Volume 5, Issue 5, May 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Survey on Semi-Supervised and Supervised Web Information Extraction Techniques

Aleem Ansari, Hemlata Vasishtha
Shri Venkateshwara University,
Andhra Pradesh India

Abstract— The World Wide Web is perhaps the largest repository of information. There is a huge need for making use of the publicly available information for providing value added services such as comparative shopping, market intelligence, meta-querying and search. Since web pages are formatted for visual appearance and not for data extraction, they cannot be queried like relational data. Hence there is a great need for Information Extraction (IE) from such Web Pages. There has been extensive research in the field of Information Extraction from Web Pages and many tools have been developed till date. In this paper we categorize the Web Information Extraction approaches into four categories: Manual, Supervised, Semi-supervised and Unsupervised. This paper presents the challenges, prominent techniques, tools and progress made in this area.

Keywords— Information Extraction, Wrapper Generation, Web Content Mining, Semi-Structured Data, Data Record Extraction, Information Retrieval

I. INTRODUCTION

The World Wide Web is perhaps the largest repository of information. There is a huge need for making use of the publicly available information for providing value added services such as comparative shopping, market intelligence, meta-querying and search. Since web pages are formatted for visual appearance and not for data extraction, they cannot be queried like relational data. Hence there is a great need for Information Extraction (IE) from such Web Pages. There has been extensive research in the field of Information Extraction from Web Pages and many tools have been developed till date. Other areas of work are also related to this study such as ontology learning, data integration and web page cleaning [1].In this paper we categorize the Web Information Extraction approaches into four categories: Manual, Supervised, Semi-supervised and Unsupervised. This paper presents the challenges, prominent techniques, tools and progress made in this area.

