2012 2nd International Conference on Information Communication and Management (ICICM 2012)
IPCSIT vol. 55 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V55.19

An Algorithm of Product Information Extraction from Web Pages: a Document Object Model Analysis Approach

Worasit Choochaiwattana[+]

Search Engines and Intelligent Information Systems Research Laboratory
Graduate Program in Web Engineering, Faculty of Information Technology, Dhurakij Pundit University

Abstract. A number of people searching for information on the Internet have been rapidly increasing. Web search engines become the most efficient and convenient way to discover interesting information. With this huge collection of information, a development of product search engines becomes a challenging task because of variety of web page formats and authoring styles. To improve an efficiency of the product search engines, classification and extraction features should be embedded web crawlers. This paper focused only on the extraction features and investigated how well the DOM tree structures of web pages contributes to product information extraction task when they were used to identify pieces of the product information on web pages. Two algorithms for product information extraction tasks were proposed: Extraction Based on Sub-Tree (ExBST) and Extraction Based on Sibling Nodes (ExBSN). The result of the experiment revealed that the combination of ExBST and ExBSN algorithm had a 93.63% of accuracy.

Keywords: product information extraction, product search engine, price comparison system

1. Motivation

Over the last decade, the growth and popularity of Internet usage has resulted in a huge amount of information available on the Internet. According to the study of Internet usage statistic[1], people spend around 21% of their Internet usage time on searching and finding their desire information. Thus, using web search engines has become the most convenient way to help people discover their desired contents in this huge collection of information. Searching for product information on web search engine, however, is clearly inadequate and has some limitations. Search results returned from the web search engine only match the similarity between query terms and content of web pages. People by themselves have to consider the results to select pieces of information they desire such as product image, product description, and price.

The introduction of vertical search and entity search embedded with information extraction techniques makes it possible to develop more sophisticated web based system, e.g. product search engines [1] [2] and research paper search engines [3]. Techniques of Information extraction from web pages has been applied to the sophisticated web based systems for different proposes, such as, extracting course information [1], extracting product information [4] [5] [6] [7] , and extracting travelling information [8].

The diversify of web page format and variety of authoring style makes the process of information extraction from web pages more challenging. Several techniques have been proposed. Fatima Ashraf *et al.* [9] and Kostyantyn Shchekotykhin *et al.* [10] employed clustering technique for extracting information from HTML documents. They found that product information in many electronic commerce systems represented in tabular form consisted of attributes, sub attributes and values of sub attributes. In addition, information extraction techniques, proposed by Wolfgang Gatterbauer *et al.* [11], showed evidence that extracting useful information from web table could be done effectively by analysing the structure of the data in the table on

[+] Corresponding author. Tel.: + 66-2-954-7300 ext 786; fax: +66-2-589-1140
E-mail address: worasit.cha@dpu.ac.th
[1] http://www.go-gulf.com/blog/online-time

web pages. As mentioned in Mahmoud Shaker *et al.* [4], tables on web pages were used for presenting and constructing a layout of web pages. With the introduction of Cascading Style Sheet (CSS), however, the authoring style on how to construct the layout of web pages has been changed from relying mainly on tables to dividing information into division and using CSS to construct the layout of web pages. Thus, more robust and effective techniques need to be developed. There were, however, few published research papers focused on analysing Document Object Model (DOM) tree structure or web page structure to augment information extraction task especially for product information on electronic commerce systems.

Thus, this paper investigated how well the DOM tree structures of web page contributes to product information extraction when they were used to identify pieces of the product information on web pages.

2. Proposed Approach

2.1. Problem description

One of the main problems that vertical search systems and entity search systems encounter with is how to extract useful information that helps them to perform more efficiently and more effectively. For this particular study, the goal of product search engines is to present useful information about products for their user. Thus, the product search engines need to accurately identify product information web pages and extract important information from these pages. The important information includes product name, product description, product image, and product price.

The traditional approach to develop effective product search engine is to 1) develop a general purpose crawler for downloading all web resources, 2) design and build a mechanism to extract product information for indexing process, 3) implement a searching function to retrieve and display search results. Most of search engines employ this approach to build their product search feature. This approach is appropriate for search engine developers with product searching feature. It is, however, considerably inefficient when the developers plan to develop only product search engines or price comparison systems since the crawler has to download not only product related web resources but also others web resources.

