Reducing Human Effort: Web Data Mining, Learning a New Characteristics from Big Data

Mr. M. Srinivasan
Associate Professor
Department of Information & Technology
Priyadarshini Engineering College, Vaniyambadi, Vellore, India

Dr. S. Koteeswaran
Associate Professor
Department of Computer Science & Engineering
Vel Tech University, Chennai, India

Abstract

This paper presents a Reducing Human Effort: Web Data Mining, Learning a New Characteristics from Big data, reducing human effort in extracting precise information from undetected Web sites. Our approach aims at automatically adapting the information extraction knowledge previously learned from a source Web site to a new undetected site, at the same time, discovering previously undetected attributes. There is a two kinds of text related evidences from the source Web site are considered. The first kind of evidences is obtained from the extraction pattern contained in the previously learned wrapper. The second kind of evidences is derived from the previously extracted or collected items. A generative model for the generation of the web site independent content information and the site dependent layout format of the text fragments related to attribute values contained in a Web page is designed to connect the insecurity involved. We have conducted extensive experiments from more than 50 real world Web sites in more than five different domains to demonstrate the effectiveness of our context.

Keywords- Big Data, DOM, Extraction Pattern, Wrapper Learning & Adaption

I. INTRODUCTION

Information extraction systems aim at automatically extracting precise and exact text fragments from documents. They can also transform largely unstructured information to structured data for further intelligent processing.

A common information extraction technique for semi structured documents such as Web pages are known as wrappers. A wrapper normally consists of a set of extraction rules which were typically manually constructed by human experts in the past. Recently, several wrapper learning approaches have been proposed for automatically learning wrappers from training examples.

For instance, consider a Web page shown in Fig. 1 collected from a Web site1 in the book catalog domain. To learn the wrapper for automatically extracting information from this Web site, one can manually provide some training examples. For example, a user may label the text fragment "C, C++Programming" as the book title, and the fragments "Dr. Balagurusamy" as the corresponding authors. A wrapper learning method can then automatically learn the wrapper based on the text patterns embedded in training examples, as well as the text patterns related to the layout format embodied in the HTML document.

The learned wrapper can be applied to other Web pages of the same Website to extract information. Wrapper learning systems can significantly reduce the amount of human effort in constructing wrappers. Though many existing wrapper learning methods can effectively extract information from the same Web site and achieve very good performance, one restriction of a learned wrapper is that it cannot be applied to previously undetected Web sites, even in the same domain.

For example, the wrapper previously learned from the source Web site shown in Fig. 1 can be adapted to the new undetected site Shown in Fig.2 the adapted wrapper can then be applied to Web pages of this new site for extracting data records. Consequently, it can significantly reduce the human effort in preparing training examples for learning wrappers for different sites. Another shortcoming of existing wrapper learning techniques is that attributes extracted by the learned wrapper are limited to those defined in the training process.

As a result, they can only handle pre specified attributes. For example, if the previously learned wrapper only contains extraction patterns for the attributes title, author, and price from the source Web site shown in Fig. 1, the adapted wrapper can at best extract these attributes from new undetected sites. However, a new undetected site may contain some new additional attributes that do not appear in the source Web site. For instance, book records in Fig.2 contain the attribute ISBN that does not exist in Fig.1. The ISBN of the book records cannot be extracted. This observation leads to another objective of this paper. We investigate the problem of new attribute discovery which aims at extracting the unspecified attributes from new undetected sites. New attribute discovery can effectively deliver more useful information to users.

II. RELATED WORK

Previous proposed a method which alleviates the problem of manually preparing training data by investigating wrapper adaptation. From number of Web sites some rules are learned and these rules are used for data extraction. One disadvantage of this method is that training examples from several Web sites must be collected to learn such heuristic rules.

Here bootstrapping data repository is assumed, which is called as source repository that contains a set of objects belonging to the same domain. This approach assumes that attributes in source repository must match the attributes in new web site. However, exact matching is not possible. The training stage consists of background knowledge acquisition, where data is collected in a particular domain and a structural description of data is learned. Now based on learned rules data from new site is extracted. The extracted data are then organized in a table format.

