ISSN:0975-9646

A Survey on Unsupervised Extraction of Product Information from Semi-Structured Sources

Abhilasha Bhagat ,
ME Computer Engineering,
G.H.R.I.E.T., Savitribai Phule University, pune
PUNE , India

Vanita Raut
Assistant Professor Dept. of Computer Engineering
G.H.R.I.E.T., Savitribai Phule University, pune
PUNE , India

***Abstract*— Now a days, searching Product information has become one of the most important application areas of the Internet. However, the large amount of data is available about the products on web and its various representations may easily overstrain potential customers. Online product information is generally semi-structured format because it is represented through template-generated HTML pages. Such pages only follow an implicit data schema since information and presentation part are mixed up. Goal of the information extraction task is to access the product information in a structured manner efficiently. The proposed technique is based on a clustering approach that uses structural and visual features of web page elements. The information which has been extracted allow user to effectively compare products while saving the manual extraction work. Here, in this survey paper we have described various wrapper generation system such as RoadRunner, SG-WRAP, X-WRAP, DeLA in detail with their advantages and limitation.**

Keywords: **Information Extraction(IE), Unsupervised Extraction, Wrapper, Wrapper Induction**

I. INTRODUCTION

In this Internet world ,people have been more attracted towards an online shopping, an enormous amount of online malls emerged during the last years. Producers and various third-parties person try to presenting their products online in an appealing and more informative way. In this way the WWW has become the modern day's most important product information and shopping facility. However, the variety of product information sources is getting increasingly and it is very much difficult to overlook for a single customer and hence requires the integration of such sources information at single place .

When including information from online malls, these systems are generally able to query available Web Services (e.g., Amazon) or get the offers by feed-like mechanisms (e.g.,Buy.com). For including actual product specifications requires more manual work ,for example locate the producer's website, find the product information detail page, and extract the information about the product specifications.

The extraordinary growth of the Internet and World Wide Web has been fueled by the ability it gives content providers to easily and cheaply publish and distribute electronic documents. Companies create web sites to make available their online catalogs, annual reports, marketing brochures, product specification. Government agencies create web sites to publish new regulations, tax forms, and service information. Independent organizations create web sites to make available recent research results. Individuals create web sites dedicated to their professional interest and hobbies. This brings good news and bad news.

The good news is that the bulk of useful and valuable HTML-based Web information is designed and published for human browsing. This has been so successful that many Net businesses rely on advertisement as their main source of income, o ering free email services, for example. The bad news is that these \human-oriented" HTML pages are dificult for programs to parse and capture. Furthermore, the rapid evolution of Web pages requires making corresponding changes in the programs accessing them. In addition, most of the web information sources are created and maintained autonomously, and each o ers services independently. Interoperability of the web information sources remains the next big challenge.

A popular approach to address these problems is to write wrappers to encapsulate the access to sources. For instance, the most recent generation of information mediator systems , all include a pre-wrapped set of web sources to be accessed via database-like queries. However, developing and maintaining wrappers by hand turned out to be labor intensive and error-prone.

II. PROPOSED IMPLEMENTATION

In Steps 1 to 4 , the specification page is retrieved through a web page analyzer and XPath queries as well as the spatial arrangement (coordinates, contained texts, visibility, etc.) of all elements are extracted.

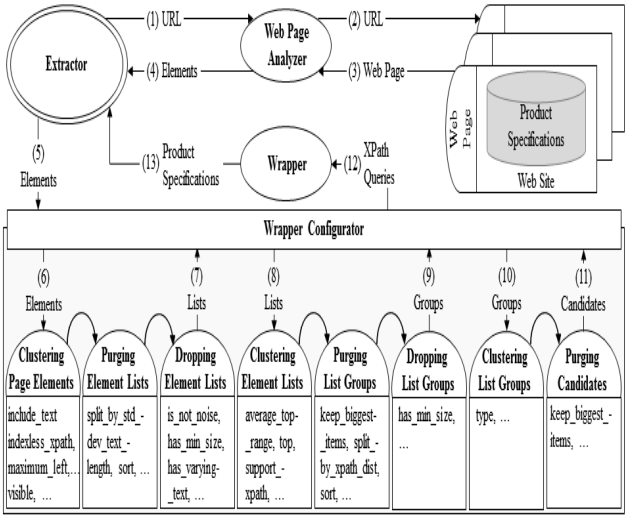

Fig: System Architecture of Proposed System

In Step 5, the information extracted in above series of steps is provided to a clustering coordinator .

