Automatic wrapper adaptation system

A. A. Tekale, *S. S. Nandgaonkar*

Department of computer engineering, VPCOE, Baramati, Pune.

tekale.abhijeet@gmail.com

sushma.nandgaonkar@gmail.com

Abstract— Extracting precise information from Web sites is a useful task to obtain structured data from unstructured or semi structured data. This data is useful in further intelligent processing. Wrappers are the common information extraction systems which will transform largely unstructured information to structured data. Method in this paper is meant for extracting Web data. Some of the existing techniques require manually preparing training data and some does not require manual intervention. Wrapper generated for one site cannot be directly applied to new site even if the domain is same. Some methods only extract those data attributes which are specified in wrapper but, unseen Web pages may have additional attributes which needs to be identified. Automatically adapting the information extraction knowledge to a new unseen site, at the same time, discovering previously unseen attributes is the challenging task. Proposed system learns information extraction knowledge for new web site is automatically. New attributes will be discovered as well.

Index Terms— Wrapper Learning, Wrapper Adaptation, Web Mining, DOM Tree.

I. INTRODUCTION

Information extraction systems aim at automatically extracting exact data from documents. They can also transform largely unstructured information to structured data. A common information extraction technique for semi structured documents such as Web pages is known as wrappers. A wrapper consists of a set of extraction rules. Previous technique required manually preparing set of rules to construct a wrapper. Semi-automatic technique requires training a wrapper manually first and then using the same wrapper for remaining Web pages of same site for automatically extracting information. One restriction of a learned wrapper is that it cannot be applied to previously unseen Web sites, even in the same domain. To construct a wrapper for an unseen site, a separate human effort for the preparation of training examples is required.

Information extraction system should reduce the manual effort required to prepare training examples by wrapper adaptation which aims at automatically adapting a previously learned wrapper from one Web site, known as a source Web site, to new unseen sites in the same domain. 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 these wrappers at best can extract prespecified attributes only. A new unseen site may contain some additional attributes which are not present in the source Web site. We survey the problem of new attribute discovery which aims at extracting the unspecified attributes from new unseen sites. New attribute discovery can effectively deliver more useful information to users.

II. RELATED WORK

Cohen and fan [2] 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. Golgher and Silva [3] proposed bootstrapping method which tries to solve the wrapper adaptation problem. Here a 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. Lerman, Gazen, Minton, and Knoblock [4] suggested a method called ADEL which is able to extract records from Web sites and semantically label the attributes in new unseen sites. 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 labelled 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. These incorrectly extracted entries will be assigned a wrong attribute label. Liu, Grossman, and Zhai [5] proposed MDR, a method to mine data records in a Web page automatically. A *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,

Step 1: Build a HTML tag tree of the page.

Step 2: Mining data regions in the page using the tag tree and string comparison.

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

Consider a 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.

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 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.

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 a 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.

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 unseen attributes such as ISBN, publisher etc as shown in fig 2.

.