Each column of the table is labeled by matching with the entries in the column and the patterns learned in the source site. It provides only a single attribute for the entire column which, may consists of inconsistent or incorrectly extracted data.
Generalized node of length r consists of r nodes in the HTML tag tree with the following two properties:

1) The nodes all have the same parent.
2) The nodes are adjacent.

A data region is a collection of two or more generalized nodes.
This method works as follows,

1) Step 1: Build a HTML tag tree of the page.
2) Step 2: Mining data regions in the page using the tag tree and string comparison.
3) Step 3: Identifying data records from each data region.

This method suffers from a major drawback that it cannot differentiate the type and the meaning of the information extracted. Hence, the items extracted require human effort to interpret the meaning.

III. PROBLEM DEFINITION

To take domain = D, for example book domain which contains number of pages.
P= {p1, p2, p3...}.
A page contains number of records.
R= {r1, r2, r3...}.
Particular record contains number of attributes.
A= {a1, a2, a3...}.
For example book domain site contains web pages which in turn consist of book records.
A record consists of attributes like title, author and price.

A. Wrapper Learning:

Wrapper is the common system used to extract information form web site. Given a set of web pages P, goal of wrapper is Fig 1: Sample Web page to extract records from these web pages. Wrap(w1) is wrapper for web site w1. To extract records from site w1 Wrap(w1)should be trained with training examples of site w1. Wrap(w1)will be learned by using training examples of site w1.

B. Wrapper Adaptation:

Wrapper created for one web site cannot be directly used to extract information from another web site even in the same domain. Wrapper adaptation aims at automatically learning wrapper Wrap(w2) for the Web site (w2) without any training examples from (w2), such that the adapted wrapper Wrap(w2)can extract text fragments belonging to the pages of (w2).

C. New Attribute Discovery

New attribute discovery aims at automatically identifying attributes which were not present in web site w1. For instance, suppose we have a wrapper which can extract the attributes title, author, and price of the book records in the Web site shown in fig 1. New attribute discovery can identify the text fragments referring to the previously undetected attributes such as ISBN, publisher etc.. as shown in fig 2.

