Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Extracting Company Information from the Web

Man I Lam, Zhiguo Gong, Jingzhi Guo
Faculty of Science and Technology, University of Macau, Macao, PRC
{ma46522, fstzgg, jzguo}@umac.mo

***Abstract* - As World Wide Web is becoming the most important information repository, increasing amount of information is available. Currently, web search engines can only provide document oriented searches. In order to fully make use of information from the web, some effective and efficient extraction algorithms are definitely desirable. In this paper, some existing achievements are investigated firstly. Then our current technique on web information extraction is discussed in detail. In our approach, rules and patterns are extracted from sample pages through training process, with human involvements. We use both keywords and regular expressions to represent rules and patterns in our system. The keywords work as anchors to locate the positions of the potential information and regular expressions work as validations of the values. In our system, all the extracted information is represented in XML format.**

I. INTRODUCTION

With the increasing development of Internet technologies, the World Wide Web grow rapidly, it becomes one of the primary information repositories. According to the statistical results by Miniwatts Marketing Group[1], during 2000 to 2008, the usage growth of the Internet is 290%. From the statistic for March 2008, there are more than 1.4 billion Internet users from world regions. At the same time, both web sites and web pages are increasing accordingly, which almost covers any kind of information needs. Thus this attracts much attention on how to extract the useful information from the Internet.

Currently, the target web documents can be easily found by using some keywords based search engine, such as GOOGLE[2] or YAHOO[3]. However, the drawback is that the results are ranked according to the term occurrences in the documents other than a piece of specific knowledge. Therefore, this type of search engine does necessarily provide data rich pages [1]. For this reason, many researchers are trying to develop solutions to perform the web data extraction in a more efficient and automatic way.

During the past decade, information extraction has been extensively studied. Since the late 1980's, through the message understanding conference (MUC) sponsored by defense advances research project agency, many information extraction systems have been successfully developed and quantitatively evaluated [2].

In the area of information extraction, the information source can be classified into three main types, including free text, structured text and semi-structured text. For each different kind of information, various techniques are developed. Originally, the extraction system focuses on free text extraction. Natural Language Processing (NLP) techniques are developed to extract this type of unrestricted, unregulated information, which employs the syntactic and semantic characteristics of the language to generate the extraction rules. On the contrary, the structured information provides rigid or well defined formats of information, which usually comes from databases, therefore, it is easy to extract through query language such as Structured Query Language (SQL). The last type is the semi-structured information, which falls between free text and structured information. Web pages are a typical example of semi-structured information. In this paper, we focus on extracting text information from web pages.

There are several challenges in developing web information extraction systems. The main reason is due to the fact that web pages are designed for human browsing, rather than for machine interpretation. Most of the web pages are presented in Hypertext Markup Language (HTML) format[4], which is a semi-structured language. The drawbacks of HTML include the lack of schema, ill formatting, high update frequency and semantic heterogeneity. All of these drawbacks increase the difficulty in developing the system. Therefore, in order to overcome these challenges, our system transforms the page into another format called Extensible Hypertext Mark-up Language (XHTML)[5], which is a stricter and cleaner version of HTML and it is fully compatible with HTML. Then, with a human training process, extraction patterns are generated and represented by regular expressions. The relevant information is extracted by making use of the DOM tree hierarchy of a web page and the extraction patterns. And the extracted information is represented in XML format[6].

The remainder of the paper is organized as follows: in section 2, some related works are illustrated; then detail techniques in our approach are addressed in section 3; in section 4, experimental results are explained; finally, in the last section, the conclusion and future work are mentioned.

II. RELATED WORK

With the explosive growth of the World Wide Web, users can access almost all kind of information in a convenient way. Web content can be reached either by search engine or by manual browsing. From time to time, many extraction systems have been developed. In the very beginning, a procedure is designed for extracting content from particular web pages and

[1] http://www.internetworldstats.com, Miniwatts Marketing Group
[2] http://www.google.com, GOOGLE Search Engine
[3] http://www.yahoo.com, YAHOO Search Engine
[4] http://www.w3.org/TR/html4/, HTTP, W3C Recommendation
[5] http://www.w3.org/TR/xhtml1/, XHTML, W3C Recommendation
[6] http://www.w3.org/XML/, XML, W3C Recommendation

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

this type of procedure is called a wrapper. However, to construct the extraction rules used in a wrapper for a specific domain is complicated and knowledge intensive, only experts may have knowledge to do that. In addition, a wrapper is not adaptive to change, once the structure of the source page is changed, it should be reconstructed accordingly. No doubt, the development cost and the inflexibility for construction are the main disadvantages of using wrappers.