In Steps 6 to 7, based on given features, it first clusters all elements into lists during the step 6 and 7. Some of example features that are given in the above figure.

In steps 8 to 9, element lists are clustered into groups. In the best case, one of the groups contains the elements list with the product specification keys and the elements list with the values (G2). Again, created groups are purged to drop insignificant ones (G1).

In Steps 10 to 11, consist of creating extraction candidates. Based on the best-rated candidate (C1), the wrapper configurator can finally create a wrapper consisting of a set of XPath queries (step 12) and provide it to the extractor (step 13). The extractor executes the wrapper and delivers the actual product specifications.

III. LITERATURE SURVEY

A.RoadRunner: Towards Automatic Data Extraction from Large Web Sites.

I.Introduction

In this proposed work [2], author have described techniques for extracting data from HTML sites through the use of automatically generated wrappers. For automatic wrapper generation and the data extraction process, author have developed a novel technique RoadRunner ,it compares the HTML pages and generate a wrapper based on their similarities and differences.

On Internet the large amount of information is available in HTML format and it grows at a very fast rate,thats why we can consider that the Web as the biggest "knowledge base" which is publicaly available to the user.

Data extraction from HTML pages is usually performed by software modules called wrappers. During Early days , manual techniques have been used for wrapping Web sites .But the key problem with manually coded wrappers is that writing the wrapper is usually a difficult and labor intensive task, and it is also difficult to maintain.

II.Advantages and Limitationof RoadRunner

1. RoadRunner does not depend on user-specified examples, it also does not require any interaction with the user during the wrapper generation process; this shows that wrappers are generated and data are extracted is totally automatic procedure.
2. In RoadRunner the wrapper generator has no a priori knowledge about the page contents.(For example schema of the HTML pages ,according to which data are organized)
3. RoadRunner is not restricted to flat records, it can also handle an arbitrarily nested structures.

III.Comparision With Other System:

1. Wien [3]and Stalker[4] generated their wrappers by examining a number of labeled examples, and therefore the systems had a precise knowledge about the target schema of HTML Pages. On the other hand RoadRunner did not have any a priori knowledge about the organization of the pages.
2. CPU time used by roadRunner to generate wrapper is lower than those needed to learn the wrapper both by Wien and Stalker.
3. RoadRunner can easily handle the optional fields in the schema. But, Wien is unable to handle optional fields.
4. Stalker[4] has more considerable expressive power since it can handle disjunctive patterns; On the other hand RoadRunner fails in this cases.
5. Wien and Stalker cannot handle nested structures, and therefore they fail on PharmaWeb. On the other hand RoadRunner correctly discovers the nested structure and automatically generates the wrapper.

B.SG-WRAP :A Supervised Visual Wrapper Generator for Web-Data Extraction.

I.Introduction

In this paper[5],Author have developed a schema-guided approach to generate wrapper . Here, in this work author have provided a user-friendly interface that allows users to define the schema of the data to be extracted, and specifies mappings from a HTML page to the target schema. Based on the mappings, the system can automatically generate an extraction rule which is used to extract the data from the page.

A schema-guided approach to wrapper generation can significantly reduce the work of human beings. In this case user never have to worry about the internal extraction rule, or even familiarity with the details of HTML. Based on the mappings, the SG-WRAP can automatically generate a wrapper to extract data from the page.

Experiments on real-world Web pages show that this approach to wrapper generation can significantly reduce the work of human beings in this process and get satisfactory precisions. Those users who are not familiar with wrappers he can also easily master the procedure of wrapper generator with SG-WRAP.

