IBIMA Publishing
Communications of the IBIMA
http://www.ibimapublishing.com/journals/CIBIMA/cibima.html
Vol. 2013 (2013), Article ID 743515, 9 pages
DOI: 10.5171/2013.743515

Research Article

Region Based Data Extraction

Pui Leng Goh, Jer Lang Hong, Ee Xion Tan and Wei Wei Goh

School of Computing and IT, Taylor's University, Malaysia

Correspondence should be addressed to: Pui Leng Goh; Puileng.Goh@taylors.edu.my

Received 23 May 2012; Accepted 19 September 2012; Published 30 December 2012

Academic Editor: Abdullah Gani

Abstract

Wrappers are tools used to extract relevant information from HTML pages. Current approaches use DOM tree, visual cue, and ontology to extract data. DOM tree based techniques are fast and simple. However, they are not as accurate as visual based wrappers due to lack of additional information needed to perform data extraction. Visual based wrappers, on the other hand, are slow due to the extra processing needed to obtain visual cue from the underlying browser rendering engine. Ontology based wrappers are accurate, but they are also slow and need manual tuning to operate them. In this paper, we propose a novel visual based wrapper to extract information from HTML pages. Our wrapper uses visual cue to eliminate unnecessary regions, hence reduces the running time of extraction task as our wrapper only needs to consider the relevant region for extraction. Then, our wrapper removes irrelevant data from the relevant region using visual cue. Experiment results show that our wrapper outperforms state-of-the-art wrapper WISH in data extraction.

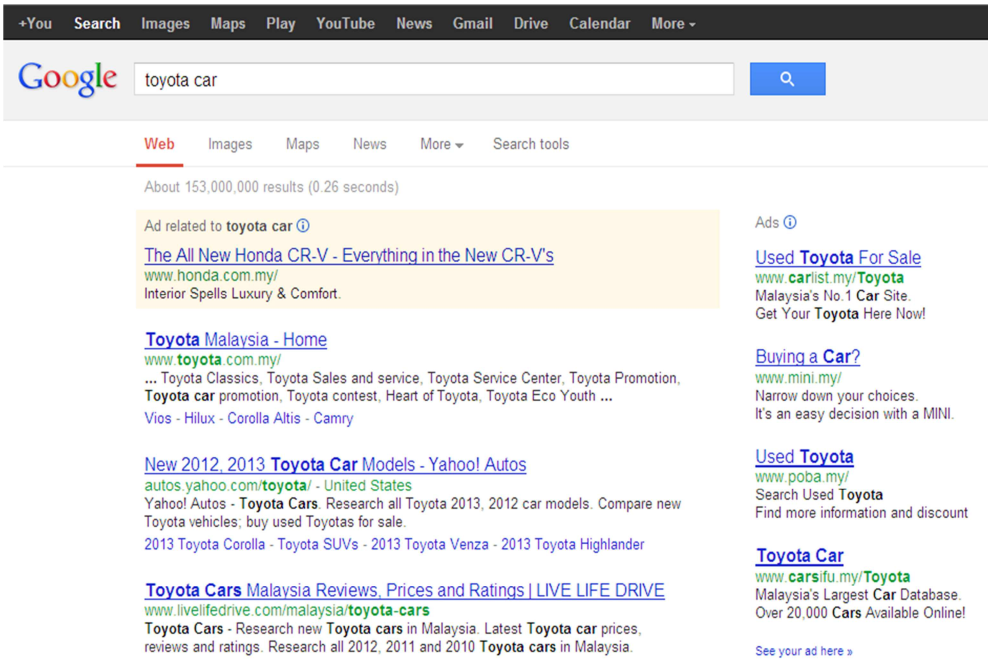
Keywords: Automatic Wrapper, Search Engines, Deep Web.