To overcome the extensive work in manually constructing a wrapper, many wrapper generation techniques have been developed. They could be classified into several classes, including language for wrapper development based, HTML-aware based, natural language processing based, wrapper induction based, modeling based and ontology based [3].

Language for Wrapper Development Based: A new language was designed specially to assist the user to accomplish the extraction task. The famous systems for this type include TSIMMIS [4] and Web-OQL [5]. One of the drawbacks of such a model is that not all users are familiar with the new query language, so the performance of the system may not be as expected.

HTML-aware Based: As most of the web pages are in HTML format, another type of extraction system, HTML tree processing based system, was proposed. By parsing the tree structure of a web page, a system is able to locate useful pieces of information. XWRAP [6] and RoadRunner [7] are examples in this respect. In this solution, the system rely on the inherent structure features of HTML document, therefore, web pages need to be in well format. Transformed web pages into a structured formation, such as XHMTL or XML format, can overcome the inherent drawback of HTML pages.

Language Processing Based: Natural Language Processing (NLP) is popularly used to extract data existing in natural language, which belongs to free text information. It makes use of filtering, part-of-speech tagging and lexical semantic tagging technology to build up the extraction rules. For some pages which are mainly composed of grammatical text or paragraphs, this type of tools can be used. SRV [8], WHISH [9] and KnowItAll [10] are examples of this technique. However, for some pages which are composed of the tabular or list format, NLP based tools may not be effective since the internal structure of the page cannot be fully exploited.

Wrapper Induction Based: The wrapper induction based systems can induce the contextual rules for delimiting the information based on a set of training samples, it rely on the formatting features which indicate the structure of the pieces of data. WIEN [11], SoftMealy [12] and STALKER [13] are typical examples. Same as HTML-aware based tools, the performance would be affected due to the ill-formatting of HTML pages.

Modeling Based: In modeling based systems, according to a set of modeling primitives, for example tables or lists, the structure of data is provided. Then the system tries to locate the information which is conformed to the pre-given structure. Therefore providing modeling primitives would be the critical point of this type of tool. NoDoSe [14] is an example of this type of systems.

Ontology Based: Ontology techniques can be used to decompose a domain into objects, and further to describe these objects [13]. This type of system does not rely on the structures of web pages or the grammars of texts but instead an object is constructed for a specific type of data. WebDax [13] is a typical example in this respect.

Besides classifying by the main techniques used, the wrapper can also be grouped into semi-automatic wrapper or fully-automatic wrapper. TSIMMIS [4] and XWRAP [6] belong to semi-automatic wrapper, in which human involvements are necessary and most of extraction tools belong to this type. For the fully-automatic wrapper, such as STAVIES [16], no intervention is needed, it make use of tree structures, or the visual structures of pages to perform the extraction task.

Our proposed system belongs to the type of semi-automatic wrappers. We suppose that with human training, the system can be more adaptive to different type of pages if the training samples are broad enough.

III. EXTRACTION PROCESSING

The system consists of three main processes and one sub-process, they are pages collection process, training process, extraction process and transformation sub-process as shown in Fig. 1.