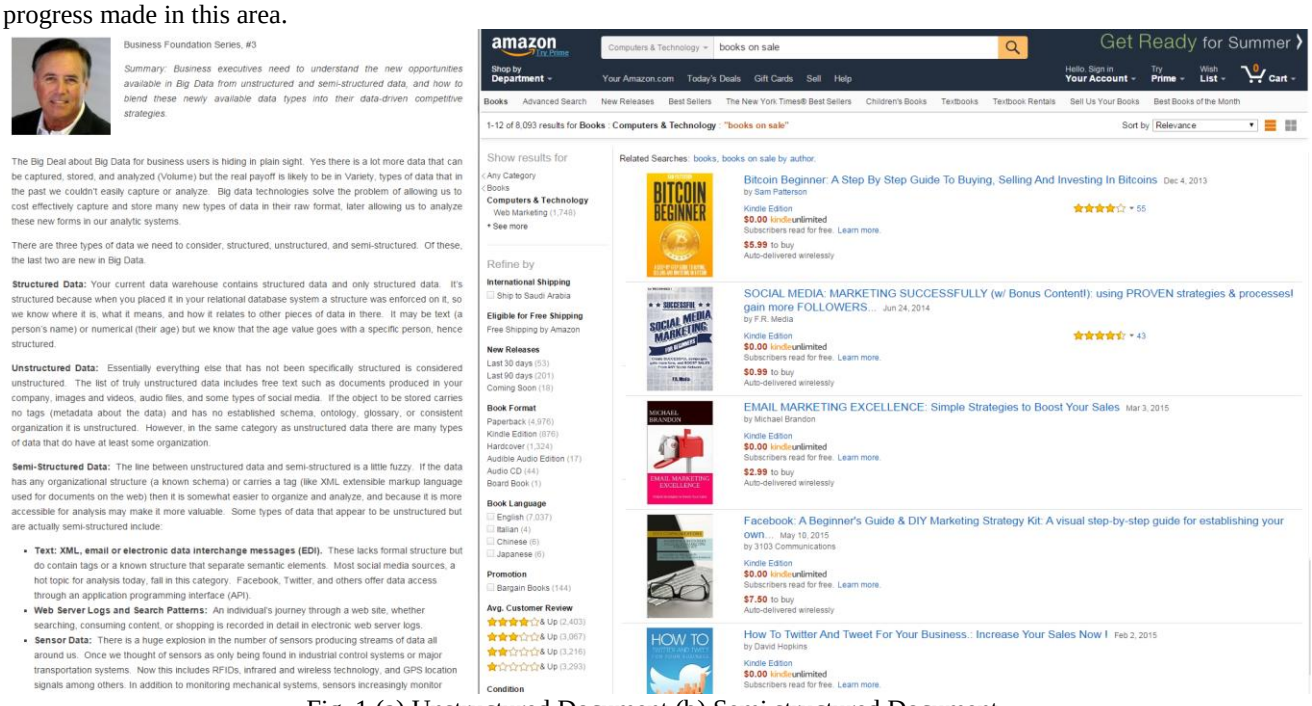

Fig. 1 (a) Unstructured Document (b) Semi structured Document

Formally an Information Extraction task is defined by its input and its extraction target [2]. The input can be unstructured documents such as text (e.g. Figure 1 (a)), word processor document, audio, video, images or semi-structured document such as tables or lists (e.g. Figure 1 (b)). The extraction target of an Information Extraction task can

be defined as a relation of k-tuple (where k is the number of attributes in a record) or it can be complex object with hierarchically organized data [2]. An attribute in a record may contain single value, multiple values or no value at all. Further these attributes can occur in any order in the input documents.

The data in semi-structured input pages may be presented in HTML or non-HTML format [2]. Semi-structured Web pages are usually generated from structured databases with predefined templates or layouts. For example, the set of book pages from Amazon (as shown in Figure 1 (b)) has the same layout for the title, authors, price, rating, etc. Web pages generated from the same database with same template (and hence program) form a page class [2]. For Information Extraction task, the input pages can be of the same page class or they can be heterogeneous pages from different Web sites.

Since Web pages are specifically formatted for visual appearances, they incur several challenges in extracting information from them. These challenges include but are not limited to: lack of a well defined schema, poor formatting, high update frequency and semantic heterogeneity of the information [3]. The Web Information Extraction approaches can be classified into Manual Approaches, Supervised Approaches, Semi Supervised Approaches, and Unsupervised Approaches.

II. MANUAL APPROACHES

Manual Information Extraction systems make use of wrapper generation and wrapper induction techniques. In wrapper generation, user has to manually write an extraction program for each web site based on observed format patterns using general programming languages. The problem of wrapper generation can be described as follows. Given a web page P containing a set of implicit objects, determine a mapping M that populates data repository R with the objects in P. The mapping W must also be capable of recognizing and extracting data from any other pages P1 similar to P. In this context a wrapper is a program that executes the mapping M [4]. Since even small change in the web site may prevent the wrappers from working properly, and the template or layout of web pages are often subject to change, maintaining those wrappers is expensive and inefficient [5]. Hence these systems do not scale to large number of web sites. Such systems include Minerva [6], TSIMMIS [7], Web-OQL [8], etc.

III. SUPERVISED APPROACHES

The Supervised Approach can be classified into two types: sequenced based and tree based [9]. The former, such as WIEN [10], Soft-Mealy [11], and Stalker [12] represents input documents as sequences of tokens or characters, and generates delimiter based extraction rules through a set of training examples. The latter, such as W4F [13] and XWrap [14] parses input document into hierarchical tree (DOM tree) structure, based on which they perform the extraction process. In these systems, the user first manually labels a set of training pages as positive and negative examples. The learning system then generates rules from the training pages. The resulting rules are then applied to extract target items from related web pages [9]. These approaches require prior syntactic knowledge and substantial manual efforts for labelling sample pages, which is labour-intensive and time-consuming. Additionally, for different sites or even pages from the same site, the manual labelling process needs to be repeated because they follow different templates/patterns.

In order to improve the efficiency and reduce manual efforts, most recent researches focus on Semi-supervised and Unsupervised approach instead of Manual or Supervised ones.

IV. SEMI-SUPERVISED APPROACHES

A. IEPAD

IEPAD (Information Extraction based on PAttern Discovery) [15] discovers extraction patterns from unlabeled Web pages. This method is based on the assumption that if a Web page contains multiple (homogeneous) relevant data records, they are often rendered regularly using the same template. Thus, repetitive patterns can be discovered if the pages are well encoded. Therefore, wrappers can be solved by discovering repetitive patterns. IEPAD uses PAT trees [16] to discover repetitive patterns in the Web page. Since PAT trees only records the exact match for suffixes, it further applies center star algorithm to align multiple strings which start from each occurrence of a repeat and end before the start of next occurrence. Finally, a signature representation is used to denote the template to comprehend all data records.

