Domain adaptation of web data extraction based on bootstrapping method

Dong-Lan Liu[1,†], Xin Liu[1], Lei Ma[1], Hao Yu[1], Yong Zhao[2] and Guo-Dong Lv[2]

[1]*State Grid Shandong Electric Power Research Institute*
2000 Wangyue Road, Jinan 250003, PR China
[2]*Shandong Zhongshi Yitong Co.Ltd.*
2000 Wangyue Road, Jinan 250003, PR China
[,†]*e-mail: liudonglan2006@126.com*

Abstract: With the fast development of electric power enterprise information, special structured storage and management system is becoming more and more important. As a uniform interface for multitude of data sources, the efficiency to extract unstructured and semi-structured data existing in webpage is a key issue for Web data integration. In this paper, we grope for the problem of building domain adaptation wrappers for web data extraction. We design the domain adaptation extraction framework based on bootstrapping method. Meanwhile, we discuss the main technologies. We can get the extraction model for recruitment site and then random sampling pages at other power system sites for training the latest wrapper, a uniform data accessing infrastructure in power system domain can be built. In addition, the wrapper has high multipurpose, realizing the domain adaptation extraction. The result of experiment indicates our method can improve the accuracy of extraction in various fields.

Keywords: Domain Adaptation; Web Data Extraction; Bootstrapping; Electric Power Enterprise Information; Wrapper.

1. Introduction

A lot of business systems have been built in the power grid company, and the business systems isolate from each other. With the deepening of enterprise information construction, data, especially unstructured data of every business system sharp increases in the amount. It's inconvenient for people to search for data.

With the rapid development of electric power enterprise information, special structured storage and management system is urgent needed. The goal of a data integration platform is to provide a uniform interface for multitude of data sources. Using the interface, users don't need to consider the problem about mode heterogeneity, data extracting and combination, etc. How to effectively extract the unstructured and semi-structured data existing in webpage is a key issue of

Web data integration, it is the foundation for Web data integration system, and it can provide services for data fusion and analysis.

A certain number of websites can generate extremely structured HTML from databases by applying scripts, for example electric power sites, recruitment sites, shopping sites and form-based websites. The physical correspondence of script-generated webpages can benefit to data extraction systems for using straightforward rules. These rules are so-called wrappers. Lots of texts go halves same HTML tree structure on script generated websites, permitting users to successfully extract interested data from webpages through wrappers. On the other hand, the wrappers breakdown habitually and demand to be re-learned. This is called Wrapper Breakage Problem. As a result, it is extremely significant for web data integration to excellently perk up the adaptive ability of web information extraction.

In this paper, we build a domain adaptation wrapper which can accomplish considerably advanced robustness. The wrapper can extract web data records from webpage, which is stand for "distinguished node". As a consequence, the wrapper can extract data from webpage and decodes into structured data spontaneously by building a set of rules. In most cases, we can only re-locate the data by adjusting wrapper scripts when webpage changes outside the limitation of wrapper script. We apply the existing data of web data integrated system joining with other techniques to identify and put a label on the data elements and attribute tags when webpage changes. After that, we can produce an optimum training sample. At last, we can rebuild a new wrapper by applying the existed wrapper induction methods.

Our contributions include four parts in this research. Firstly, we design a domain adaptation of web data extraction model based on bootstrapping method. Secondly, we present a novel and efficient algorithm for itemizing all minimal candidate wrappers to accelerate the robustness of the evaluation. Thirdly, we propose a robust web extraction algorithm for evaluating the robustness of wrappers. Finally, we carry out a massive set of experiments blanket of numerous websites, the experiment indicates our method can improve the accuracy of extraction in various fields.

The remainder of the paper is prearranged as shown below. In Section II, we introduce domain adaptation extraction framework and related algorithms. The experimental evaluation is submitted in Section III. In Section IV, we introduce the related work. In Section V, we provide the conclusion. Lastly, acknowledgements and related references are presented in this paper.

2. Domain Adaptation Extraction Framework

2.1. Architecture

Firstly, we put forward an architecture of our domain adaptation extraction based on bootstrapping method on the basis of our previous work [18]. Meanwhile, the algorithms in this paper are more improved than previous work. This architecture is illustrated in Fig. 1. We can bootstrap from a few training sites by creating a model of data values and contexts for each schema column in our target domain schema. Furthermore, we can put in the model to lots of websites without the need for human intervention.