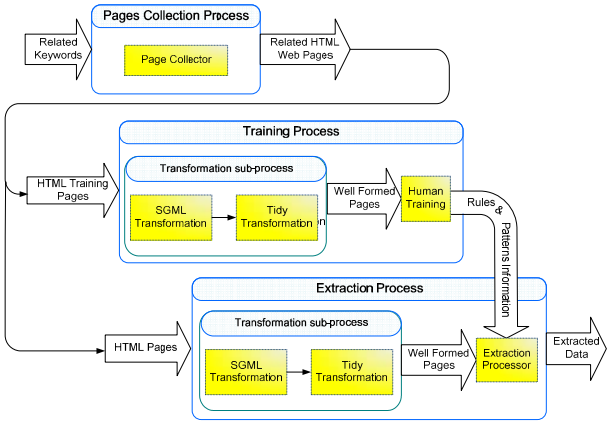

Fig. 1: System architecture

The page collection process is used to collect related web documents, which are the source pages of the system. The page collector use a famous keyword search engine GOOGLE to collect the potential web documents. In the training process, combined with the user's interaction, some information used to generate the extraction will be gathered. The extraction rule generation will be discussed in detail. The transformation sub-process exists in both of the training process and extraction process. It aims to transform web pages into XHTML format, in order to overcome the ill representations of HTML documents. In this approach, two transformation steps are presented, which are SGML and Tidy Transformation. Finally, in the extraction process, the extraction processor bases on the human training result and the DOM tree structure to extract different type of information fields automatically.

A. *Pages Collection*

In this process, the user only needs to input some keywords and select the maximum number of the targeted web documents to the page collector. Then it will automatically download the related HTML web documents.

In our system, for each keyword search, the page collector can only download around 1000 pages and it is due to the restriction of GOOGLE search engine. Although the number of results which appear on the top right corner is always over 1000 pages, it is over-estimated in fact[7]. Most of the famous keyword search engine, such as YAHOO, has this limitation also. In order to solve this limitation, we can search from different search engines, get all the results and do the combination to get more than 1000 results. Then, the more independent search sources you use, the more search results you will get.

The page collector uses the link pattern shown in link 1 to obtain the GOOGLE result page, each of the attributes in link 1 has different purpose, where the value of "q" is the search keywords, "num" is the maximum number of result return to each GOOGLE result page, "start" is the start number of the result links, "lr" set the language of the result links, finally, "as_filetype" restricts the searching document format. By setting different value to the link pattern, different GOOGLE result pages can be obtain.

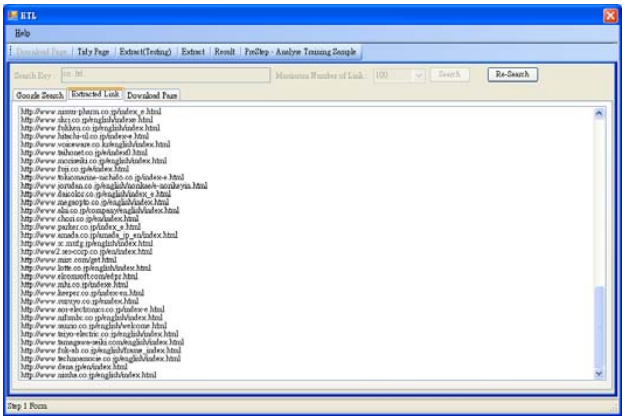

Fig. 2: Screenshot of the page collector (extracted links)

Then by analyzing the HTML, result links inside the GOOGLE result page will be extracted, as shown in Fig. 2. Finally, according to the extracted links, the page collector will download the target web documents and keeps locally.

Table 1

http://www.google.com/search?q=XX&num=100&hl=zh-TW&start=100&sa=N&lr=lang_en&as_filetype=html	[Link 1]

B. *Pages Transformation*

In the transformation sub-process, in order to guarantee the web pages are in well-formatting, two techniques are used. They are Microsoft's SGMLReader[8] and World Wide Web Consortium (W3C) recommended Tidy[9] library. In this system, we use both of them as transformation pipeline. Since the Document Type Definition (DTD) used in Tidy is more restricted then SGMLReader, the SGMLReader transformation will perform first.

C. *Human Training & Extraction Information Gathering*

With human involvement, after the training process, patterns for target information are produced. Let $r(f_1, f_2, \ldots, f_n)$ be a target information schema, where f_i is the field of information and $PSet=\{p_1, p_2, \ldots, p_m\}$ be a set of sample training web pages. Suppose that target records can be extracted at least partly from each of those training pages.