II.System Architecture

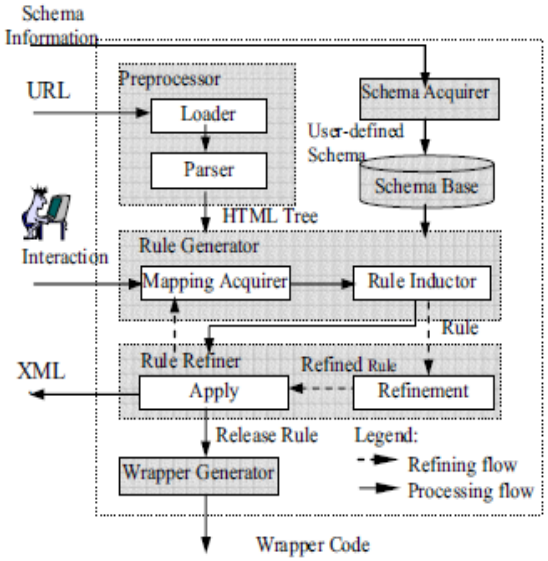

Fig: System Architecture of SG-WRAP

Above figure shows the SG-WRAP system architecture, it has five following major components

Preprocessor

It is responsible for setting up the environment for the system. It fetches the Web page using the URL given by

the user. The fetched HTML page is displayed before user in the browser for the next steps.

Schema Acquirer.

It is used to fetch the user-defined schema from the extracted data. A user defined schema can be saved in a Schema Base for later use, or shared for other sources from which the same kinds of data to be extracted.

Rule Generator

It generates data extraction rule by using an induction algorithm on the user assigned mappings between schema elements and HTML document data nodes,. Induction algorithm takes the list of mapping instances as input and it returns a candidate rule.

Rule Refiner

It generates an XML document by applying the induced rule on the input page. From the displayed document, a user determines whether the current extraction rule correctly extracts required data from the source page. If not the it can identify more mapping instances. The induction process will repeat and the Refiner will merge the previously generated rules with the refined rules. The system will display a new version of result to the user for checking. This refining process continues until the user is satisfied with the data which is extracted.

Wrapper Generator

Finally ,Wrapper Generator materializes the extraction rule into Java program and outputs it for repeatedly usages.

III. *Advantages and Limitation of SG-WRAP:*

1. SG-WRAP was developed to semi-automatically generate to extract well-formatted data from Web HTML pages. In contrast to extracting small locally well-formatted data, large text information elements in Web pages creates a several new problems.
2. This approach to wrapper generation can significantly reduce the work of human beings in this process and get satisfactory results.
3. What's more, ordinary users who are not familiar with wrappers , he can also easily master the procedure of wrapper generator with SG-WRAP.

C.XWRAP: An XML-enabled Wrapper Construction System for Web Information Sources

I.Introduction

XWRAP[6] is semi-automatic wrapper generation framwork . By using XWRAP we are able to extract the metadata about information content tand encoded explicitly as XML tags in the wrapped documents.

Features of XWRAP Wrapper Generation Framework :

1. XWRAP explicitly separates tasks of building wrappers that are specfic to a Web source from the tasks that are repetitive for any source, and it also uses a component library inorder to provide basic building blocks for wrapper programs.
2. XWRAP provides inductive learning algorithms that discover wrapper patterns by reasoning about sample pages or sample specifications.
3. XWRAP consist of two phase code generation framework, in which first phase utilizes an interactive interface facility to encode the source-specific metadata knowledge identified by individual wrapper developers as declarative information extraction rules. The second phase combines the information extraction rules that are generated at the first phase with the XWRAP component library to construct an executable wrapper program for the given web source.

II.System Architecture:

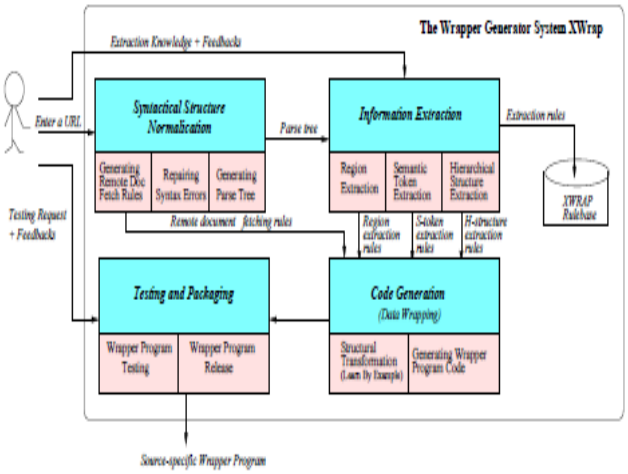

Fig: Wrapper Generation Process

The architecture of XWRAP for data wrapping consists of following four components .