The method used by IEPAD is probable to generate incorrect patterns along with the correct ones, so human post-processing of the output is required [17]. Further this technique is applicable for structured data only.

B. OLERA

OLERA (On-Line Extraction Rule Analysis) [18] is a semi-supervised IE system that acquires rough examples from the user for generating extraction rule. OLERA can learn extraction rules for pages containing single data records. Figure 2 (a) show a sample web page submitted to OLERA. Here the user highlights a sample record for training. The corresponding result is shown in figure 2 (b).

OLERA consists of three main operations:

- Enclosing an information block of interest: where the user marks an information block containing record to be extracted for OLERA to discover other similar blocks (using approximate matching technique) and generalize them to an extraction pattern (using multiple string alignment technique)
- Drilling-down/rollingup an information slot: drilling-down allows the user to navigate from a text fragment to more detailed components, whereas rolling-up combines several slots to form a meaningful information unit.
- Designating relevant information slots for schema specification as in IEPAD.

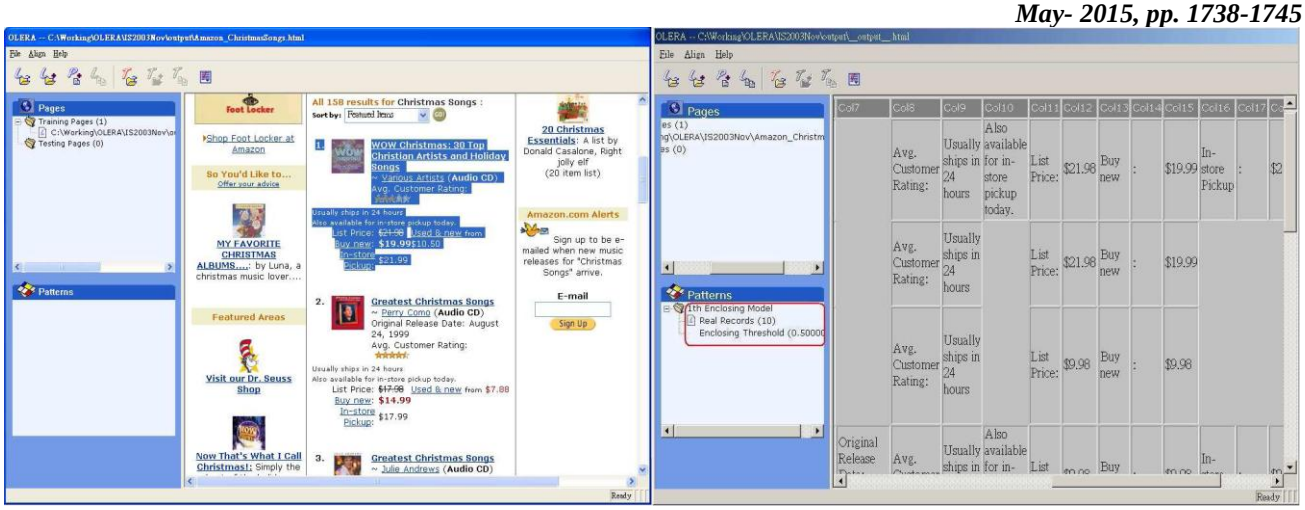

Fig. 2 (a) Enclosing one record (highlighted area) for analysis in OLERA (b) Extracted records displayed in a spreadsheet [18].

C. Thresher

Thresher [19] uses semi-supervised approach similar to OLERA. Thresher takes the example from the user for extracting data about a particular object, and uses it to reverse engineer the templating engine, for extracting corresponding information from the web pages. Thresher allows users to describe examples of semantic patterns (or wrappers) simply by highlighting and marking up relevant content of the web page in the Haystack browser. Figure 3 shows snapshot of user interaction with the Haystack browser on the MIT CSAIL faculty page taken from [19]. Here the user is creating a wrapper to match faculty information.