For each field fi, a pattern set, annotated as PS_i, is constructed from sample pages $PSet$. Let $PS_i=\{pn_{i1}, pn_{i2}, \ldots, pn_{im}\}$, a pattern pn_{im} in PS_i is the characteristic of context of f_i in page p_m, which is modeled as a format $\{KW_{im}, EW_{im},\}$, where KW_{im} is the keywords which represent the instance of f_i in page p_m, and EW_{im} is the formulation rule of f_i's value in page p_m, in our system, EW_{im} is modeled using regular expression. For each pattern pn_{im}, suppose multiple of format $\{KW_{im}, EW_{im},\}$ can be derived from page p_m, therefore, $pn_{im}=\{\{(KW_i)_{m1}, (EW_i)_{m1}\}, \{(KW_i)_{m2}, (EW_i)_{m2}\}, \ldots\{(KW_i)_{mj}, (EW_i)_{mj}\}\}$, as shown in Fig 3.

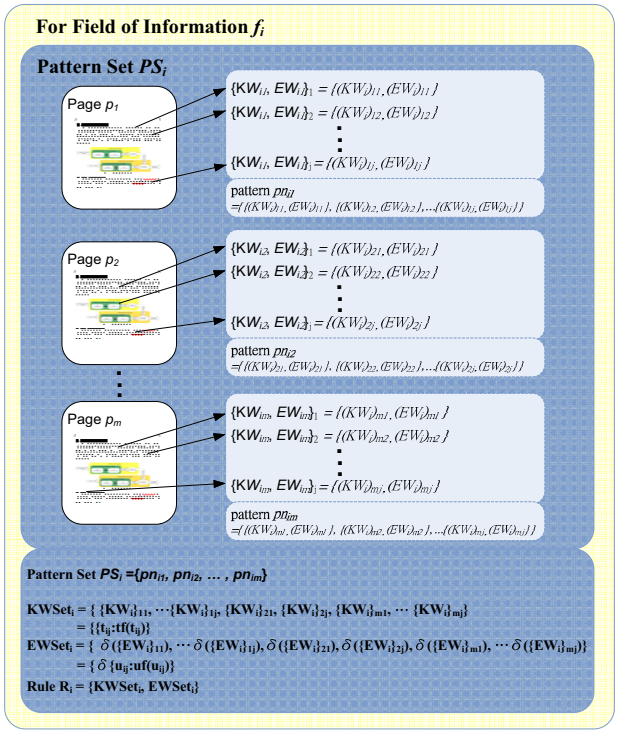

Fig. 3: Rule formulation

The size of pattern set PS_i may become big with more training pages in $PSet$. However, many redundant keywords

[7] http://answers.google.com/answers/threadview?id=18885, Google Limitation

[8] http://msdn.microsoft.com/en-us/library/aa302299.aspx, SGMLReader

[9] http://tidy.sourceforge.net/, Tidy Library

KW_i and value formulation rules EW_i may occur for the same field fi. In order to reduce the size of pattern set PS_i, some reduction algorithm is performed. We further form a set $KWSet_i$ which compose of the keyword KW_i in each pattern pn_{im}, $KWSet_i = \{ \{KW_i\}_{11}, \ldots\{KW_i\}_{1j}, \{KW_i\}_{21}, \{KW_i\}_{2j}, \{KW_i\}_{m1}, \ldots \{KW_i\}_{mj}\}$, then merge the equivalent keywords, so $KWSet_i = \{\{t_{ij}:tf(t_{ij})\}$, where t_{ij} is a keyword from sample pages with respect to field f_i, $tf(t_{ij})$ is the number of occurrences of t_{ij} for field f_i. For the value formulation rules, we also form a set $EWSet_i$, which compose of the regular expression of extracted fields value EW_i . Let δ be the transformation function, then $EWSet_i = \{ \delta(\{EW_i\}_{11}),\ldots\delta(\{EW_i\}_{1j}),\delta(\{EW_i\}_{21}),\ldots\delta(\{EW_i\}_{2j}), \delta(\{EW_i\}_{m1}),\ldots\delta(\{EW_i\}_{mj})\}$, after merging the equivalent part, $EWSet_i = \{\delta\{u_{ij}:uf(u_{ij})\}$, where u_{ij} is a regular expression of field value from sample pages with respect to field f_i, $uf(u_{ij})$ is the number of occurrences of u_{ij} for field f_i. Finally, the extraction rule for f_i, denoted as $R_i = \{KWSet_i , EWSet_i \}$, which provides the rule information for the extraction processor.