Introduction

The advent of World Wide Web has seen great amount of data available, which can be images, HyperText Markup Languages (HTML) pages, or audio/video clips. To locate these data efficiently, search engines are developed to locate, index, and present them in a meaningful format for user viewing. However, these data are of diverse nature and contain various formats, layouts, and presentation. To search these data, specialized search engines (known as meta search engine) such as Google Scholar are developed which can locate, understand, and process this particular data before presenting them for user viewing. Before a meta search engine can search for data, it needs to locate the data (also known as data records) from other search engines (Figure 1), extract them, filter out the irrelevant data, and rank them accordingly. The tools to extract this data from various search engines are termed *wrapper* Jer L. H. (2010), Jer L. H. (2011). However, extraction of data from search engines is difficult due to the following reasons 1) HTML language is ambiguous and not uniformly presented 2) Data in search engines are semi structured, they contain various forms and layouts 3) Search engine result pages may also contain other irrelevant data (known as data regions) in addition to the relevant data. To develop an accurate extraction tool, we need to develop a set of heuristic rules which can cater for all the above mentioned problems.

Cite this Article as: Leng Goh, Jer Lang Hong, Ee Xion Tan and Wei Wei Goh (2013), "Region Based Data Extraction," Communications of the IBIMA, Vol. 2013 (2013), Article ID 743515, DOI: 10.5171/2013.743515

In this paper, we propose a novel wrapper which can extract search results from search engine result pages using a two stage extraction techniques. Unlike existing approaches, our wrapper first identifies the relevant region at a global level and then removes the irrelevant data within that region at a local level. This approach could lead to higher accuracy as we remove irrelevant regions first at global level and then we remove irrelevant data at local level within the relevant region. In other words, our wrapper removes irrelevant regions and data at appropriate level of extraction stages. Preliminary version of this paper has appeared in Pui L. G., Jer L. H., Ee X. T., Wei W. G. (2012).



toyota car

About 153,000,000 results (0.26 seconds)

Ad related to toyota car

The All New Honda CR-V - Everything in the New CR-V's
www.honda.com.my/
Interior Spells Luxury & Comfort.

Toyota Malaysia - Home
www.toyota.com.my/
... Toyota Classics, Toyota Sales and service, Toyota Service Center, Toyota Promotion, Toyota car promotion, Toyota contest, Heart of Toyota, Toyota Eco Youth ...
Vios - Hilux - Corolla Altis - Camry

New 2012, 2013 Toyota Car Models - Yahoo! Autos
autos.yahoo.com/toyota/ - United States
Yahoo! Autos - Toyota Cars. Research all Toyota 2013, 2012 car models. Compare new Toyota vehicles; buy used Toyotas for sale.
2013 Toyota Corolla - Toyota SUVs - 2013 Toyota Venza - 2013 Toyota Highlander

Toyota Cars Malaysia Reviews, Prices and Ratings | LIVE LIFE DRIVE
www.livelifedrive.com/malaysia/toyota-cars
Toyota Cars - Research new Toyota cars in Malaysia. Latest Toyota car prices, reviews and ratings. Research all 2012, 2011 and 2010 Toyota cars in Malaysia.

Ads

Used Toyota For Sale
www.carlist.my/Toyota
Malaysia's No.1 Car Site.
Get Your Toyota Here Now!

Buying a Car?
www.mini.my/
Narrow down your choices.
It's an easy decision with a MINI.

Used Toyota
www.poba.my/
Search Used Toyota
Find more information and discount

Toyota Car
www.carsifu.my/Toyota
Malaysia's Largest Car Database.
Over 20,000 Cars Available Online!

See your ad here »

Figure 1: Single Section Data Records (Google Search Engine)

This paper is divided into several sections. Section II describes the works that are related to ours. Section III describes the methodology and operation of our wrapper while Section IV gives the experimental evaluation of our wrapper. Finally Section V concludes our work.

Related Work

Overview

In the early days, manual wrappers were developed to extract specific data records from a HTML page. This approach is inflexible as it requires a knowledgeable user to operate a wrapper. Manual wrappers are also error prone, and not easily extensible, as the programmer needs to make the necessary modifications to the wrapper should a company requirements for its web site changes. Therefore the developer has to recourse to the development of supervised and semi supervised wrappers which can extract data records automatically. Before these wrappers begin their operation, they require human labeling of the data so that the wrappers can extract the relevant portion of the data, which is of interest to the user. Semi supervised wrappers can predict the extraction rule once data labeling and data extraction are carried out. However, these wrappers also have limitations, notably that they require human labeling and intervention before they begin their operation. To overcome this limitation, fully automatic wrappers were developed to extract data records without any human involvement Arvind A. and Hector G-M (2003), Valter C., Giansalvatore M., and Paolo M. (2001).