The characteristics of the bootstrapping method only need a very small number of labeled data to a field samples, then it can be used in the field of all websites achieve a better extraction effect. Bootstrapping method can be divided into three stages, as shown in Fig. 1. The first two stages are used to extract the field type definitions and sample learning, and the third stage is used to distinguish different node information extraction, which is the key stage defining the efficiency.

In the first stage, for a given node, a candidate wrapper can be generated based on the information of the node and by using the algorithm 1 mentioned in next chapter. After that, the wrapper can be used to manage the process of data extracting. In the second stage, a sample can be extracted from the webpages, then by using the distinguished nodes as trigger word and using algorithm 2, robust wrappers can be selected as initial model. In the third stage, a new data extraction model can be obtained by learning the above stages, in case to survive in the large-scale Web data extraction accurately. New extraction based on the node position of the wrapper corresponds to each website templates, and then improve the original extraction model.

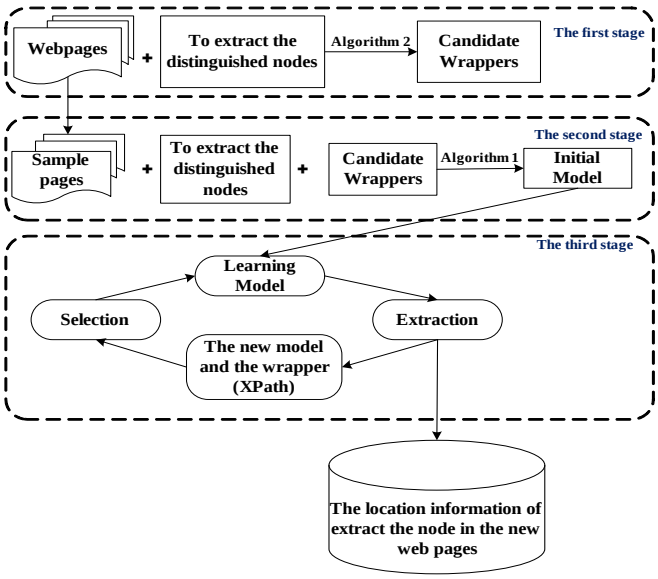

Fig. 1. Domain adaptation extraction framework based on bootstrapping method

2.2. Generating candidate wrappers

We aim to collect the most robust wrapper according to our model by acquiring a series of substitute wrappers. The collection of substitute wrappers should cover a range of hypothetically robust ones. Early research [11] on automatically learning XPath rules from labeled webpages works in a top-down method, that is to say, it begins to the particular paths in each webpage and simplifies them to a solitary XPath. Unluckily, the outcome comprises all conceivable predicates through all webpages.

The literature [2] lately offered a procedure for itemizing wrappers. Nevertheless, the method is not absolute. In addition, they are still not been resolved the headache of building the most robust wrapper. We presented a more helpful approach based on their method in our previous work [18]. It builds wrappers in a bottom-up mode, by starting from the most general XPath which matches and specializes every node until it harmonizes the objective node. But there are some shortcomings in the algorithm proposed by us. In this paper, we will make some improvements to the previous algorithm.

We use D stands for a collection of labeled XML documents, namely, it covers lots of illustrious nodes in matching webpage. For each $d \in D$, we aim at extracting the objective nodes being from D, marked as T(d). For a given XPath expression *x*, we want to get an XPath expression *x* which satiate the next

condition: for each d, *x(d) = T(d)*, i.e. the extraction result is just equivalent to the target node. Assessment Rules are defined as follows.

$$\text{Precision}(x) = \sum_{d} (x(d) \cup T(d)) / x(d) \tag{1}$$

$$\text{Recall}(x) = \sum_{d} (x(d) \cup T(d)) / T(d) \tag{2}$$

We aim at producing an XPath expression, which lets both precision and recall equal to 1. We use the following expression to illustrate.

$$x_0 \equiv \text{// html/body/div/ table/*/td/text()} \tag{3}$$

We describe a one-step specialism of *x* to be an XPath expression generated by any of the following four controllings on x:

1. Increasing a predicate to some node in x. E.g.

$$x_1 \equiv \text{//html/body/div/table[@}width\text{= '98%']/*/td/text()} \tag{4}$$