D. *Information Extraction Processor*

After the web pages are transformed into well format, the extraction processor makes use of the rules and patterns to extract the target information.

a) *Methodology*

Suppose that the target information field is f_i and from the human training process, we get the rule $R_i = \{KWSet_i , EWSet_i\}$. In the extraction process, the extraction processor makes use of the $KWSet_i$, the keywords of f_i as anchors to locate the position of the interesting information. By analyzing the tag structure, the system gets the nearest tag's value, then the $EWSet_i$ is used to generate a regular expression rule $RERule_i$, which will use to check the validation of the value. If the value does not match with $RERule_i$, the parent tag's value is extended, until the value is matched. The generation of the regular expression rule $RERule_i$ is illustrated in the following part.

Regular Expression Rule Generation

From the human training, a rule information R_i is derived for the target information field is f_i, which contains two sets, $KWSet_i$ and $EWSet_i$. After performing the reduction algorithm, the $EWSet_i$ is the set of regular expression of extracted fields value from sample pages with respect to field f_i, as shown in equation (1) (2) and (3), where δ function is used to transform the content into a regular expression. Then the generated regular expression rule $RERule_i$ is composed of $\delta\{u_{ij}\}$ with "or" relationship.

$$R_i = \{KWSet_i , EWSet_i \} \quad 1)$$

$$EWSet_i = \{ \delta(\{EW_i\}_{11}),\ldots\delta(\{EW_i\}_{1j}),\delta(\{EW_i\}_{21}),\ldots\delta(\{EW_i\}_{m1}), \ldots\delta(\{EW_i\}_{mj})\} \quad 2)$$

$$EWSet_i = \{\delta\{u_{ij}:uf(u_{ij})\} \quad 3)$$

For the δ function, the main syntaxes we have used are shown in table 1. The main idea is to transform all the space character into "\s*", all the text patterns into [\w]* and all the digit patterns into [\d]*. For simplicity, assume that the extraction field fi is the email address, the steps of generating the extraction rule RERulei is shown in figure 4.

$EWSet_i = \{ \delta(staffA@compa.com),\delta(staffB@compb.com),$
$\delta(staff123@compc.com) \}$
$= \{ ([\w]*@[\w]*.[\w]*), ([\w]*@[\w]*.[\w]*),$
$([\w]*[\d]*@[\w]*.[\w]*)\}$
$= \{ ([\w]*@[\w]*.[\w]*), ([\w]*[\d]*@[\w]*.[\w]*)\}$
$RERule_i = ([\w]*@[\w]*.[\w]*)$ "or"
$([\w]*[\d]*@[\w]*.[\w]*)$

Fig. 4: Steps of generating the extraction rule for email

Target Information Extraction

Figure 4 shows an example of a web page's DOM tree structure. In our system, by using the keyword set $KWSet_i$, the extraction processor finds out the position of the keyword first. As in the example, the keyword is located in the tag <TD>, so the first checking area is the content inside the tag <TD>, as shown in checking area 1. If the content is matched with the regular expression rule $RERule_i$, then content will add to the result set, otherwise, the parent tag's content would be the next checking area, as shown in checking area 2, performing the checking again, until finding out the matched content. In the next session, the experimental results is discussed, which will provide two set of resulting figures, one is lacking regular expression rule checking and directly add the content into result set; the other one is including the regular expression rule checking.