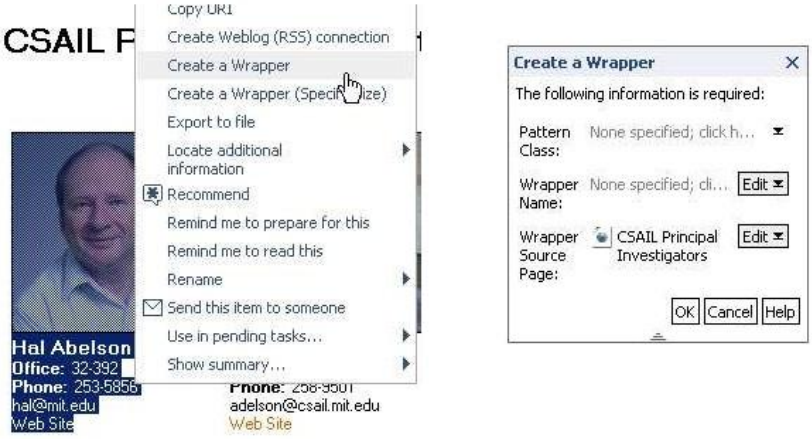

Fig. 3 Creating a wrapper on the CSAIL Faculty page using Thresher Technique [19].

From these positive examples, thresher induces required pattern by applying tree edit distance metric to find the best mapping between the example's and a target's document's HTML parse trees. To get feedback from the user, matches are displayed directly on the browser. Once the wrapper is created, user can give it semantic meaning by indicating the class (type of object) extracted by the wrapper, as well as marking up attributes of that object.

V. UNSUPERVISED APPROACHES

Data objects on the web pages are normally database records retrieved from underlying web databases and displayed in the web pages with some fixed templates. Unsupervised approaches aim to find patterns/grammars from the web pages for extracting data. In some cases, several schemas may comply with the training pages due to the presence of nullable data attributes, leading to ambiguity [20]. The choice of determining the right schema is left to users. Similarly, if not all data is needed, post-processing may be required for the user to select relevant data and give proper naming (attribute name) to each piece of data.

A. Road Runner

RoadRunner [21] is based on an unsupervised learning algorithm. Its goal is to automatically extract data from Web sources by exploiting similarities in page structure across multiple pages. It generates wrapper by comparing two pages, using the ACME (Align, Collapse under Mismatch, and Extract) technique to align matched tokens and collapse for mismatched tokens. There are two kinds of mismatches: string mismatches and tag mismatches. String mismatches indicates attributes whereas tag mismatches indicates iterator(s) or optional attribute(s). Since there can be several alignments, RoadRunner uses UFRE (union-free regular expression) to reduce the complexity [22]. The alignment result of the first two pages is then compared to the third page in the page class.

Advantages of the Road Runner are: it does not require any interaction with the user during the wrapper generation process, it has no prior knowledge about the schema of the web page, it is not restricted to the flat records, and it can handle nested structures also [23]. However the problem with this method is that it cannot deal with disjunctions in the input schema and it requires receiving as input multiple pages conforming to the same template.

B. NET

NET (Nested data Extraction using Tree matching and visual cues) [24] method is applicable to Web pages that contain a set of flat or nested data records. It works in two stages. First stage is building a tag tree of the page and second stage is identifying data records and extracting data from them. This method uses tree edit distance algorithm and visual cues to identify data records at different levels. Benefits of NET technique come from accurate alignment and extraction of both flat and nested data records. However this technique assumes that the web page contains two or more data records.

C. EXALG

EXALG [25] takes as input set of pages created from unknown template T and the values to be extracted. EXALG then deduces the template T for extracting data from the input pages. EXALG use two techniques for detecting the unknown template T: differentiating roles and equivalence classes (EC). In differentiating roles technique, the occurrences with two different paths of a particular token have different roles. An equivalence class is a maximal set of tokens having the same occurrence frequencies over the training pages (occurrence-vector).The insight is template tokens that encompass a data tuple have the same occurrence vector and form an equivalence class. However, to avoid data tokens to accidentally form an equivalence class, ECs with insufficient support (the number of pages containing the tokens) and size (the number of tokens in an EC) are filtered out. In addition, to conform to the hierarchical structure of the data schema, equivalence classes must be mutually nested and the tokens in an EC must be ordered. Those valid ECs are then used to construct the original template.