There are two types of automatic wrappers available currently 1) Document Object Model (DOM) Tree Based Wrappers 2) Visual Based Wrappers. Details of these two types of wrappers are presented in the next section.

DOM Tree Based Wrappers

DOM Tree Based Wrappers use the properties of DOM Tree such as sibling nodes, parent nodes, and the depth of the trees. It then manipulates these trees by finding regularity in their structures. Mining Data Records (MDR) Bing L., Robert G., and Yanhong Z. (2003) checks for repetitive patterns such as HTML Tags to detect data records. It can also group these patterns as blocks of tags. Then, MDR uses string edit distance to match the similarity of data records. Although MDR is able to match data records efficiently, its operation matches the similarity of data records in a single level only. Visual Perception based Extraction of Records (ViPER) Kai S. and Georg L. (2005) on the other hand, enhances the algorithm of MDR by using primitive tandem repeat which detects the repetitive sequence of HTML Tags using a matrix.

Wrapper Incorporating Set of Heuristic Techniques (WISH) Jer L. H., Eugene S., Simon E. (2010) uses frequency measures to match the tree structures of data records. WISH works in a time complexity of O(n) and is able to match tree structures of data records containing iterative and disjunctive data. However, tree matching algorithm of WISH is not able to match data records with dissimilar tree structures.

Data Extraction based on Partial Tree Alignment (DEPTA) Yanhong Z. and Bing L. (2005) uses a bottom up tree matching algorithm to match tree structures of data records. A tree matching algorithm matches two tree structures and determines how the first tree can be transformed into the second tree. DEPTA's tree matching algorithm determines the maximum matches between two trees by comparing the location and identity of the nodes in the tree structures. Although this algorithm solves the problem emerged in data matching successfully, the algorithm requires a complex data structure for its implementation. Therefore, an algorithm that could simplify the implementation process will be helpful. DEPTA's tree matching algorithm works in a time complexity of O (n_1n_2) time (where n_1 is the number of nodes in the first tree and n_2 is the number of nodes in the second tree).

Visual Based Wrappers

Visual Based Wrappers use visual cue of HTML page, such as font size, block boundaries, and object relative position, to extract relevant information. Visual and Tags (ViNT) Hongkun Z., Weiyi M., Zonghuan W., Vijay R., and Clement Y. (2005) extracts content line features from the HTML page, where a content line is a type of text which can be visually bounded by a rectangular box. Content lines are categorized into 8 types, each with their own distinguishing characteristics and features, which are grouped to form content blocks. ViNT parses these content blocks to identify the data records. Essentially, ViNT defines a data record as a content block containing a specific ordering of content lines. This is a reasonable assumption as search engine result pages are typically highly structured. However, ViNT may exclude valid data records with irregular content block patterns. Moreover, ViNT uses several sample pages to extract the correct data region from the HTML page and does this by parsing the samples to identify static and dynamic HTML regions, excluding dynamic regions from the overall data region of interest. While we consider ViNT to be very robust as a result, our wrapper questions the need for such overheads.

ViPER Kai S. and Georg L. (2005) takes a more "natural" approach by projecting the contents of the HTML page onto a 2-dimensional X/Y co-ordinate plane, effectively simulating how the HTML page may be rendered on a printed hard-copy. This enables ViPER to compute two content graph profiles, one for each X and Y planes, which it uses to detect data regions by locating valleys between the peaks as the separation point between two data records (valleys are usually the space within two data records, separating them apart). However, this method assumes that the data regions are separated by well defined empty space regions, an assumption which may not always hold true and is a source of error for the ViPER wrapper.

Visual Segmentation based Data Records (VSDR) Longzhuang L., Yonghuai L., Abel O., and Matt W. (2007) and Visual Data Extraction (ViDE) Wei L., Xiaofeng M., and Weiyi M. (2006), Wei L., Xiaofeng M., and Weiyi M. (2009) makes use of the visual observation that, generally, data records are found at the centre of a web page, the centre as rendered by the web browser. This wrapper computes the visual centre of a HTML page and constructs a boundary region to encompass the data area. However, VSDR may overestimate the size of the boundary region and include non record data. Within the identified data record region VSDR attempts to identify individual records by fitting bounding boxes around text regions, again spacing is a key factor for this method to successfully extract the data records correctly. Any non record data, such as menu bars, advertisements, will also be identified as a data records, VSDR could potentially be improved with the addition of a filter.