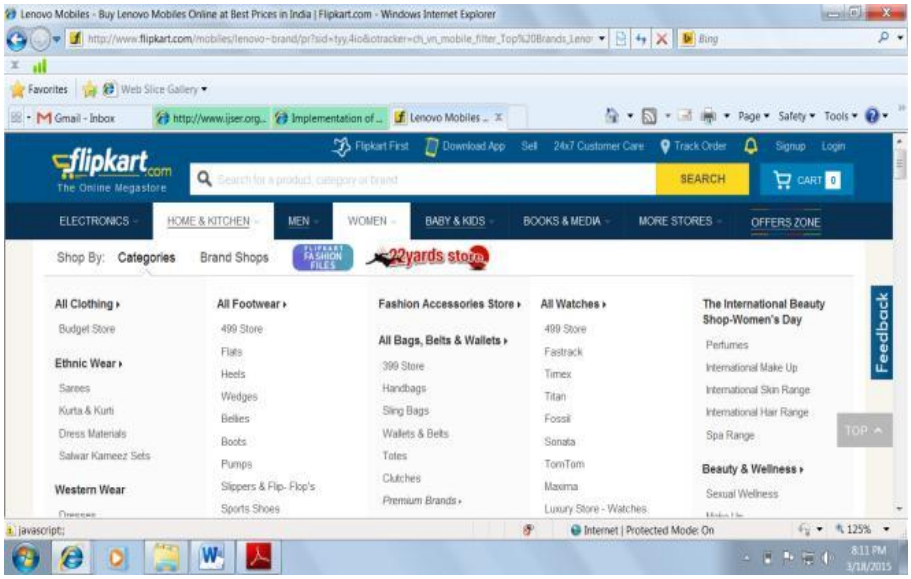

Fig. 1: Sample Web page

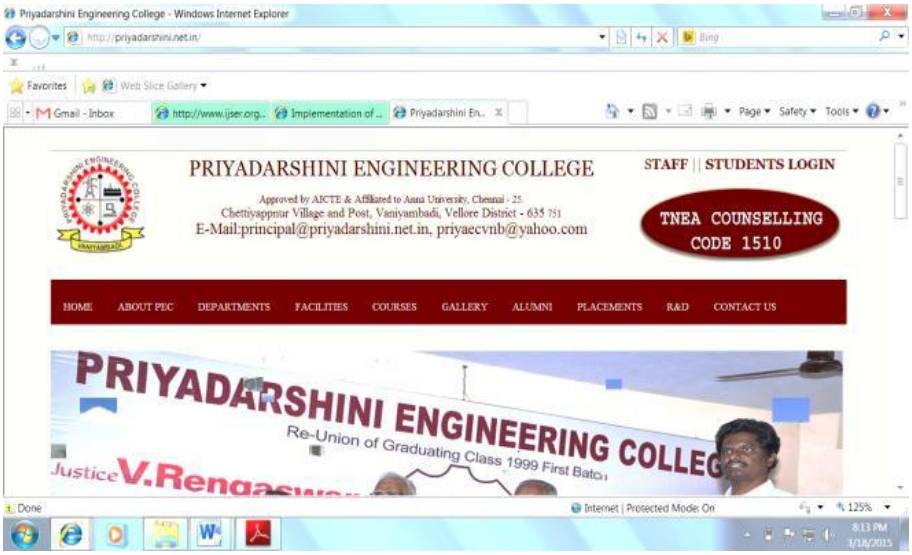

Fig 2: Sample Web page

Fig. 2.1: Sample Web page

IV. PROPOSED SYSTEM

In order to adapt information extraction wrapper to new site we need to take sample web pages of that site for training. Web pages of a site are divided in two sets. First set (training set) contains two web pages and are used for training. Second set (testing set) contains remaining pages of same site and are used for testing.

A. Steps:

1) Selecting Training Data:

We provide two web pages of a site for training. For example in a book domain select a page which contains all records of "C,C++" and select second page which contains all records of "C programming". These web pages are used as training set for the wrapper.

2) Useful Text Fragments Identification:

To identify useful text fragments from web page, web page can be considered as DOM structure. It is tree like structure. Internal nodes of this tree are HTML tags and leaf nodes are the text fragments displayed on the browser. Each text fragment is associated with a root-to-leaf path, which is the concatenation of the HTML tags as shown in fig 4. Suppose we have two Web pages of the same site containing different records. The text fragments related to the attributes of a record are likely to be different, while text fragments related to the Irrelevant information such as advertisements, listings or copyright statements is likely to be similar in both the pages.

In DOM (Document Object Model) tree representation all anchor tags are considered from both the web pages of same site.

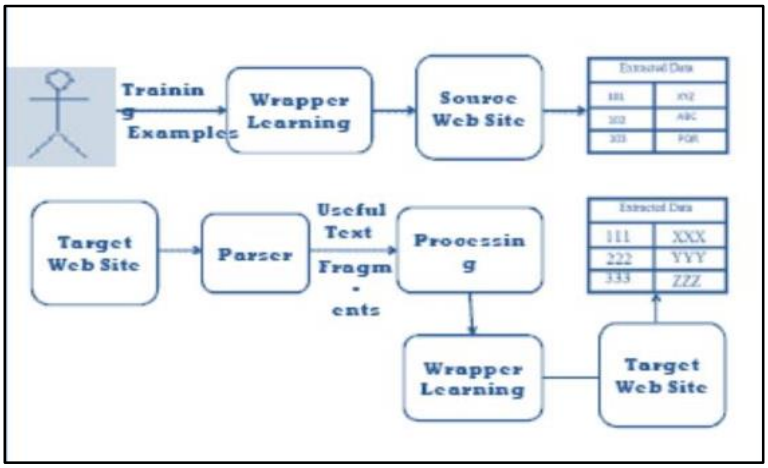

Fig. 3: "Automatic wrapper adaptation system"

Anchor tags related to title of the book are likely to be different, but anchor tags related to other information such as listings of categories advertisements are likely to be similar. Delete the entire anchor tags which have same contents on both web pages. Remaining are the useful text fragments.