2. Increasing child position to some node in x. E.g.

$$x_2 \equiv \text{// html/body/div[2]/ table/*/td/text()} \tag{5}$$

3. Increasing a //* at the top of x. E.g.

$$x_3 \equiv \text{//*/ html/body/div/ table/*/td/text()} \tag{6}$$

4. Translating a * to an HTML label name. E.g.

$$x_4 \equiv \text{// html/body/div/ table/tr/td/text()} \tag{7}$$

We represent $x_0 \to x_1$ as a one-step specialism of x_0, and mark $x_0 \xrightarrow{*} x_1$ as multi-step specialisms, more specifically, x_1 can be gained from x_0 using a series of specialisms. The method is preserving a set P of partial wrappers. Each of the elements belong to P is an XPath which has a recall equal to 1 and a precision less than 1. At first, P comprises the single XPath "//*" that harmonizes every node in every document. The method uses specialism steps to XPaths in *P* to obtain new ones constantly until the precision of XPaths equal to 1. After that, XPaths are deleted from P and added to the collection of output wrappers. We can list all XPaths by this method. On the other hand, it is really inefficient to assess the robustness of wrappers by enumerating all XPaths, in addition, the effectiveness is very low. Therefore, we deliberate enumerating all "minimal" candidate wrappers to ameliorate our approach.

For a collection of *D* and an XPath *x*, we say *x* is minimal if there is no other XPath x_0 fits: $x_0 \xrightarrow{*} x$, Precision(x_0) = Precision(x) and Recall(x_0) = Recall(x).

We assume x is a wrapper expressed by XPath expression, that is to say, its precision and recall equal to 1, but x doesn't satisfy the clauses of minimal wrapper. We demand for finding out a smaller XPath x_0 which is also a wrapper. In addition, smaller XPaths expressions are less prospective to break when extracting the data in the pages.

TABLE I. PRODUCING MINIMAL CANDIDATE WRAPPERS BASED ON BOTTOM-UP METHOD

Algorithm 1 Producing minimal candidate wrappers based on bottom-up method

Input: A collection of labeled webpages.
Output: *ResultSet*: a set of wrappers expressed by XPath expressions.

1. *ResultSet* = ∅
2. *P* = {"//*"}
3. **while** $P \neq \varnothing$ **do**
4. Suppose x be any XPath expression in P of partial wrappers, namely, each $x \in P$.
5. $P = P - x$
6. **for all** x_0 s.t. $X \rightarrow X_0$ **do**
7. **if** *isMinimal*(x_0) and *Precision*(x_0) $<$ 1 and *Recall*(x_0) = 1 **then**
8. $P = P \cup x_0$
9. **end if**
10. **if not exist** x_0 and $X \rightarrow X_0$ s.t. *isMinimal*(x_0) = false **then**
11. $P = P - x_0$
12. **end if**
13. **if** *isMinimal*(x_0) and *Precision*(x_0) = *Recall*(x_0) = 1 **then**
14. *ResultSet* = *ResultSet* $\cup$ x_0
15. **end if**
16. **end for**
17. **end while**
18. **Return** *ResultSet*

Clearly, we assume X be any XPath and x a wrapper, such that $X \xrightarrow{*} x$. If x is minimal, X is also minimal. Therefore, we can get all minimal wrappers by listing wrappers and deleting non-minimal XPath in the collection P after every specialism.

The algorithm includes three situations. First of all, if the wrapper x_0 satisfies the qualifications of minimal wrapper, the recall equals to 1 but the precision is less than 1, we will put x_0 in set P and continue looping until the precision reaches to 1. After that, if x_0 in set P is not the minimal wrapper, we will remove it from P. In conclusion, if the wrapper is the minimal one and the precision equal to 1, then we delete the wrapper x_0 from P and add it to the set of *ResultSet*. The algorithm for enumerating minimal wrappers is illustrated in Table I. It can be seen that Algorithm 1 is very comprehensive. As a result, it produces all minimal candidate wrappers and only minimal ones.

2.3. Evaluating the robustness of wrappers

In this part, we describe edit costs for three edit operations: inserting nodes, deleting nodes and substituting labels of nodes. Let L represent the collection of all labels, $l_i \in L$. We assume that there is a cost function cost(x) for computing the cost for each edit operation. S is a given edit script and the cost denoted as *cost(S)*.