EXALG makes some assumptions about web pages which do not hold in a significant number of cases: for instance, it is assumed that template assigns relatively large number of tokens to each type constructor [26]. It is also assumed that a substantial subset of the data fields to be extracted have a unique path from the root in the DOM tree of the pages. It also requires receiving as input multiple pages. Further this technique is applicable to pages with structured data only.

D. ViNTs

ViNTs (Visual information aNd Tag structure based wrapper generator) [27] uses both tag and visual features to deduce wrapper from a set of training page. It first utilizes the visual data value similarity without considering the tag structure to identify data value similarity regularities and then combines them with the HTML tag structure regularities to generate wrappers. Both visual and non visual features are used to weight the relevance of different extraction rules.

However ViNTs algorithm has several limitations. First it requires several input result pages, each of which must contain at least four Query Result Records (QRRs). It also requires users to input no-result page which may not exist for many web sites; since these sites may respond with closely similar records when no match is found for the query. If the data records are distributed over multiple data regions only the major data region is reported. Further it only focuses on data record extraction, without considering data item extraction.

E. VIDE

Vision-based Data Extractor (ViDE) [28] is used to extract structured records from deep Web pages. ViDE is primarily based on the visual features human users can capture on the deep Web pages. It also makes use of non-visual information such as data types and frequent symbols to make the solution more robust. ViDE consists of two main components, Vision based Data Record extractor (ViDRE) and Vision-based Data Item extractor (ViDIE).

ViDE employs a four-step strategy. First, given a sample deep Web page from a Web database, obtain its visual representation and transform it into a Visual Block tree; second, extract data records from the Visual Block tree; third, partition extracted data records into data items and align the data items of the same semantic together; and fourth, generate visual wrappers (a set of visual extraction rules) for the Web database based on sample deep Web pages such that both data record extraction and data item extraction for new deep Web pages that are from the same Web database can be carried out more efficiently using the visual wrappers.

One of the limitations of ViDE is that its functioning is dependent on the browser and it work for structured data only.

F. Álvarez Contribution

Álvarez [29] presented a technique for extracting list of structured records from single web page. This method detects list of records by finding the node with maximum repetitive path patterns from the root to its leaf nodes in the DOM tree representation for the web page. Then it uses string alignment and edit-distance similarity algorithm for separating the list into individual records. However this method requires web pages to contain multiple records and all the records must be contiguous [5]. Hence this technique is not applicable to pages containing single data record.

G. ClustVX

ClustVX [30] is based on two fundamental observations. First, data records on the Web Pages are presented using similar template from the underlying databases. Hence each Data Record has almost the same Xpath (tag path from root

node in DOM tree to particular web page element). Second, although underlying data differ from page to page, humans easily understand them by analysing repeating visual patterns. In short the data with same semantic meaning are displayed using the same style. ClustVX encode both Xpath and visual features (such as font, color, etc) combination into the string called Xstring. By clustering Xstrings it identifies visually similar elements, which are located in the same region of a web page and in turn have same semantic meaning.

H. MDR

MDR (Mining Data Record) [31] assumes that most of data records are formed by table and form related tags i.e. <TABLE>, <FORM>, <TR>, <TD>, etc. The algorithm is based on two assumptions:

- A group of data records are always presented in a contiguous region of the web page and are formatted using similar HTML tags. Such region is called a Data Region.
- The nested structure of the HTML tags in the web page usually forms a tag tree and a set of similar data records are formed by some child sub-trees of the same parent node.