Recently, Ontology Assisted Data Extraction (ODE) wrapper Weifeng S., Jiying W., and Frederick H. L. (2009) uses ontology technique to extract, align and annotate data from search engine results pages. However, ODE requires training data to generate the domain ontology. ODE is also only able to extract a specific type of data records (single section data records), thus it is not able to extract irregular data records such as multiple sections data records and loosely structured data records.

Proposed Methodology

Overview of RegionWrap

Our wrapper uses two stages extraction, global and local extraction (Figure 2). In the first stage, our wrapper will identify the list of regions based on their visual boundary. A region is considered as potential region if its visual boundary is of acceptable size (e.g. more than 500 pixels). Potential region is defined as region which may contain the relevant data, search results. Once the list of potential regions is identified, our wrapper chooses the region which has the largest size (calculated based on its visual boundary), since search results usually have the largest visual boundary (shown as dotted rectangles in (Figure 2).

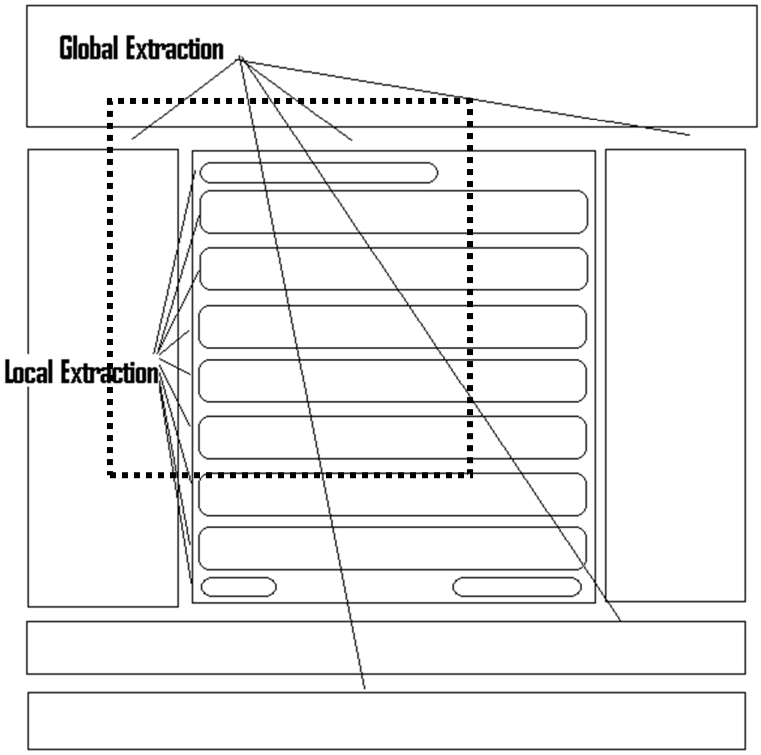


Figure 2: Global and Local Extraction Overview

When the largest region is identified, our wrapper will use local extraction stage to remove the remaining irrelevant data within the region such as search identifiers. As search identifiers usually have different visual boundary compared to data records, we use visual boundary of data records to remove these irrelevant data. Details of global and local extraction are presented below.

Global Region Extraction

First, we construct a DOM Tree and obtain the visual cue from the underlying browser rendering engine. Then we traverse the tree using a Breadth First Search (BFS) algorithm Yanhong Z. and Bing L. (2005). BFS is a graph search algorithm that begins its search at the root level and traverses all the remaining children nodes from left to right for each level, using a top down approach. For our case, we traverse the DOM Tree using BFS algorithm where nodes are considered either as HTML Tags or HTML Texts. Therefore, in this case, root is the <HTML> tag in the HTML page input source.