Confidence [3] is a measure of how much we trust our extraction of the distinguished node for the given new version. Let S_1 be the smallest cost edit script that takes w to w', namely $S_1(w)$ and w' are isomorphic trees, the node extracted by wrapper is marked $S_1(d(w))$, and the equivalent cost is $cost(S_1)$. We also look at the smallest cost edit script S_2 that takes w to w' but does not map d(w) to the node corresponding to S1(d(w)), and the corresponding cost is $cost(S_2)$. We describe the confidence of extraction as cost $(S_2) - cost(S_1)$.

The literature [3] offered a technique by enumeration for calculating the minimum cost scripts and extracting the data. Afterwards, we invent a more competent algorithm using dynamic programming to calculate the costs of all edit scripts professionally. We will ahead compute the costs of all edit scripts, and finally pick out the most matched data to extract the data. The progression is demonstrated In Table II.

TABLE II. Robust web extraction algorithm

Algorithm 2 Robust Web Extraction Algorithm

Input: W: A webpage; W': the future version of webpage W;$d(W)$: a illustrious node.
Output: $New\_d(W)'$: new location of $d(W)$ in W';*Extr_Conf*: confidence of extraction.

1. For each webpage pair $(W,W') \in \{(W_0, W_1), (W_1, W_2), \cdots, (W_t, W_{t+1})\}$
2. $cost(\varnothing, x)$ =Cost(insert a new node);
3. $cost(x, \varnothing)$ = Cost(delete a exist node);
4. $cost(x, y)$ =Cost(substitute the node x to another node y);
5. *cost function* s.t. $cost(\varnothing, x) + cost(x, \varnothing) + cost(x, y) = 1$ and
   $\arg\min_{x} \prod_{(w_t, w_{t+1}) \in ArchivalData} \{ cost(\phi, x), cost(x, \phi), cost(x, y) \}$
6. Choose W to W' by a set of edit scripts, computing the costs of all edit scripts.
7. Pick out the minimum cost script S_1 such that W' are obtained by applying S_1 in W, namely $S_1(W)$ and W' are isomorphic, i.e.
   $S_1(W) \equiv W'$; $New\_d(W)' = S_1(d(W))$;$Extr\_C1 = cost(S_1)$;
8. Pick out the minimum cost script S_2 such that W' are obtained by applying S_2 in W, i.e. $S_2(W)$ and W' are isomorphic but does not map $d(W)$ to the node corresponding to $S_1(d(W))$, i.e.
   $S_2(W) \equiv W'$ and $New\_d(W)' \neq S_1(d(W))$; $Extr\_C2 = cost(S_2)$;
9. $Extr\_Conf = Extr\_C2 - Extr\_C1$;
10. **Return** $New\_d(W)'$, $Extr\_Conf$

In Algorithm 2, output parameter New_d(W)' stands for the new position of data is articulated by XPath. Output parameter Extr_Conf stands for the confidence. If Extr_Conf is large, the extraction is likely to be precise. The confidence can be accustomed to choose whether to apply the extracted results or not.

2.4. Learning model

We need an assemblage of detail pages from a small number of websites within the target domains as training data which have been instinctively marked using a supervised wrapper induction technology. The objective is to comment on these pages in relation to the domain schema, classifying where schema columns are articulated in spans of text on the webpages. This can be done by first perceiving possible data fields on the webpages, and then categorizing the data fields using a model learned from the training data. We perceive data fields through the pages on the target website by using the Partial Tree Alignment algorithm, which is part of the DEPTA system [16]. We use the method which is based on the one recently proposed by Andrew Carlson et al. [17]. KL-Divergence is a widespread technique for comparing two divisions.

3. **Experimental Evaluation**

We estimate the consequence of our robust domain adaptation extraction framework on two datasets of crawled pages on two real-world sites in our experiments. They include recruitment sites and electric power system sites.

3.1. Data sets

To examine the robustness of our modus operandi, we take advantage of archival data from two websites: recruitment sites and electric power system sites. Each data set contain a collection of webpages from the above websites observed over last three months individually. We select about fifty webpages taking action as archival editions from each website, and we crawl every version weekly.

In every of our data sets, we manual choose illustrious nodes which can be recognized. We pick out illustrious nodes consist of "Position", "Working place" and "Requirement" and so on.

3.2. Evaluation criterion

We use assessment measure from information retrieval [12] in this paper.