The algorithm works in three steps:

- The DOM tree of the input document is built.
- It then finds subsets of nodes that fulfil the following conditions: they are siblings, they are adjacent, they have the same number of children, and the edit distance amongst them does not exceed a predefined threshold. The idea is to detect regions that contain repetitive similar structures. The edit distance is calculated on the strings that result from serializing the nodes to be compared as strings; this serialization does not take text nodes into account, only HTML tags. The subsets of generalized nodes that result from the previous step are considered as data regions since each data region is supposed to contain two or more data records that have similar structures.
- The algorithm then separates the data records inside the previous data regions using the following heuristics: If the region consists of only one generalized node, it then checks if this node is not a table row, but all of its children are similar; if the condition is met, then the children are returned as independent data records; otherwise the generalized node itself is returned as a data record. ii) If the generalized node contains two or more nodes with the same number of children and these children are similar to each other, then it means that they are non-contiguous data records, i.e., the data region is an HTML table in which each data record is formatted in columns and not in rows; otherwise, the whole generalized node is returned.

The limitation of MDR is that it is html tag dependent and it does not align the data item. MDR fails in documents that have large menus, long listings of user comments, or documents in which the relevant information is rendered using lists, divisors, or other HTML constructs.

I. DEPTA

DEPTA [32] (Data Extraction based on Partial Tree Alignment), also known as MDR-2, is similar to MDR in its data region identification mechanism. DEPTA uses visual information (locations on the screen at which the tags are rendered) to build the HTML tag tree, for eliminating problems caused by HTML tag structure irregularities. It uses tree edit distance for identifying similar data records. A single data record may be composed of multiple subtrees due to noisy information. DEPTA makes use of the visual gaps between data records for finding right subtrees. However it requires content analysis to identify the main data region from the set of data regions obtained. Further, it also relies on TABLE tags for identifying data regions.

Specifically, this method also uses visual cues to find data records. Since DEPTA is tag dependent, it requires considerable time in building tag tree, traversing whole tag tree and string comparison. DEPTA assumes that the same number of sub-trees must form all records and the visual gap between two data records in a list is bigger than the gap between any two data values from the same record. These assumptions do not hold in all web sources. DEPTA only detects data records using tree regularities without considering semantics.

J. Other Approaches

HCRF [33] is a probabilistic model for both data record extraction and attribute labelling. Comparatively HCRF also uses VIPS algorithm [34] to represent Web pages, but the tag information is still an important feature in HCRF. HRCF is based on the assumption that every record corresponds to one block in the Visual Block tree, but this assumption is not always correct. VENTex [35] implements the information extraction from Web tables based on a variation of the CSS2 visual box model. So, it can be regarded as the only related work using pure visual features. However VENTex only aims to extract information from various forms of tables that are embedded in common pages.

TABLE I SUMMARY OF QUANTITATIVE ANALYSIS

Technique / Tool	**Automation Degree**	**Page Type**	**Extraction Level**	**Extraction Rule**	**Learning Algorithm**	**Limitation**
IEPAD	Semi-Supervised	Template Page	Record Level	Regular Expression	Pattern Mining, String Alignment	Multiple-records page
OLERA	Semi-	Template	Record	Regular	String Alignment	Not Restricted

	Supervised	Page	Level	Expression		to Page
RoadRunner	Unsupervised	Template Page	Page Level	Regular Expression	String Alignment	More than one page
EXALG	Unsupervised	Template Page	Page Level	Regular Expression	Equivalent Class and Role Differentiation by DOM tree path	More than one page
DEPTA	Unsupervised	Template Page	Record Level	Tag Tree	Pattern Mining, String comparison, Partial tree alignment	Multiple-records page

VI. WEB DATA EXTRACTION TOOLS

A. DEiXTo

DEiXTo [36] is based on the Document Object Model. It enables user to easily create extraction rules pointing out the portion of data to scrape from the Web site.

B. OXPath

OXPath [37] is a declarative formalism that extends XPath to support deep Web navigation and data extraction from interactive Web sites. OXPath is able to (1) fill web forms and trigger DOM events, (2) access dynamically computed CSS attributes, (3) navigate between visible form fields, and (4) mark relevant information for extraction.

C. Automation Anywhere

Automation Anywhere [38] helps end user to automate data extraction without any programming. It can automatically login to websites, account for changes in the source website, extract that information and copy it to another application reliably in a format specified by the user. However it requires substantial human efforts for training and labelling of data.