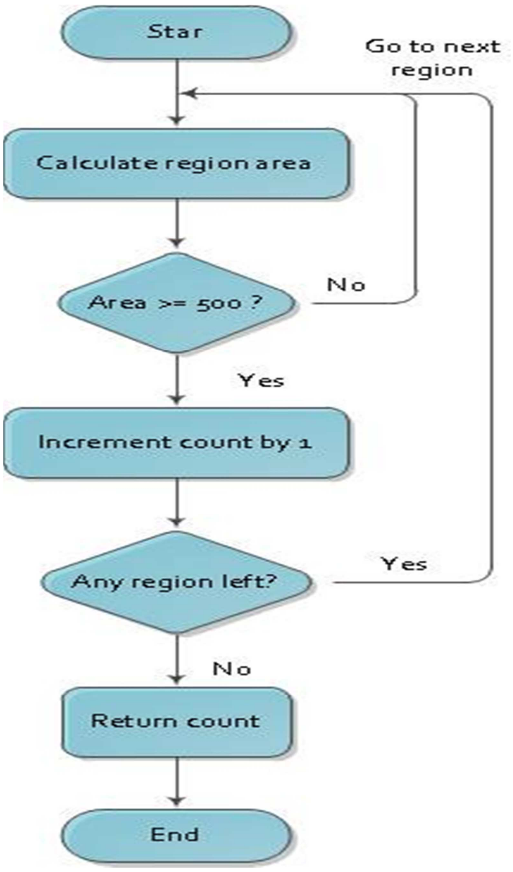


Figure 3: Flow Chart Depicting SearchRegion Function

To identify potential data regions, we implement searchRegion function which locates the list of regions based on the visual boundary of HTML Tags (Figure 3). searchRegion will use a BFS algorithm to match and identify the list of regions. Then, we implement searchRegion2 function which identifies the largest region from the list of regions (Figure 4). To locate and identify list of potential regions, our algorithm checks for at least three nodes in the same level which have visual boundaries greater than 500 pixels. Once these regions are identified, we store the nodes having visual boundaries greater than 500 pixels in a list. From this list, we choose the largest region and apply local extraction to remove the remaining irrelevant data in this region.

Local Extraction

In Local Extraction, our system checks for repetitive nodes in a particular level of a tree. These repetitive patterns are defined as group of data records. Groups of data records can be defined as a set of data records having similar parent HTML tag, containing repetitive sequence of HTML tags and are located in the same level of the DOM tree. Our wrapper uses the Adaptive Search extraction technique to determine and label potential tree nodes that represent data records. Subtrees which store data records may be contained in potential tree nodes. The nodes in the same level of a tree are checked to determine their similarity (whether they have the same contents). If none of the nodes can satisfy this criterion, the search will go one level lower and perform the search again on all the lower level nodes. Our method involves the detection of repetitive nodes which may contain data records and the rearrangement of these nodes to form groups of potential records in a list in 2 steps:

1. In a particular tree level, if there are more than 2 nodes and a particular

node occurs more than 2 times in this level, RegionWrap will treat it as a potential data record irrespective of the distance between the nodes.

2. These potential data records identified in this tree level are then grouped and stored in a list. The potential data records in this list are identified by the notation [$_{A1}$, $_{A2}$, ...$_{An}$] where $_{A1}$ denotes the position of a node in the potential data records where it first appears, $_{A2}$ is the position where the same node appears the second time and so on.