We can make use of e *A* stands for the amount of data elements extracted by the technique, *B* the amount of correct data elements, and C the amount of incorrect data elements.

Precision, Recall and F1 measure is:

$$\text{Precision} = \frac{B}{A} \tag{8}$$

$$\text{Recall} = \frac{B}{B+C} \tag{9}$$

$$\text{F1} = \frac{2 \times \text{Pr}\,ecision \times \text{Re}\,call}{\text{Pr}\,ecision + \text{Re}\,call} \tag{10}$$

Precision imitates the trust intensity of the results, and Recall imitates the cover of getting correct results, with F1 manufacturing precision and recall.

3.3. Experimental results and analysis

3.3.1. Evaluating the robustness of wrappers

We carry out two other wrappers for making contrast. One uses the full XPath covering the comprehensive succession of nodes' labels from the root node to the illustrious node in the original description of the webpage. The other one uses the probabilistic robust XPath wrappers from [2]. We name these wrappers FullXPath and ProRobustXPath correspondingly, and our wrapper BootstrapXPath.

The import of a wrapper contains the elderly and new editions of a webpage along with the position of the illustrious node in the webpage. We test out whether the wrapper discoveries the illustrious node in the new edition in respect of every implementation. We examine how well our wrapper achieves as a purpose of pass by time between the elderly and new editions of the webpage. Furthermore, we take advantage of skip sizes as a measurement for the elapsed time. We articulate a pair of editions has a skip of K if the transformation in the middle of the edition numerals of the two editions is K. We assess the precision of wrappers by adjusting the skip size. We draw the consequences of three processes on recruitment sites in Fig. 2(a). It can be seen from the results that our wrapper achieves much better. In the meantime, our wrapper is also appropriate for other domain websites, for example the electric power system sites in Fig. 2(b). For every skip size K, we operate all the wrappers and draw the outcomes in Fig.3. We can discover that the three strategies carry out inferior when skip size is enlarged, however, the results of our method are better.

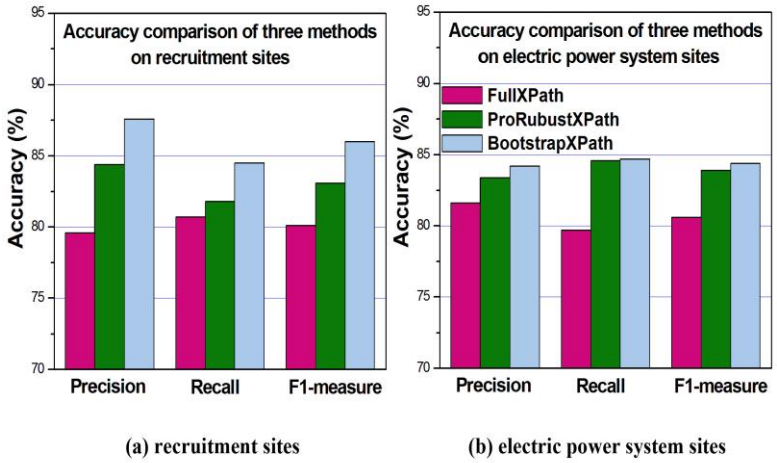

Fig. 2. Presentation evaluations of three schemes

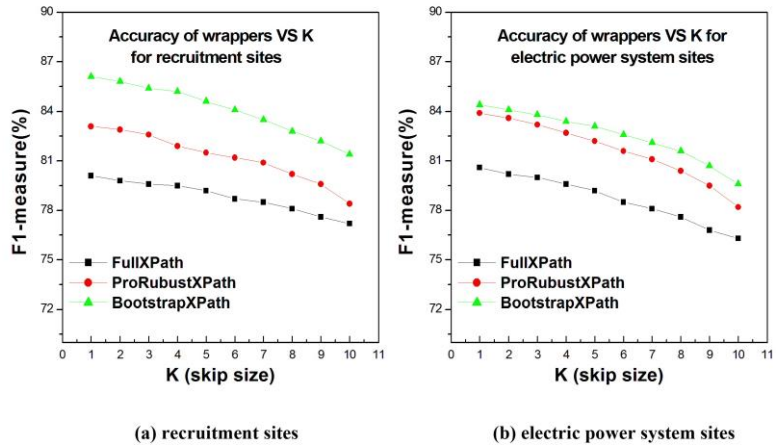

Fig. 3. Accuracy evaluations of three schemes VS K(Skip Size)

3.3.2. Performance comparison with Roadrunner

RoadRunner[7] is a classic method of automatic data extraction. RoadRunner has speculated the common model through comparing their HTML structure on a number of sample pages. The main limitation of the method is an incremental derivation, with the increase of the number of sample pages, the efficiency will be decreased sharply. Process complete automation is a unique feature of RoadRunner. It can be said to be the first fully automatic extraction tool.