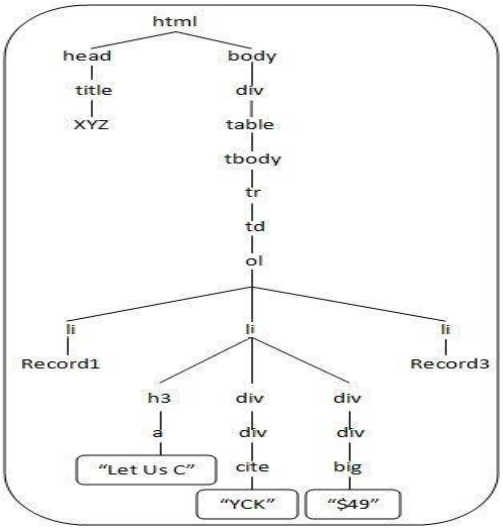

Fig. 4: Sample tag tree of a record.

Here not all the text fragments are related to book records. Still there are some text fragments which are not related to any attribute of a book record.

3) Processing Useful Text Fragments:

Now we have all anchor tags which are different on both web pages. Generally in a book domain titles of books are represented using anchor tags. Here we try to find those anchor tags which are related to titles of books. Data contained in these anchor tags is processed.

a) Remove Stop Words:

Stop words like a, and, an etc... Must be deleted first from useful text fragments as our next step in this method is frequency count. Stop words may be more in number on a web page and so they need to be deleted. Otherwise frequency count of stop words will be more than other useful words. Some of the stop words listed below will be deleted.

Stop. add () is
method. stop. add
("in"); stop. add
("an"); stop. add
("for"); stop. add
("the"); stop. add
("a");

b) Frequency Count:

After removing stop words from useful text fragments, count the frequency of each word in the remaining text fragments. For example our web page used for training contains 100 records of "C,C++" books. Each record will contain "C, C++" word in the attribute title. Get the word which has maximum frequency count value. In our case "C, C++" will be the most frequent word.

4) Locate The Path:

Word with maximum frequency will give you the attribute title. Anchor tag of title of a book will be considered. Anchor tag is at leaf of the DOM tree representation. Find root to leaf path of title of the book. To find root to leaf path, go upward in DOM tree by finding parent of each tag until root is determined. This path will give you the tag tree for attribute title.

Other attributes of a record will be present in between two titles. We consider some features to locate the path of these attributes. We consider following features.

1) Each word of a title of a book contains first letter in capitals.
2) Author name is present immediately after title or may contain "by" keyword.
3) Author name may be in italic or bold or may contain semantic label "author".
4) Price of a book may contain symbols like $ or ₹ Price are numeric values and generally are bold
5) ISBN of a book contains semantic label "ISBN" with numeric value and is in capitals always. In this way by considering features of various attributes of records we can locate all the attributes in the web page.

For example following is the path from root to leaf for attribute title.

<html>
<body>

<table>
<body>
<tr>< td >
< li >< h3 >< a >

These paths are used to train the wrapper. Wrapper is learned by using these paths and applied to remaining web pages of the site which is our testing set. Now by using these rules (paths) our wrapper can easily extract records from testing pages.

V. EXPERIMENTAL RESULTS

We conducted experiments on 8 real world Web sites collected from two domains, namely, the book domain and the electronics appliance domain to evaluate the performance of our framework.

Table 1: List of web sites

A1	*Amazon (www.amazon.com*
B1	*Powell's Books (www.powells.com)*
B2	*Abe books*
B3	*Rediff (www.rediff.com)*
B4	*eBay (www.ebay.com)*
E1	*Shoptronics (www.shoptronics.in)*
E2	*Homeshop18(www.homeshop18.com)*
E3	*Flipkartwww.flipkart.com*

E4	*Engg.Collegewww.priyadarshini.net.in*

Table 1 depicts the Web sites used in our experiment. B, B2, B3, B4 are from book domain and E1, E2, E3, E4 are from electronic appliances domain. Data is extracted from all the above listed sites (Table 1) by using automation anywhere and our method. The extraction performance is evaluated by two commonly used metrics, namely, precision and recall. Precision is defined as the number of items for which the system correctly identified divided by the total number of items it extracts.