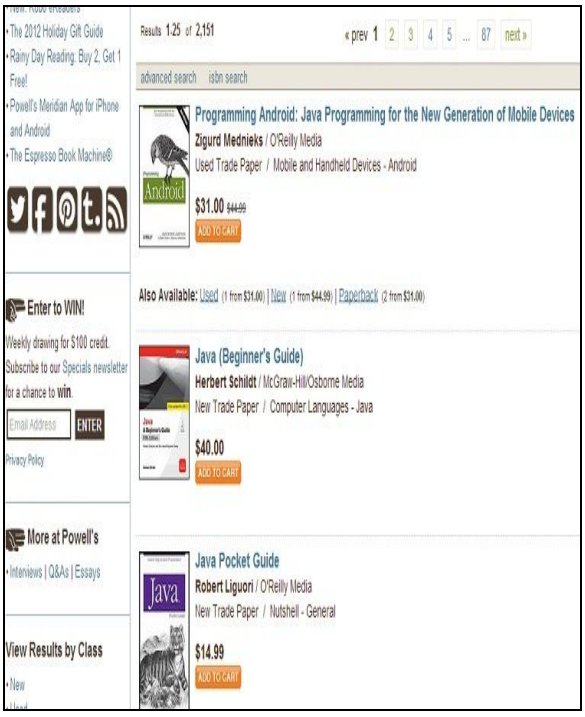


Fig 1: Sample Web page

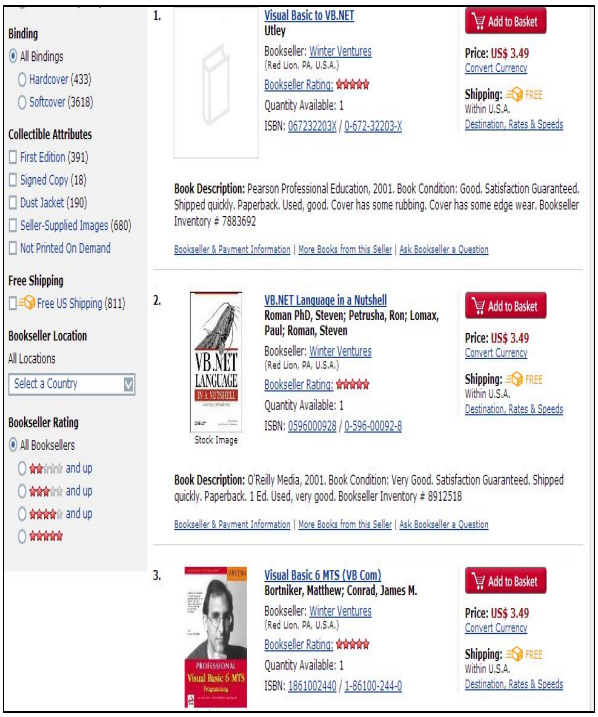


Fig 2: Sample Web page

IV. PROPOSED SYSTEM

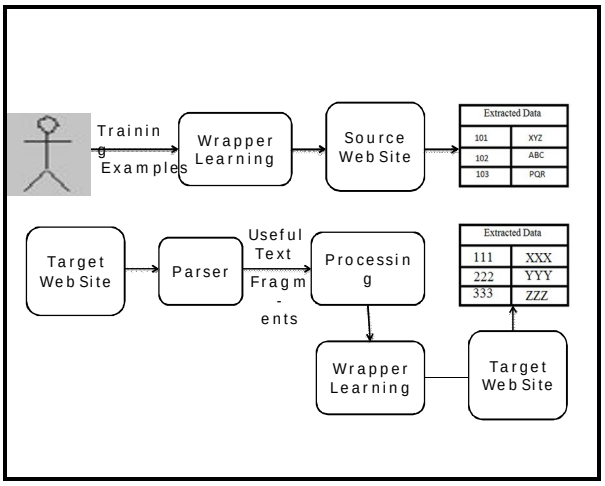


Fig 3: "Automatic wrapper adaptation system"

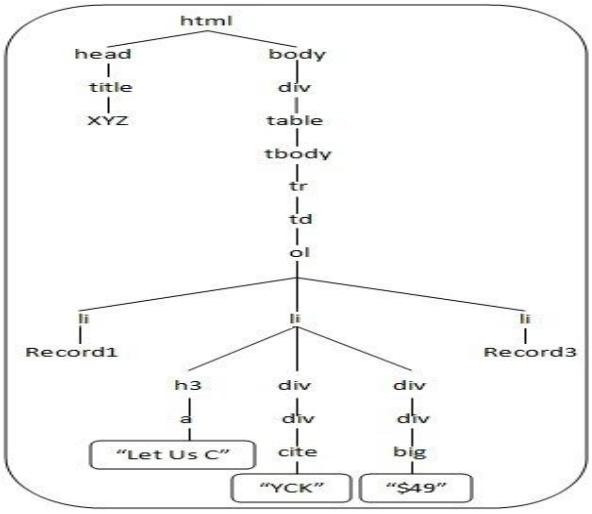


Fig 4: Sample tag tree of a record.

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.

Steps:

i. 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 "java" and select second page which contains all records of "C programming". These web pages are used as training set for the wrapper.

ii. 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 are likely to be similar in both the pages. In DOM tree representation all anchor tags are considered from both the web pages of same site. 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 all the anchor tags which have same contents on both web pages. Remaining are the useful text fragments. 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.

iii. 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 "java" books. Each record will contain "java" word in the attribute title. Get the word which has maximum frequency count value. In our case "java" will be the most frequent word.

iv. Locate the path:

Word with maximum frequency will give you the attribute title. Anchor tag of title of a book will be considered. This

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 Rs. 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. Tags are identified first and then root to leaf path in DOM tree is found for that attribute. For example from fig. x we can locate the paths from root to leaf for attribute title, author and price.