To improve an efficiency of the product search engine, the web crawler should be embedded with a classification and extraction features. The classification feature is responsible for determining whether content of web resources contain product information. Meanwhile, the extraction feature is responsible for extracting product information and other useful information. However, this paper only focused on the extraction feature of the web crawler for product search engines.

2.2. Proposed algorithms of product information extraction

The representation of HTML Web page is in a format of DOM tree structure. As illustrated in Figure 1, the leaf nodes in the DOM tree contain text content and format information.

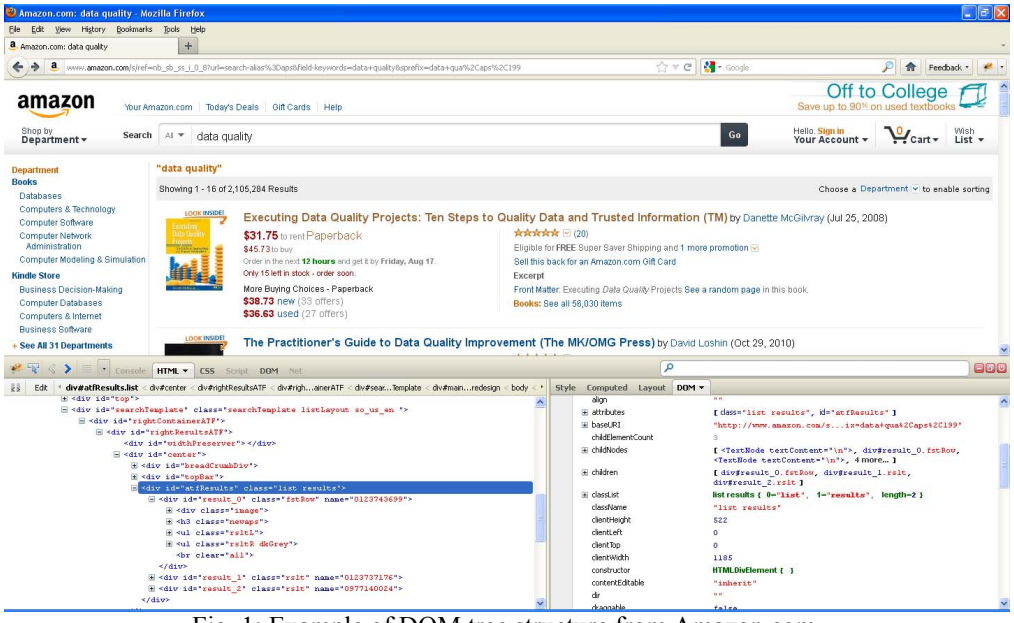

Fig. 1: Example of DOM tree structure from Amazon.com

After observing DOM tree structure of several electronic commerce web sites, two algorithms for product information extraction were proposed: Extraction Based on Sub-Tree (ExBST) and Extraction Based on Sibling Nodes (ExBSN). The assumption of these two algorithms is that product information object contains product image, product name and description, and product price. The ExBST assumes that product information is normally tied together as a sub-tree. On the other hand, the ExBSN consider each node and its sibling nodes to determine whether the considered nodes contain the product information.

Let W be a set of web pages, $W = \{w_1, w_2, w_3, \dots, w_n\}$, contains all web pages in the collections. Each web page, w_i, can be represented as $n_1, n_2, n_3,\dots,n_k$ where n is the node in DOM tree structure. Let E be a k x k association metric between the nodes in DOM tree structure. $E(n_i,n_j)$ will be equal to 1 when there is a edge from node n_i to node n_j. Let $NodeType(n_i)$ and $ParentNode(n_i)$ be a function returns type of node n_i and a function returning a parent of node n_i respectively.

In addition, let $getProductInfoNode(n_i)$ be a function returning a child node of n_i that is a product information node. Thus, another function that helps $getProductInfoNode(n_i)$ perform its task is $isProductInfoNode(n_k)$, a function for determining whether node n_k contains product name and description, and product price. Let $isSiblingNode(n_i,n_j)$ and $getSiblingNode(n_i)$ be a Boolean function returning true when node n_i and n_j share the same parent node and a function returning a sibling node that locates next to node n_i respectively. Last but not least, let $ExtractProductInfoObject(n_i, n_j)$ be a function to extract content from node n_i and n_j. The figure 2 and figure 3 show ExBST and ExBSN algorithm.