3. Figure 5 shows an example where nodes A, B, C are grouped and stored in list 1.

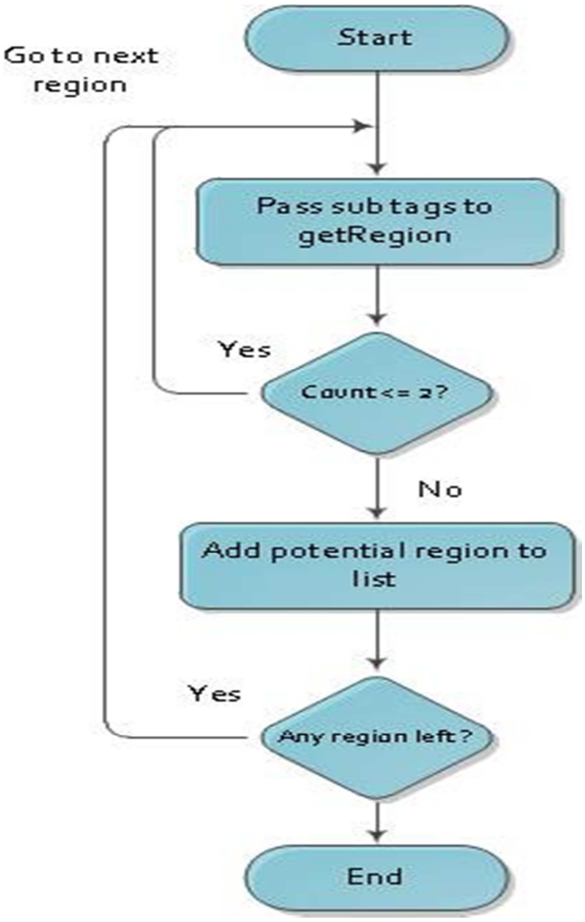


Figure 4: Flow Chart Depicting SearchRegion2 Function

After the groups of data records have been identified, we identify nodes which have similar visual boundaries, and nodes which do not have similar visual boundaries. Nodes with similar visual boundaries are grouped into one cluster while nodes which do not have similar visual boundaries are grouped into another cluster. To group them into clusters, we compare the visual boundary of the first node with that of second node. If the difference of the visual boundaries in the first node and second node is not huge (say less than 50 pixels), we put the first node into the first cluster (which contains nodes with similar visual boundaries). Otherwise, we put the first node into the second cluster (which contains nodes with dissimilar visual boundaries). The procedure is then repeated by taking the second node and third node for comparison until the last node is used. We retain only the cluster with nodes having similar visual boundaries and this cluster is known as the relevant region containing correct data records.

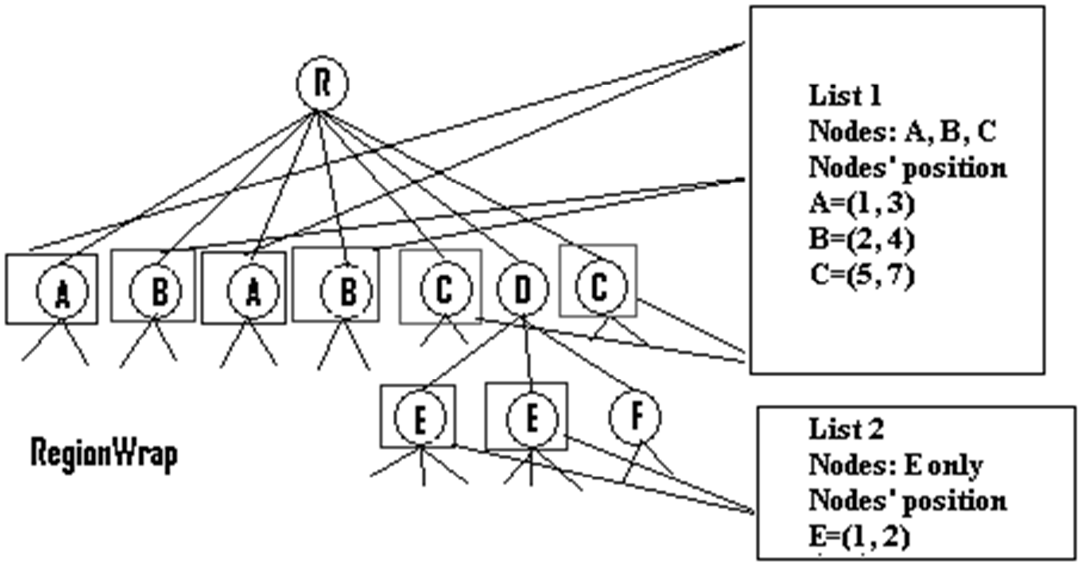


Figure 5: Potential Data Records in RegionWrap

Experiments

We collect 250 sample pages randomly from well known databases such as www.completeplanet.com for our experiment. The distribution of our sample varies, some of them are commercial based, governmental based and news related. The measures of wrapper's efficiency are based on three factors, the number of actual data records to be extracted, the number of extracted data records from the test cases, and the number of correct data records extracted from test cases. Based on these three values, precision and recall are calculated according to the formula:

*Recall=Correct/Actual*100*

*Precision=Correct/Extracted*100 (1)*

Table 1: Results of RegionWrap and WISH

Terms	Region Wrap	WISH Jer L. H., Eugene S., Simon E. (2010)
Actual	2476	2476
Extracted	2257	1914
Correct	2110	1671
Recall	85.21%	67.49%
Precision	93.49%	87.30%

We compare our wrapper with the work of WISH Jer L. H., Eugene S., Simon E. (2010) (Table 1). Experimental results show that our wrapper outperforms state-of-the-art wrapper WISH in both recall and precision rates. Our wrapper also runs at a speed of 450 milliseconds compared to WISH which runs at 150 milliseconds, which is important for real time applications. This is due to the fact that our wrapper extracts the relevant region using two stages operation, global and local extraction. In global extraction, our wrapper locates relevant region by identifying the largest region. Then our wrapper removes the remaining irrelevant data from the largest region. In WISH, they extract relevant region by identifying list of regions containing repetitive patterns. Once these regions are identified, WISH checks for the similarity of these regions based on their tree structures as relevant regions are assumed to contain patterns with similar tree structures. Then, WISH extracts the relevant region based on the assumption that relevant region contain the largest amount of texts and images. In our approach, we identify the relevant region first in global extraction, while removing the irrelevant data in local extraction. This approach could results in higher accuracy as we remove irrelevant regions at a global level, while removing irrelevant data at local level. It also helps to increase our speed performance as we do not check and match the entire regions as in WISH.

Conclusion

We develop a novel region based extractor RegionWrap which could accurately extract data records from search engine results page. Unlike existing wrappers, our wrapper extracts relevant information in two stages, global and local. In global extractor, we use visual boundary of data region to locate the relevant data region. Once relevant data region is found, we filter out the remaining irrelevant data within the region by using the individual data records' boundaries. Experimental results show that our wrapper outperforms state-of-the-art wrapper WISH in both recall and precision rates.

References

Arasu, A. & Garcia-Molina, H. (2003). "Extracting Structured Data from Web Pages," *ACM SIGMOD.*

Crescenzi, V., Mecca, G. & Merialdo, P. (2001). "Roadrunner: Towards Automatic Data Extraction from Large Web Sites," *ACM VLDB.*

Goh, P. L., Hong, J. L., Tan, E. X. & Goh, W. W. (2012). "Region Based Data Extraction," *IEEE FSKD.*

Hong, J. L. (2010). "Deep Web Data Extraction," IEEE International Conference on Systems, Man, and Cybernetics, *IEEE SMC.*

Hong, J. L. (2011). "Data Extraction for Deep Web Using Wordnet," *IEEE Transactions on Systems, Man and Cybernetics,* Part C: Applications and Reviews.

Hong, J. L., Siew, E. G. & Egerton, S. (2010). "Information Extraction for Search Engines Using Fast Heuristic Techniques," *DKE.*

Li, L., Liu, Y., Obregon, A. & Weatherston, M. (2007). "Visual Segmentation-Based Data Record Extraction from Web Documents," *IEEE IRI.*

Liu, B., Grossman, R. & Zhai, Y. (2003). "Mining Data Records in Web Pages," *In ACM SIGKDD.*

Liu, W., Meng, X. & Meng, W. (2006). "Vision-Based Web Data Records Extraction," *ACM Webdb.*

Liu, W., Meng, X. & Meng, W. (2009). "Vide: A Vision-Based Approach for Deep Web Data Extraction," *IEEE Transaction on Knowledge and Data Engineering.*

Simon, K. & Lausen, G. (2005). "ViPER: Augmenting Automatic Information Extraction with Visual Perceptions," *ACM CIKM.*

Su, W., Wang, J. & Lochovsky, F. H. (2009). "ODE: Ontology-Assisted Data Extraction," *ACM Transactions on Database Systems.*

Zhai, Y. & Liu, B. (2005). "Web Data Extraction Based on Partial Tree Alignment," *ACM WWW.*

Zhao, H., Meng, W., Wu, Z., Raghavan, V. & Yu, C. (2005). "Fully Automatic Wrapper Generation for Search Engines," *ACM WWW.*