Syntactical Structure Normalization

ISyntactical Normalizer, which prepares and sets up the environment for information extraction process by performing the following three tasks.

1.The syntactical normalizer accepts an URL selected and entered by the XWRAP user, issues an HTTP request to the remote server identified by the given URL, and fetches the corresponding web document . This page object is used as a sample for XWRAP to interact with the user to learn and derive the important information extraction rules.

2.It cleans up bad HTML tags and syntactical errors.

3.It transforms the retrieved page object into a parse tree or so-called syntactic token tree.

Information Extraction

It is responsible for deriving extraction rules that use declarative specification to describe how to extract information content of interest from its HTML formatting.

XWRAP performs following 3 steps inorder to extract the information from HTML pages.

(1) Identification of interesting regions in the retrieved document,

(2) Identification of the important semantic tokens and their logical paths and node positions in the parse tree.

(3) Identification of the useful hierarchical structures of the retrieved document.

Code Generation

It generates the wrapper program code through applying the three sets of information extraction ruls produced in the

second step.Here the key technique of this implementation is the smart encoding of the semantic knowledge which is represented in the form of declarative extraction rules and XML-template format .

The code generator interpret the XML-template rules by linking each executable components with each type of rules.

Testing and Packing

The toolkit user may enter a set of alternative URLs of the same web source to debug the wrapper program generated by running the XWRAP automated testing module. For each URL entered for testing purpose, the testing module will automatically go through the syntactic structure normalization and information extraction steps to check if new extraction rules or updates to the existing extraction rules are derived. In addition, the test-monitoring window will pop up to allow the user to browse the test report. Once the user is satisfied with the test results, user can packa the wrapper program with application plug-ins and user manual into a compressed tar file.

III.Advantages And Limitation:

1.XWRAP [6] provides a user-friendly interface program to allow users to generate their information extraction rules with a simple way.
2.It provides a clean separation of the information extraction semantics from the generation of procedural wrapper programs (e.g., Java code). Such separation allows new extraction rules to be incorporated into a wrapper program incrementally.
3.It facilitates the use of the micro-feedback approach to revisit and tune the wrapper programs at run time.

D. DELA: Data Extraction and Label Assignment for Web Databases.

I.Introduction

In this paper[7], Author have described the problem of automatically extracting data objects from a given web site and assigning meaningful labels to the data. In order to solve this problem, the data to be extracted from such web sites and its schema to be captured, which makes it easier to do further manipulation and integration of the data.

This problem is very challenging for following reasons.

1.The system needs to deal with HTML search forms, which are basically designed for human use. This makes it difficult for programs to identify all the form elements and submit correct queries.
2.The wrapper generated for each web site needs to be complex enough to extract not only plain-structured data, but also nested-structured data.
3.The generated wrapper is usually based on the structure of the HTML tags, which may not reflect the real database structure, and the original database field names are generally not encoded in the web pages.

In this work [6],author have described the ***DeLa*** (**D**ata **E**xtraction and **L**abel **A**ssignment) system that sends queries through HTML forms, automatically extracts data objects from the retrieved web pages, and finally fits the extracted data into a table and assigns labels to the attributes of the data objects, i.e., the columns of the table.

II. System Architecture

Form Crawler.

Given a web site with a HTML search form, the form crawler collects the labels of each element contained in the form and sends queries though the form elements to obtain the result pages containing data objects. We adopt the hidden web crawler, Hiwe for this task.

Wrapper Generation.

The pages collected by the form crawler are output to the wrapper generator to induce the regular expression wrapper based on the pages' HTML-tag structures. Since pre-defined templates generate the web pages, the HTML tag-structure enclosing data objects may appear repeatedly if the page contains more than one instance of a data object. Therefore, the wrapper generator first considers the web page as a token sequence composed of HTML tags and a special token "text" representing any text string enclosed by pairs of HTML-tags, then extracts repeated HTML tag substrings from the token sequence and induces a regular expression wrapper from the repeated substrings according to some hierarchical relationships among them.