ExBST Algorithm

Input : Web page w_i that contained all the node, n_1 to n_k, in DOM tree
Metric E that contained edge information between each node in w_i
Output : Extraction Result Set, S, that contained extracted product information objects

Step 1 : S = { } // initiated S
Step 2 : **for all** Node n_j in Web page w_i **do**
if $NodeType(n_j)$ == IMG_TYPE **then**
$ImgObj$ ← nil // initiated $ImgObj$ for a image object found in Node n_j
$ImgObj$ ← $DownloadIMG(n_j)$
if (isProductImage($ImgObj$)) **then**
$pNode, infoNode$ ← nil // initiated $pNode$ and $infoNode$
$pNode$ ← $ParentNode(n_j)$ // get parent node of n_j
$infoNode$ ← $getProductInfoNode(pNode)$ // find product information node in the subtree of pNode
if $infoNode$!= {} **then**
S ← $ExtractProductInfoObject(n_j, infoNode)$
end if
end if
end if
end for
Step 3 : **return** S

Fig. 2: ExBST algorithm

ExBSN Algorithm

Input : Web page w_i that contained all the node, n_1 to n_k, in DOM tree
Metric E that contained edge information between each node in w_i
Output : Extraction Result Set, S, that contained extracted product information objects

Step 1 : S = { } // initiated S
Step 2 : **for all** Node n_j in Web page w_i **do**
if $NodeType(n_j)$ == IMG_TYPE **then**
$ImgObj$ ← nil // initiated $ImgObj$ for a image object found in Node n_j
$ImgObj$ ← $DownloadIMG(n_j)$
if $isProductImage(ImgObj)$ **then**
$siblingNode$ ← nil // initiated $siblingNode$
$siblingNode$ ← $getSiblingNode(n_j)$ // get sibling node that locates next to node n_j
if $isProductInfoNode(siblingNode)$ **then**
S ← $ExtractProductInfoObject(n_j, siblingNode)$
end if
end if
end if
end for
Step 3 : **return** S

Fig. 3: ExBSN algorithm

3. Experiment

3.1. Data Collection

Data was crawled from Google between June and July 2012. Beginning with a given set of product information query terms, a set of URLs can be retrieved. Given those URLs more product-related URLs can be obtained. Over the two months of building data collection, the crawlers looked at approximately 5,000 URLs. The crawler randomly selected URLs from a set of retrieved URLs to create data collection for the experiment. The final data set consisted of 60 web pages.

3.2. Evaluation Metric and Experimental Setting

To evaluate the performance of the proposed algorithm, the percentage of accuracy was used as illustrated in equation 1.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (1)$$

where *TP (True Positive) = a number of correctly extracted product information objects*
TN (True Negative) = a number of unextracted non product information objects
FP (False Positive) = a number of extracted non product information objects
FN(False Negative) = a number of unextracted product information objects

To compute the accuracy, the extraction results from the proposed algorithm were compared against a manual extraction by human. A group of research participants, performing manual extraction, were invited from a list of faculty members from the faculty of information technology, Dhurakij Pundit University. The participants in this experiment were considered experts. Thus, the results of manual extraction were considered perfect. The evaluation system was developed to collect a result of manual extraction. Sixty web pages from the final data set were randomly present one at a time. The participants were asked to perform a task of manually extract product information object from each web page and then enter a number of extracted product information objects into the system. Before performing the task, the participants were informed that a product information object should include product image, product name and description, and product price. After the results of manual extraction by the participant were recorded, the accuracy of the proposed algorithms then was computed.

4. Result, Discussion, Conclusion and Future Work

The main objective of the experiment was to compare the performance of the proposed algorithms (ExBST Algorithm, ExBSN Algorithm, and the combination of ExBST and ExBSN Algorithm) with manual extraction by human, which is the baseline of this study. To assess the performance of the algorithms, the percentage of accuracy for each algorithm was examined. The accuracy is traditionally used for the performance evaluation in the study of information extraction. The higher of the accuracy should indicate a more effectiveness of the extraction algorithms. Table 1 provides a summary of the accuracy for all algorithms when performing product information extraction of sixty web pages in the final dataset.

Table 1: Evaluation Results of the Proposed Algorithms

	ExBST Algorithm	ExBSN Algorithm	ExBST + ExBSN Algorithm
Accuracy	89.42%	72.13%	93.63%

The result suggests that the combination of ExBST and ExBSN algorithm provides a higher accuracy in product information extraction from web page and outperforms when each of them extracts the information by its own. Although the performance of the combined algorithms is at 93.63% of accuracy, the causes of error in the combined algorithms are from the performance of the *isProductImage(ImgObj)* function and some authoring style of web pages.