D. Import.io

Import.io [39] is a platform which facilitates the conversion of semi-structured information in web pages into structured data. It offers real-time data retrieval through JSON REST-based and streaming APIs. Though data extraction is done automatically but it needs substantial user efforts for data labelling.

E. Cloud Scrape

CloudScrape [40] provides a sophisticated online web scraping platform. It facilitates browser-based scraping editor for creating and maintaining scraping robots which in turn are executed to collect online web data. However it involves substantial human efforts for labelling, page navigation, etc.

Similarly there are large numbers of available tools that are used for web information extraction such as Mozenda [41], Data Toolbar [42], TheWebMiner [43], etc.

VII. CONCLUSIONS

There are several points to make from this survey. First, we see the trend of developing highly automatic IE systems, which saves not only the effort for programming, but also the effort for labeling. Although the creation of Web services provides another way for data exchange and information integration, it may not be the best choice since the involvement of programmer is unavoidable. On the other hand, not all IE tasks can be wrapped by fully automatic IE systems.

Manual IE systems can be applied to all kinds of inputs as long as proper features are provided by the systems, though it depends on the programmers' techniques to compose the extraction rules

Semi-supervised and unsupervised IE systems can be applied only to template based pages since their success rely on the existence of the template. The extension of such systems to non-template page extraction tasks is very limited. In contrast, supervised approaches, although require annotations from users; extend well to non template pages if proper features are selected for extraction rules.

REFERENCES

[1] Ye, Shiren, and T-S. Chua. "Learning object models from semistructured web documents." Knowledge and Data Engineering, IEEE Transactions on, vol. 18, pp. 334-349, 2006.

[2] Chang, Chia Hui, et al. "A survey of web information extraction systems." Knowledge and Data Engineering, IEEE Transactions on vol. 18, pp. 1411-1428, 2006.

[3] Lam, Man I., Zhiguo Gong, and Maybin Muyeba. "A method for web information extraction." Progress in WWW Research and Development. Springer Berlin Heidelberg, pp. 383-394, 2008.

[4] Firat, Aykut, et al. "Information aggregation using the caméléon# web wrapper." E-Commerce and Web Technologies. Springer Berlin Heidelberg, pp.76-86, 2005.

[5] Zheng, Xiaoqing, Yiling Gu, and Yinsheng Li. "Data extraction from web pages based on structural-semantic entropy." Proceedings of the 21st international conference companion on World Wide Web. ACM, 2012.