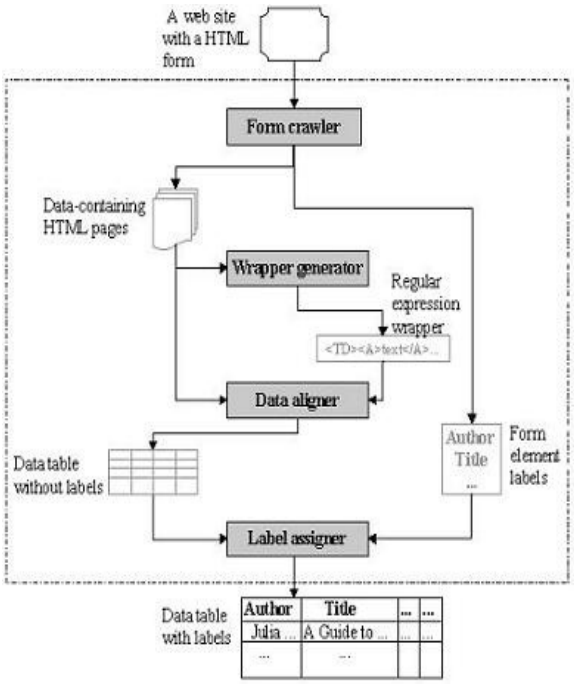

Fig: System Architecture of DeLA.

Data Aligner.

Given the induced wrapper and the web pages, the data aligner first extracts data objects from the pages by matching the wrapper with the token sequence of each page. It then filters out the HTML tags and rearranges the data instances into a table similar to the table defined in a relational DBMS, where rows represent data instances and columns represent attributes.

Label Assigner.

The label assigner is responsible for assigning labels to the data table by matching the form labels obtained by the form crawler to the columns of the table. The basic idea is that the query word submitted through the form elements will probably reappear in the corresponding fields of the data

IV. CONCLUSION

The focus had been on the effectiveness which allowed the coverage of many different product page templates. From all three page representation formats, the visual representation provided the most valuable clustering features. It is therefore indispensable to employ such information. The quite high costs in terms of time are acceptable since the process might run as a steady background task and may even be parallelized for several products, e.g., through a customized MapReduce algorithm. The technique of information extraction can be applied to non-HTML documents such as medical records and curriculum vitae to facilitate the maintenance of large semistructured documents. In the future, information extraction from cross-Web site pages will become more important as we move toward semantic Web.

The trend of developing highly automatic Information Extraction systems, which saves not only the effort for programming, but also the effort for labeling. As we know the Web services provides way for data exchange and information integration, but it may not be the best choice since the involvement of programmer is unavoidable.

REFERENCES

[1]. Maximilian Walther, "Unsupervised Extraction of Product Information from Semi-structured Sources", CINTI 2012 13th IEEE International Symposium on Computational Intelligence and Informatics, 20–22 November, 2012 , Budapest, Hungary.

[2] V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: Towards Automatic Data Extraction from Large Web Sites," in VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 109–118.

[3] N. Kushmerick, D. Weld, and R. Doorenbos, "Wrapper Induction for Information Extraction," Proc. 15th Int'l Conf. Artificial Intelligence (IJCAI), pp. 729-735, 1997.

[4] I. Muslea, S. Minton, and C. Knoblock, "A Hierarchical Approach to Wrapper Induction," Proc. Third Int'l Conf. Autonomous Agents (AA '99), 1999.

[5] C.-H. Chang, C Meng, X., Lu, H., Wang, H., and Gu, M. Schema-Guided Wrapper Generator. ICDE-02, 2002.

[6]. L. Liu, C. Pu, and W. Han, "XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources," Proc. 16Th IEEE Int'l Conf. Data Eng. (ICDE), pp. 611-621, 2000.

[7] J. Wang and F.H. Lochovsky, "Data Extraction and Label Assignment for Web Databases," Proc. 12th Int'l Conf. World Wide Web (WWW), pp. 187-196, 2003.

[8] C.-H. Chang and S.-C. Lui, "IEPAD: Information Extraction Based on Pattern Discovery," in WWW '01: Proceedings of the 10th International Conference on World Wide Web. New York, NY, USA: ACM, 2001, pp. 681–688.

[9] K. Simon and G. Lausen, "ViPER: Augmenting Automatic Information Extraction with Visual Perceptions," in CIKM '05: Proceedings of the 14th International Conference on Information and Knowledge Management. New York, NY, USA: ACM, 2005, pp. 381–388.