This research contrasts the performance of our scheme vs. RoadRunner with the collections of electric power system sites. We pick out thirty databases from

the electric power system sites as testing data and build wrappers applying our scheme and RoadRunner. The middling outcome is reflected as the evaluation criteria. Table 3 shows the performance comparison for 10 electric power system sites. We can find our approach excelled RoadRunner from these results. The main reason is that RoadRunner extraction of data from the target page, according to the web page code to learn extraction rules. And it does not consider other features, such as data content features. Therefore, the extraction accuracy is not high. Our approach makes full use of all kinds of information and features to effectually advance the accuracy of Web data extraction.

TABLE III. Performance comparison of BootstrapXPath and RoadRunner

Electric Power Websites (URL)	**BootstrapXPath**			**RoadRunner**		
	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)
http://zhaopin.sgcc.com.cn/index.jsp	92.6	91.8	92.2	81.5	79.6	80.5
http://www.sgcc.com.cn/index.shtml	91.7	94.5	93.1	78.4	76.2	77.3
http://www.bjx.com.cn	91.9	92.7	92.3	80.1	79.4	79.7
http://12315.ketop.cn	92.4	91.8	92.1	79.1	76.2	77.6
http://www.95598.cn/person/index.shtml	90.2	91.7	90.9	76.5	78.4	77.4
http://www.dlbh.net/ch/index.aspx	90.7	91.2	90.9	75.4	73.9	74.6
http://www.gzdlw.com	89.8	90.6	90.2	72.3	75.5	73.9
http://www.epsforum.com	90.6	89.9	90.2	78.4	80.6	79.5
http://seeipsa.scu.edu.cn	91.1	92.5	91.8	76.2	71.6	73.8
http://www.toobiao.com	89.7	90.8	90.2	72.9	74.2	73.5

4. Related Work

Few study in a straight line explore constructing domain adaptation wrappers. Three researches [1, 2, 3] estimate the robustness of wrappers by putting to the test them on earlier editions of the similar webpage. The literature [1] suggested that convinced wrappers are more robust than others, and it can have evidently decreased fracture. The literature [2] suggested a probabilistic tree-edit method to encapsulate how webpage changes. In spite of their methods permit us to select between a collection of substitute XPaths by assessing wrapper robustness, the drawback of building the most robust wrapper is still left open. The literature [3]

deliberated two methods to research for building the most robust wrapper, namely, the adversarial model and probabilistic model. Evaluating wrapper robustness is accompanying to wrapper repair [13]. The design here is in universal to apply content models of the desired data to learn or repair wrappers. Wrapper induction methods concentrate on how to discover a small number of wrappers from an insufficient training examples [4-11].

In conclusion, there has been some research in recent times on finding robust wrappers. On the other hand, the great majority of this research either finds robust wrappers by human help [1] or finds them from an immovable wrapper language [2]. We can represent one tree to another or searching the smallest tree edit distance has been used to resolve other data extraction difficulties in the literatures [14, 15]. The literature [17] offered the technology is used to extract structured records from semi-structured web pages without human intervention. The literature [19] studied a bootstrapping-based method to automatically extract data-usage statements from academic texts. The literature [20] offered an automatic extraction system which is related to entity and attribute relations (attributes and values) of separate storage. The literature [21] proposed an automatic and effective approach for web entity relation extraction based on priority sequential pattern.

