478-490.

Chang, C. H.; Kuo, S. C. (2004): OLERA: Semisupervised web-data extraction with visual support. *IEEE Intelligent Systems*, vol. 19, no. 6, pp. 56-64.

Chang, C. H.; Lui, S. C. (2001): IEPAD: Information extraction based on pattern discovery. *Proceedings of the 10th International Conference on World Wide Web*, pp. 681-688.

Crescenzi, V.; Mecca, G.; Merialdo, P. (2001): Roadrunner: Towards automatic data extraction from large web sites. *VLDB*, pp. 109-118.

Cunningham, H.; Maynard, D.; Bontcheva, K.; Maynard, H.; Tablan, V. (2002): GATE: A framework and graphical development environment for robust NLP tools and applications. *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 168-175.

Donnelly, J.; Trochim, W. (2007): *The Research Methods Knowledge Base*. Ohio: Atomic Dog Publishing.

Doorenbos, R. B.; Etzioni, O.; Weld, D. S. (1997): A scalable comparison-shopping agent for the world-wide web. *Proceedings of the First International Conference on Autonomous Agents*, pp. 39-48.

Grigalis, T. (2013): Towards web-scale structured web data extraction. *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, pp. 753-758.

Hiremath, P.; Algur, S. P. (2009): Extraction of data from web pages: A vision based approach. *International Journal of Computer and Information Science and Engineering*, vol. 3, no. 1.

Hobbs, J. R.; Appelt, D.; Tyson, M.; Bear, J.; Israel, D. (1992). SRI International: description of the FASTUS system used for MUC-4. *SRI International Menlo Park CA*.

Huffman, S. (1996): Learning information extraction patterns from examples. *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, Springer-Verlag.

JSOUP. https://jsoup.org/.

Kaiser, K.; Miksch, S. (2005). Information extraction-a survey. *Technical Report, Vienna University of Technology, Institute of Software Technology and Interactive Systems.*

Kim, J. T.; Moldovan, D. I. (1995): Acquisition of linguistic patterns for knowledge-based information extraction. *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, no. 5, pp. 713-724.

Kobayashi, N.; Inui, K.; Matsumoto, Y. (2007): Opinion mining from web documents: Extraction and structurization. *Information and Media Technologies*, vol. 2, no. 1, pp. 326-337.

Kotler, P. (2009): *Marketing Management: A South Asian Perspective.* Pearson Education India.

Krupka, G.; Jacobs, P. S.; Rau, L.; Childs, L. C.; Sider, I. (1992): GE NLTOOLSET: Description of the system as used for MUC-4. *Fourth Message Undersanding Conference*, McLean, Virginia.

Kushmerick, N.; Weld, D. S.; Doorenbos, R. (1997): *Wrapper Induction for Information Extraction*. University of Washington, Washington.

Liu, W.; Meng, X.; Meng, W. (2009): ViDE:A vision-based approach for deep web data extraction. *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 3, pp. 447-460.

Miao, G.; Tatemura, J.; Hsiung, W. P.; Sawires, A.; Moser, L. E. (2009): Extracting data records from the web using tag path clustering. *Proceedings of the 18th International Conference on World Wide Web*, pp. 981-990.

Mooney, R. (1999): Relational learning of pattern-match rules for information extraction. *Proceedings of the Sixteenth National Conference on Artificial Intelligence*.

Mundluru, D.; Raghavan, V. V.; Wu, Z. (2010): Automatically extracting web data records. *International Conference on Active Media Technology*, pp. 510-521.

Muslea, I.; Minton, S.; Knoblock, C. (1999): A hierarchical approach to wrapper induction. *Proceedings of the Third Annual Conference on Autonomous Agents*, pp. 190-197.

Prieto, V. M.; Alvarez, M.; López-García, R.; Cacheda, F. (2012): A scale for crawler effectiveness on the client-side hidden web. *Computer Science and Information Systems,* vol. 9, no. 2, pp. 561-583.

Riloff, E. (1993): Automatically constructing a dictionary for information extraction tasks. *AAAI,* vol. 1, no. 1, pp. 811-816.

Rivero, C. R.; Frantz, R. Z.; Ruiz, D.; Corchuelo, R. (2011): On using high-level structured queries for integrating deep-web information sources. *Integration*, vol. 16, pp. 37.

Sleiman, H. A.; Corchuelo, R. (2013): TEX: An efficient and effective unsupervised web information extractor. *Knowledge-Based Systems*, vol. 39, pp. 109-123.

Soderland, S. (1999): Learning information extraction rules for semi-structured and free text. *Machine Learning*, vol. 34, no. 1-3, pp. 233-272.

Soderland, S.; Fisher, D.; Aseltine, J.; Lehnert, W. (1995): CRYSTAL: Inducing a conceptual dictionary. arXiv:cmp-lg/9505020.

Wang, J.; Lochovsky, F. H. (2002): Wrapper induction based on nested pattern discovery. *World Wide Web Internet and Web Information Systems*, pp. 1-29.

Wang, J.; Lochovsky, F. H. (2003): Data extraction and label assignment for web databases. *Proceedings of the 12th International Conference on World Wide Web*, pp. 187-196.

Wang, Y.; Lu, J.; Chen, J. (2009): Crawling deep web using a new set covering algorithm. *International Conference on Advanced Data Mining and Applications*, pp. 326-337.

Wang, Y.; Lu, J.; Liang, J.; Chen, J.; Liu, J. (2012): Selecting queries from sample to crawl deep web data sources. *Web Intelligence and Agent Systems: An International Journal*, vol. 10, no. 1, pp. 75-88.

Weng, D.; Hong, J.; Bell, D. A. (2011): Extracting data records from query result pages based on visual features. *British National Conference on Databases*, pp. 140-153.

Yangarber, R.; Grishman, R. (1998): NYU: Description of the Proteus/PET system as used for MUC-7 ST. *Seventh Message Understanding Conference*, Fairfax, Virginia.

Yuliana, O. Y.; Chang, C. H. (2018): A novel alignment algorithm for effective web data extraction from singleton-item pages. *Applied Intelligence*, vol. 48, no. 11, pp. 4355-4370.

Yuliana, O. Y.; Chang, C. H. (2020): DCADE: Divide and conquer alignment with dynamic encoding for full page data extraction. *Applied Intelligence*, vol. 50, no. 2, pp. 271-295.

Zheng, X.; Gu, Y.; Li, Y. (2012): Data extraction from web pages based on structural-semantic entropy. *Proceedings of the 21st International Conference on World Wide Web*, pp. 93-102.