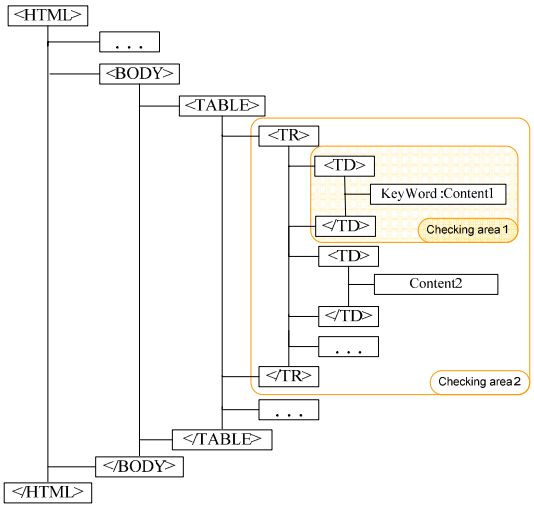

Fig. 5: DOM tree structure of web page example

IV. EXPERIMENTS

All the web pages used in experiment are gathered from GOOGLE search engine, with the keyword "co+ltd.+about+us" or "company+about+us", therefore, the structures and content of them are comparatively divergent. There are totally 500 pages for the training process and 3000 pages for the extraction process.

In our experiment, the extraction record contains 5 fields,

$r(f_1, f_2, f_3, f_4, f_5) = r(ComapanyName, Fax, Telephone, Address, Email)$

The evaluation metrics recall and precision measurements are used to evaluate our system. Precision and recall are defined as the equation (4) and (5) [2]:

Recall = # correct answers / # total possible corrects 4)

Precision = # correct answers / # answers produced 5)

Recall measures the amount of the relevant information that the system correctly extracts, and the precision measures the reliability of the information extracted [17]. Table 2 contains the percentage of page that contains the target filed information, we can notice that not many pages contain the fax or email information. Table 3 shows the resulting figure of our experiments, method 1(M1) is lacking the regular expression rule checking and directly add the content into result set, method 2 (M2) is including the regular expression rule checking.

From the result in table 3, it shows that the overall performance of method2 is better than method 1, on average the recall and precision rate in method 1 is around 0.4, while in method 2 is 0.7. In addition, for the fields which contain a relative stable structure, such as fax number, telephone number and email address, the precision rate is higher than 0.8.

Table 2: Percentage of page that contains the target field information

Extraction Fields				
Company Name	**Fax**	**Telephone**	**Address**	**Email**
95%	37%	57%	48%	37%

Table 3: Performance of the Extraction

	Company Name		**Fax**		**Telephone**	
	M1	**M2**	**M1**	**M2**	**M1**	**M2**
Recall	0.968	0.989	0.622	0.892	0.474	0.702
Precision	0.989	0.989	0.697	0.939	0.750	0.825
	Address		**Email**		**Average**	
	M1	**M2**	**M1**	**M2**	**M1**	**M2**
Recall	0.042	0.542	0.081	0.595	**0.437**	**0.744**
Precision	0.077	0.038	0.115	0.909	**0.526**	**0.744**

However, for the fields that do not have any specific structure, such as the address information, the precision rate is rather low in both methods. This is due to the fact that the regular expression rule has a better performance while the information structure is more representative. In addition, for the address information, many researchers had developed different methodologies which focus on extracting this special type of information [18, 19, 20]. Basically, a knowledge base of countries and cities is necessary and this can be collected from online gazetteer [10]. On average, the existing methodologies claim that the precision and recall rate is around 0.7. In our approach, we could apply some existing address extraction methodologies to increase the performance and this will be an essential task in the future works.

[10] http://www.world-gazetteer.com/

V. CONCLUSION

In this paper, we have discussed the history and current developments in web information extraction and then our approach is discussed in detail. Through training process, our system makes use of analyzing the DOM tree structure and generating the regular expression rule to extract the fields of information. Through the experiment, it shows that the precision and the recall rate are acceptable for overall performance. However, the weakness appears when extracting the information which does not have any specific structure. In addition, the extraction methodology depends on the regular expression rules, which rely on the human involvement. In the future work, we are going to extend our system to overcome these weaknesses.