Title : a h3 li ol td tr tbody table div body html
Author : cite div div li ol td tr tbody table div body html
Price : big div div li ol td tr tbody table div body html

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

<html>
<body>

<table>
<tbody>
< tr >
< td >
< ol >
< 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 depicts the Web sites used in our experiment. B1,B2,B,3B4 are from book domain and E1,E2,E3,E4 are from electronic appliances domain.

	Web site (URL)
B1	Powell's Books (www.powells.com)
B2	Abebooks (www.abebooks.com)
B3	Amazon (www.amazon.com)
B4	Rediff (www.rediff.com)
E1	ebay (www.ebay.com)
E2	Flipkart (www.flipkart.com)
E3	Shoptronics (www.shoptronics.in)
E4	Homeshop18 (www.homeshop18.com)

Table 1: List of web sites

To compare the results we have used the tool- Automation anywhere 6.6. 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 can be adapted to other sites. Our wrapper adaptation approach achieves better performance compared with Automation anywhere. Table 2 and Table 3 shows the comparison of results for book domain and electronic appliances domain respectively. 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.

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 unseen 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 Web

sites in different domains were conducted and the results demonstrate that our method achieves a very promising performance.

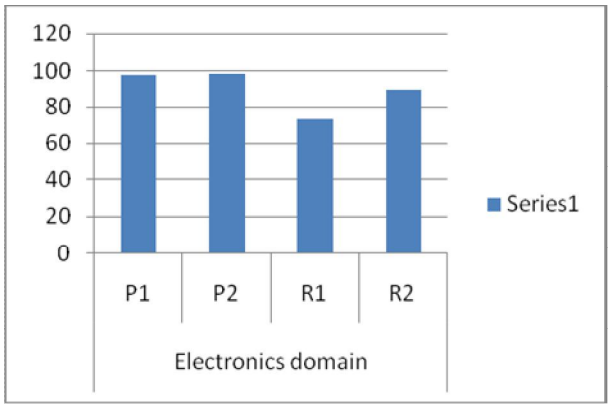
Website	Automation Anywhere		Our approach	
	P(%)	R(%)	P(%)	R(%)
B1	99.3	91.0	99.3	85.2
B2	88.8	79.0	98.0	90.0
B3	70.0	76.0	90.0	100.0
B4	91.6	100.0	93.2	100.0

Table 2: Extraction performance for book domain

Website	Automation Anywhere		Our approach	
	P(%)	R(%)	P(%)	R(%)
E1	95.9	90.7	100.0	92.5
E2	100.0	75.0	100.0	98.5
E3	97.0	62.5	98.3	99.0
E4	98.7	66.6	96.3	66.6

Table 3: Extraction performance for electronics appliances domain

Graph 1



Graph 2

REFERENCES

[1] Tak-Lam Wong and Wai Lam, "Learning to Adapt Web Information Extraction Knowledge and Discovering New Attributes via a Bayesian Approach", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 4, APRIL 2010.

[2] W. Cohen and W. Fan, "Learning Page-Independent Heuristics forExtracting 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-BasedData 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, "Populating the Semantic Web," Proc. AAAI Workshop Advances in Text Extraction and Mining, 2004.

[5] B. Liu, R. Grossman, and Y. Zhai, "Mining Data Records in Web Pages," Proc. Ninth ACM SIGKDD, pp. 601-606, 2003.

[6] W.Y. Lin and W. Lam, "Learning to Extract Hierarchical Information from Semi-Structured Documents," Proc. Ninth Int'l Conf. Information and Knowledge Management (CIKM), pp. 250-257, 2000.

[7] T.L. Wong and W. Lam, "Adapting Information Extraction Knowledge for Unseen Web Sites," Proc. IEEE Int'l Conf. Data Mining (ICDM), pp. 506-513, 2002.

[8] T.L. Wong and W. Lam, "Text Mining from Site Invariant and Dependent Features for Information Extraction Knowledge Adaptation," Proc. SIAM Int'l Conf. Data Mining (SDM), pp. 45-56, 2004.

[9] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and H.-W. Hon, "Webpage Understanding: An Integrated Approach," Proc. 13th ACM SIGKDD, pp. 903-912, 2007.