5. Conclusion

In this paper, we explore the conundrum of constructing domain adaptation wrappers based on bootstrapping method for web data extraction. We propose the domain adaptation extraction framework and related algorithms. The process deliberates three edit operations under HTML tree structural varies. It consists of inserting nodes, deleting nodes and substituting labels of nodes. We get hold of the change frequencies of three edit operations for each HTML label-name by making use of our technology on two datasets of crawled pages on two real-world sites, namely, recruitment sites and electric power system sites. The model carries archival data on the real-world recruitment sites as input and learns a model that best fits the data, such that the parameter values minimize the cost of each edit operation. This research bootstrapped onto new sites using training examples from electric system sites. Finally, the model selects the most appropriate data to extract the attentive data from webpages. By means of estimating on real websites, it displays that the proposed strategy can improve the extraction precision of target data, effectually solve the adaptive wrapper for the enormous Web data in various fields.

Acknowledgments

Nearly all this work was done during my working period at State Grid Shandong Electric Power Research Institute. The authors wish to thank the anonymous reviewers and all co-workers for their useful comments.

References

[1] Jussi Myllymaki, Jared Jackson, "Robust web data extraction with xml path expressions," In CiteSeer (2002)
[2] Nilesh Dalvi, Philip Bohannon, Fei Sha, "Robust web extraction: an approach based on a probabilistic tree-edit model," In SIGMOD (2009)
[3] Aditya Parameswaran, Nilesh Dalvi, Hector Garcia-Molina, Rajeev Rastogi, "Optimal Schemes for Robust Web Extraction," In VLDB (2011)
[4] Nilesh Dalvi, Ravi Kumar, Mohamed Soliman, "Automatic Wrappers for Large Scale Web Extraction," In VLDB (2011)
[5] Michael J. Cafarella, Alon Halevy, Nodira Khoussainova, "Data Integration for the Relational Web," In VLDB (2009)
[6] Robert Baumgartner, Georg Gottlob, Marcus Herzog, "Scalable Web Data Extraction for Online Market Intelligence," In VLDB (2009)
[7] Rahul Gupta, Sunita Sarawagi, "Domain Adaptation of Information Extraction Models," SIGMOD Record, 37(4):35–40 (2008)
[8] Gjergji Kasneci, Maya Ramanath, Fabian Suchanek, Gerhard Weikum, "The YAGO-NAGA Approach to Knowledge Discovery," In SIGMOD Record, 37(4):41–47 (2008)
[9] Michael J. Cafarella, Jayant Madhavan, Alon Halevy, "Web-Scale Extraction of Structured Data," In SIGMOD (2008)
[10] Yeonjung Kim, Jeahyun Park, Taehwan Kim, Joongmin Choi, "Web Information Extraction by HTML Tree Edit Distance Matching," ICCIT (2007)
[11] Tobias Anton, "Xpath-wrapper induction by generating tree traversal patterns," In LWA, p. 126–133 (2005)
[12] R. C. van.: Information Retrieval [M]. Butterworths (1979)
[13] Boris Chidlovskii, Bruno Roustant, Marc Brette, "Documentum eci self-repairing wrappers: performance analysis," In SIGMOD, p. 708–717 (2006)
[14] Davi de Castro Reis, Paulo B. Golgher, Altigran S. da Silve, "Automatic web news extraction using tree edit distance," In WWW, p. 502–511 (2004)
[15] Wang W, Xiao C, Lin X, Zhang C, "Efficient approximate entity extraction with edit distance constraints," In SIGMOD, p.759-770 (2009)

[16] Y. Zhai and B. Liu, "Web data extraction based on partial tree alignment," In: Proceedings of 14th international conference on World Wide Web, p. 76-85(2005)

[17] Andrew Carlson, Charles Schafer, "Bootstrapping Information Extraction from Semi-structured Web Pages," ECML PKDD, Part 1, LNAI 5211, p.195-210(2008)

[18] Donglan Liu, Lei Ma, Xin Liu, "Research on Adaptive Wrapper in Deep Web Data Extraction," In IOV2015, LNCS 9502, p.409–423 (2015)

[19] Qiuzi Zhang, Qikai Cheng, Yong Huang, Wei Lu, "A Bootstrapping-based Method to Automatically Identify Data-usage Statements in Publications," In JDIS, 1(1), p.69-85 (2016)

[20] Zou Yuwei, Gu Jinguang, Fu Haidong, "Medical Entity and Attributes Extraction System Based on Relation Annotation. Wuhan University Journal of Natural Sciences," 21(2), p.145-150 (2016)

[21] Min Yin, Wei Zhang, Jing Yang, Zhishu Wang, "A Web Entity Relation Extraction Algorithm Based on Priority Sequential Pattern," In CITCS (2015)