REFERENCES

[1] Hayrettin Kolukýsaoðlu: *Data Extraction from Repositories on the web: A Semi-Automatic Approach*, Journal of Integrated Design & Process Science, Vol. 7, No. 4, pp. 13-23, (September 2003)

[2] Eikvil, L.: *Information Extraction from World Wide Web – A Survey, Technical Report*, 945, Norweigan Computing Center, Oslo, Norway (July 1999)

[3] Laender, A.H.F., Ribeiro-Neto, B.A., da Silva, A.S., Teixeira, J.S.: *A brief survey of Web data extraction tools*. ACM Sigmod Record 31(2), 84–93 (2002)

[4] Hammer, J., McHugh, J., Garcia-Molina, H.: *Semistructured Data: The TSIMMIS Experience*. Proc. I East-European Workshop on Advances in Database and Information Systems - ADBIS 1997, Petersburg, Russia (1997)

[5] Arocena, G., Mendelzon, A.: *WebOQL: Restructuring Documents, Databases, and Webs*. In: Proc. IEEE Intl. Conf. Data Engineering 1998, Orlando (February 1998)

[6] Liu, L., Pu, C., Han, W.: *XWRAP: An XML-enabled wrapper construction system for web information sources*. In: Proceedings of the international conference on data engineering (ICDE), pp. 611–621 (2000)

[7] Crescenzi, V., Mecca, G., Merialdo, P.: *Roadrunner: Towards automatic data extraction from large web sites*. In: Proc 27th Very Large Databases Conference, VLDB 2001, pp. 109–118 (2001)

[8] Freitag, D.: *Information Extraction from HTML: Application of a General Learning Approach*. In: Proceedings of the 15th National Conference on Artificial Intelligence (AAAI 1998) (1998)

[9] Solderland, S.: *Learning Information Extraction Rules for Semi-structured and Free Text*. Machine Learning 34, 233–272 (1999)

[10] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: *Web-scale information extraction in KnowItAll*

(preliminary results). In: Proceedings of the 13th World Wide Web Conference, pp. 100–109 (2004)

[11] Kushmerick, N. *Wrapper induction: Efficiency and expressiveness*. Artificial Intelligence Journal 118, 1-2 (2000), 15-68

[12] Hsu, C.-N., Dung, M.-T.: *Generating finite-state transducers for semi-structured data extraction from the web*. Information Systems 23(8), 521–538 (1998)

[13] Muslea, I., Minton, S., Knoblock, C.: *Hierarchical wrapper induction for semistructured information sources*. Autonomous Agents and Multi-Agent Systems 4(1/2), 93–114 (2001)

[14] Adelberg, B.: *NoDoSE—A Tool for Semi-Automatically Extracting Structured and Semistructured Data from Text Documents*. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, Seattle, Washington, June 1998, pp. 283–294 (1998)

[15] Snoussi, H., Magnin, L., Nie, J.-Y.: *Toward an Ontology-based Web Data Extraction*. AI-2002 Workshop on Business Agents and the Semantic Web (BASeWEB) held at the AI 2002 Conference (AI-2002)

[16] Papadakis, N.K., Skoutas, D., Raftopoulos, K.: IEEE Computer Society. In: Varvarigou, T.A. (ed.) *STAVIES: A System for Information Extraction from Unknown Web Data Sources through Automatic Web Wrapper Generation Using Clustering Techniques*, IEEE Transactions on Knowledge and Data Engineering, Vol. 17(12), pp. 1638–1652 (December 2005)

[17] Cardie, C.: *Empirical methods in information extraction*. AI Magazine 18(4), 65–80 (1997)

[18] Saeid Asadi, Guowei Yang, Xiaofang Zhou, Yuan Shi, Boxuan Zhai, Wendy Wen-Rong Jiang: *Pattern-Based Extraction of Addresses from Web Page Content*, Progress in WWW Research and Development 10th Asia-Pacific Web Conference, APWeb 2008, Shenyang, China, April 26-28, pp. 407-418 (April 2008)

[19] Wentao Cai, Shengrui Wang, Qingshan Jiang: *Address Extraction: Extraction of Location-Based Information from the Web*, Web Technologies Research and Development - APWeb 2005, 7th Asia-Pacific Web Conference, Shanghai, China, March 29 - April 1, 2005, pp. 925-937 (March 2005)

[20] Abbasi, Rabeeh Ayaz: *Information Extraction Techniques for Postal Address Standardization*, 9th International Multitopic Conference, IEEE INMIC, pp. 1-6 (December 2005)