Recall is defined as the number of items for which the system correctly identified divided by the total number of actual items. The results indicate that after applying our full wrapper adaptation approach, the wrapper learned from a particular Web site cane adapted to other sites. Our wrapper adaptation approach achieves better performance compared with Automation anywhere. Table 2 and Table 3 show the comparison of results for book domain and electronic appliances domain respectively.

Table 2: Extraction performance for book domain

Website	*Automation Anywhere*		*Our approach*	
	P (%)	*R (%)*	*P (%)*	*R (%)*
B1	*99.4*	*90.8*	*99.2*	*86.1*
B2	*88.8*	*79.0*	*98.0*	*90.0*
B3	*70.0*	*76.0*	*90.0*	*100.0*
B4	*91.6*	*100.0*	*93.2*	*100.0*

Graph represents precision and recalls of both domains. P1 and P2 are precisions of extracted data by Automation anywhere and our approach respectively. Similarly, R1 and R2 are recalls of extracted data by Automation anywhere and our approach respectively.

Table 3: Extraction Performance for Electronics Appliances Domain

Website	*Automation Anywhere*		*Our approach*	
	P (%)	*R (%)*	*P (%)*	*R (%)*
W1- E1	*95.9*	*90.7*	*100.0*	*92.0*
W2- E2	*100.0*	*75.0*	*100.0*	*98.5*
W3- E3	*97.0*	*62.5*	*98.3*	*99.0*
W4- E4	*98.7*	*66.6*	*96.3*	*66.6*

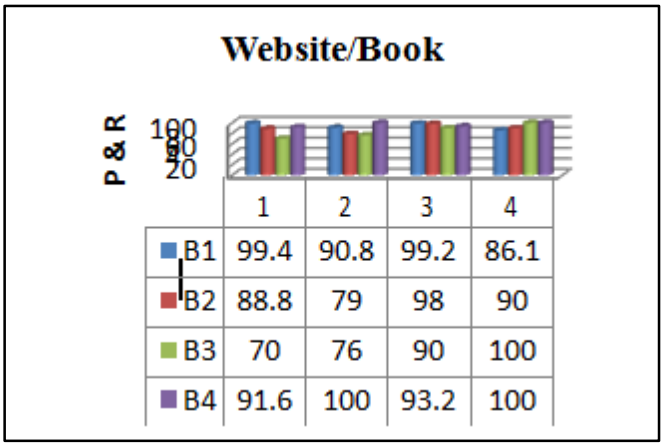

Fig. 5: Website/book

VI. CONCLUSION

We have a system for adapting information extraction wrappers with new attribute discovery. Our approach can automatically adapt the information extraction patterns for new undetected sites; at the same time can discover new attributes. DOM tree representation is used for the generation of useful text fragments related to the attributes and to find paths of those attributes from root to leaf. DOM tree technique with path identification is employed in our framework for tackling the wrapper adaptation and new attributes discovery tasks. Experiments for real world websites in different domains were conducted and the results demonstrate that our method achieves a very promising performance.

REFERENCES

[1] Tak-Lam Wong and Wai Lam, "Learning to Adapt Web Information Extraction Knowledge and Discovering New Attributes via a Bayesian Approach".
[2] W. Cohen and W. Fan, "Learning Page- Independent Heuristics for Extracting Data from Web Pages," Computer Networks, vol. 31, nos. 11-16, pp. 1641-1652, 1999.
[3] P. Golgher and A. da Silva, "Bootstrapping for Example-Based Data Extraction," Proc. 10th ACM Int'l Conf. Information and Knowledge Management (CIKM), pp. 371-378, 2001.
[4] K. Lerman, C. Gazen, S. Minton, and C. Knoblock