[6] Crescenzi, Valter, and Giansalvatore Mecca. "Grammars have exceptions." Information Systems, vol 23, pp. 539-565, 1998.
[7] Hammer, Joachim, Jason McHugh, and Hector Garcia-Molina. "Semistructured Data: The TSIMMIS Experience.", 1997.
[8] Arocena, Gustavo O., and Alberto O. Mendelzon. "WebOQL: Restructuring documents, databases and Webs." Data Engineering, 1988. Proceedings., 14th International Conference on. IEEE, 1998.
[9] Varade, Mrudula, and Vimla Jethani. "I-ViDE: An Improved Vision-Based Approach for Deep Web Data Extraction.", 2014.
[10] Kushmerick, Nicholas. Wrapper induction for information extraction. Diss. University of Washington, 1997.
[11] Hsu, Chun-Nan, and Ming-Tzung Dung. "Generating finite-state transducers for semi-structured data extraction from the web." Information systems, vol. 23, pp. 521-538, 1998.
[12] Muslea, Ion, Steve Minton, and Craig Knoblock. "A hierarchical approach to wrapper induction." Proceedings of the third annual conference on Autonomous Agents. ACM, 1999.
[13] Sahuguet, Arnaud, and Fabien Azavant. "Building intelligent web applications using lightweight wrappers." Data & Knowledge Engineering, vol. 36, pp. 283-316, 2001.
[14] Liu, Ling, Calton Pu, and Wei Han. "XWRAP: An XML-enabled wrapper construction system for web information sources." Data Engineering, 2000. Proceedings. 16th International Conference on. IEEE, 2000.
[15] Chang, Chia-Hui, and Shao-Chen Lui. "IEPAD: information extraction based on pattern discovery." Proceedings of the 10th international conference on World Wide Web. ACM, 2001.
[16] Gonnet, Gaston H., Ricardo A. Baeza-Yates, and Tim Snider. "New Indices for Text: Pat Trees and Pat Arrays." pp. 66-82, 1992.
[17] Álvarez, Manuel, et al. "Finding and extracting data records from web pages." Journal of Signal Processing Systems, vol. 59, pp. 123-137, 2010.
[18] Chang, Chia-Hui, and Shih-Chien Kuo. "OLERA: Semisupervised web-data extraction with visual support." IEEE Intelligent systems, vol. 19, pp. 56-64, 2004.
[19] Hogue, Andrew, and David Karger. "Thresher: automating the unwrapping of semantic content from the World Wide Web." Proceedings of the 14th international conference on World Wide Web. ACM, 2005.
[20] Yang, Guizhen, I. V. Ramakrishnan, and Michael Kifer. "On the complexity of schema inference from web pages in the presence of nullable data attributes." Proceedings of the twelfth international conference on Information and knowledge management. ACM, 2003.
[21] Crescenzi, Valter, Giansalvatore Mecca, and Paolo Merialdo. "Roadrunner: Towards automatic data extraction from large web sites." VLDB. vol. 1, 2001.
[22] Bhalerao, Rushikesh Shantaram, Shital Aher, and Seema B. Siledar. "Page-Level Web Data Extraction From Dynamic Pages: Fivetech." 2013.
[23] Vidya.V. "A Survey of Web Data Extraction Techniques.", 2014.
[24] Liu, Bing, and Yanhong Zhai. "NET–a system for extracting web data from flat and nested data records." Web Information Systems Engineering–WISE 2005. Springer Berlin Heidelberg, pp. 487-495, 2005.
[25] Arasu, Arvind, and Hector Garcia-Molina. "Extracting structured data from web pages." Proceedings of the 2003 ACM SIGMOD international conference on Management of data. ACM, 2003.
[26] Dong, Yongquan, and Qingzhong Li. "A Robust Approach of Automatic Web Data Record Extraction." Journal of Computer Information Systems, vol 5, pp. 1757-1766, 2009.
[27] Zhao, Hongkun, et al. "Fully automatic wrapper generation for search engines." Proceedings of the 14th international conference on World Wide Web. ACM, 2005.
[28] Liu, Wei, Xiaofeng Meng, and Weiyi Meng. "Vide: A vision-based approach for deep web data extraction." Knowledge and Data Engineering, IEEE Transactions on, vol. 22, pp. 447-460, 2010.
[29] Álvarez, Manuel, et al. "Extracting lists of data records from semi-structured web pages." Data & Knowledge Engineering, vol. 64, pp. 491-509, 2008.
[30] Grigalis, Tomas. "Towards Automatic Structured Web Data Extraction System." DB&Local Proceedings. 2012.
[31] Liu, Bing, Robert Grossman, and Yanhong Zhai. "Mining data records in Web pages." Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2003.
[32] Zhai, Yanhong, and Bing Liu. "Web data extraction based on partial tree alignment." Proceedings of the 14th international conference on World Wide Web. ACM, 2005.
[33] Zhu, Jun, et al. "Simultaneous record detection and attribute labeling in web data extraction." Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2006.
[34] Cai, Deng, et al. "Extracting content structure for web pages based on visual representation." Web Technologies and Applications. Springer Berlin Heidelberg, pp. 406-417, 2003.
[35] Gatterbauer, Wolfgang, et al. "Towards domain-independent information extraction from web tables." Proceedings of the 16th international conference on World Wide Web. ACM, 2007.
[36] Kokkoras, Fotios, Konstantinos Ntonas, and Nick Bassiliades. "Deixto: A web data extraction suite." Proceedings of the 6th Balkan Conference in Informatics. ACM, 2013.
[37] Furche, Tim, et al. "Oxpath: A language for scalable, memory-efficient data extraction from web applications." Proceedings of the VLDB Endowment, vol. 4, pp. 1016-1027, 2011.

[38] www.automationanywhere.com
[39] www.import.io
[40] www.cloudscrape.com
[41] www.mozenda.com
[42] www.datatoolbar.com
[43] www.thewebminer.com