As mentioned earlier, product image is an important element in product information object. To determine whether an image is product image, for this particular study, an image object should be at least 50 pixels of width and height. The error occurred when a product image is smaller than 50 pixels of width and height. An error in loading image object from web pages results in the *isProductImage(ImgObj)* function returning a

false even though there is a product image. In addition, some authoring style of web page such as presenting product information using iframe tag and a complex table can result in errors in extracting a product information object.

This paper investigated one of the important issues on how to improve a performance of web crawlers for the product search engine, which is the information extraction feature. The evaluation result on the proposed combined algorithms based on DOM tree structure showed a promising approach on the product information extraction. To develop more effective and efficient web crawlers, another important issue, which is classification feature, need to be examined. Last but not least, more robust extraction feature based on DOM tree structure should be explored.

5. References

[1] P. Luo, F. Lin, Y. Xiong, Y. Zhao, and Z. Shi. Towards Combining Web Classification and Web Information Extraction : a Case Study. In *Proc. of ACM SIG KDD Conference on Knowledge Discovery and Data Mining*. Paris, France. 2009, pp 1235 – 1243.

[2] M. Castellanos, Q. Chen, U. Dayal, M. Hsu, M. Lemon, P. Siegel and J. Stinger. Component Adviser: a Tool for Automatically Extracting Electronic Component Data from Web Datasheets. In *Proc. of the 7th International Conference on World Wide Web*. Brisbane, Australia. 1998.

[3] Z. Nie, J. Wen, and W. Ma. Object-level Vertical Search. In *Proc. of the Conference on Innovative Data Systems Research*. California. 2007.

[4] M. Shaker, H. Ibrahim, A. Mustapha, and L.N. Abdullah. Information Extraction from Web Tables. In *Proc. of The 11th International Conference on Information Integration and Web-based Applications & Services*. Kuala Lumpur, Malaysia. 2009, pp 470 – 476.

[5] S. Vadrevu, F. Gelgi, and H. Davulcu. Information Extraction from Web Pages using Presentation Regularities and Domain Knowledge. *World Wide Web*. 2007, **10**: 157–179.

[6] T.L. Wong, W. Lam, and T.S. Wong. An Unsupervised Framework for Extracting and Normalizing Product Attributes from Multiple Web Sites. In *Proc. of The 31st Annual International ACM SIGIR Conference*. Singapore. 2008, pp 35 – 42.

[7] B. Li, A. Ghose, and P.G. Ipeirotis. A Demo Search Engine for Products. In *Proc. of The 20th International Conference on World Wide Web*. Hyderabad, India. 2011, pp 233 – 236.

[8] S.W. Jung, K.H. Sang, T.W. Park, and H.C. Kwon. Intelligent Integration of Information on the Internet for Travelers on Demand. In *Proc. of The 2001 IEEE International Symposium on Industrial Electronics*. Pusan, Korea. 2001, pp 338 – 342.

[9] F. Ashraf, T. Ozyer, and R. Alhajj. Employing Clustering Techniques for Automatic Information Extraction form HTML Documents. *IEEE Transactions on Systems*. 2008, **38**: 660 – 673.

[10] K. Shchekotykhin, D. Jannach ,and G. Friedrich. Clustering Web Documents with Tables for Information Extraction. In *Proc. of The 4th International Conference on Knowledge Capture*. Canada. 2007, pp 169 – 170.

[11] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krupl, and B. Poolak. Towards Domain-independent Information Extraction from Web Tables. In *Proc. of The 16th International Conference on World Wide Web*. Canada. 2007, pp 71 – 80.

Worasit Choochaiwattana was born in 1975. He earned a B.B.A.(Marketing) in 1996 from Chulalongkorn University, Thailand, a M.Sc.(Technology of Information System Management) in 2000 from Mahidol University, Thailand, a M.Sc.(Computer Science) in 2003 from Chulalongkorn University, Thailand, and a Ph.D.(Information Science) from University of Pittsburgh, Pennsylvania, United States. Now, he is an assistance professor at Faculty of Information Technology, Dhurakij Pundit University, Thailand. His research interests include search engines, adaptive web and intelligent information systems.