Information Sciences 261 (2014) 132–148

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Leveraging spatial join for robust tuple extraction from web pages

Wook-Shin Han [a], Wooseong Kwak [b], Hwanjo Yu [c], Jeong-Hoon Lee [d], Min-Soo Kim [e,*]

[a] Department of Creative IT Engineering and Department of Computer Science and Engineering, POSTECH, Republic of Korea
[b] Korea Color Steel Corp., Republic of Korea
[c] Department of Computer Science and Engineering, POSTECH, Republic of Korea
[d] Department of Creative IT Engineering, POSTECH, Republic of Korea
[e] Department of Information and Communication Engineering, DGIST, Republic of Korea

ARTICLE INFO

Article history:
Received 8 October 2011
Received in revised form 10 May 2013
Accepted 3 September 2013
Available online 17 September 2013

Keywords:
Tuple extraction
Wrapper
Spatial join

ABSTRACT

Extracting tuples from HTML pages has been an important issue in various web applications. Commercial tuple extraction systems have enjoyed some success to extract tuples by regarding HTML pages as tree structures and exploiting XPath queries to find attributes of tuples in the HTML pages. However, such systems would be vulnerable to small changes on the web pages. In this paper, we propose a robust tuple extraction system which utilizes spatial relationships among elements rather than the XPath queries. Spatial information (e.g., 2-D coordinates) of elements are maintained in the DOM tree when a web page is rendered in a browser. Our system regards elements in the rendered page as spatial objects in the 2-D space and executes spatial joins to extract target elements. Since humans also identify an element in a web page by its relative spatial location, our system extracting elements by their spatial relationships could possibly be as robust as manual extraction. To specify and execute spatial joins, we propose a new query language, RAQuery, based on topological relationships between any spatial objects in the 2-D space. We then propose spatial join algorithms that efficiently process the RAQuery using novel notions of *group match* and *prunable relation group*. We next propose a tuple construction algorithm to build tuples from the extracted elements obtained by the spatial joins, which can construct tuples even when there are no boundary HTML elements specified for the tuples in the web page. Extensive experimental results using real HTML pages confirm that our solutions are far more robust than existing tuple extraction systems without sacrificing performance.

1. Introduction

Extracting tuples from HTML pages has been an important issue [9,12,14,17,19,20,25–29] in various web applications such as web data integration, e-commerce market monitoring, and mashups that repurpose and selectively combine existing web data and services [29]. After tuples are extracted from web pages, they can be easily transformed to different structures.

Commercial tuple extraction systems have enjoyed some success to extract tuples by regarding HTML pages as tree structures and exploiting XPath queries [36] to find attributes of tuples in the HTML pages. In such systems, given a sample HTML page *T*, the user first defines a target schema *S* for the tuples to extract and associates XPath queries of the HTML elements

[*] Corresponding author. Tel.: +82 10 9080 0174.
E-mail address: mskim@dgist.ac.kr (M.-S. Kim).

http://dx.doi.org/10.1016/j.ins.2013.09.027

(corresponding to the attributes of the tuples) in *T* to target elements of *S*. The systems store these associations as mapping rules for extracting tuples. Thus, these rules can be applied to all pages similar to *T*.

Commercial tuple extraction systems typically require two steps to extract tuples from a given web page: (1) selecting the boundary element of the first tuple to extract (assuming the remaining tuples can be extracted by changing the position information of the boundary element) and (2) selecting the corresponding tag for each attribute of the tuple. Fig. 1 shows the conventional tuple extraction process in detail. Suppose a user wants to extract names, ages, and cities of the people whose names are "David Dewitt" from the Yahoo's people search page. The user is first required to select the TR tag (boundary tag) of the *first* tuple. More specifically, an absolute XPath, /HTML/BASE/BODY/DIV[2]/TABLE/TBODY/TR[3], is used to locate the TR tag. Once the XPath for the first tuple is identified, the user needs to select tags for the attributes, i.e., names, ages, and cities. Specifically, three relative XPath queries from the TR tag—./TD[2]/DIV[1]/A[1]/B[1], ./TD[3], and ./TD[4]—are used to locate these attributes. The tuple extraction system will increase the index of the TR tag to extract the second tuple. That is, /HTML/BASE/BODY/DIV[2]/TABLE/TBODY/TR[4] will be used for extracting the second tuple.

The conventional extraction systems have two serious problems. (1) Since absolute/partial match XPath queries are used to extract tuples, they are vulnerable to small changes in web pages. For example, if the email column is added just before the age column, the systems using either the absolute or partial-match XPath could not extract ages and cities of the people. (2) The extraction systems assume there exist boundary elements for tuples. However, due to increasing popularity of CSS styles, columns of a tuple to extract can be in different subtrees using DIV and SPAN tags. For example, if elements are grouped by columns using DIV tags, not by rows, there are no boundary tags for tuples. Thus, a completely different mechanism is needed to resolve these problems.

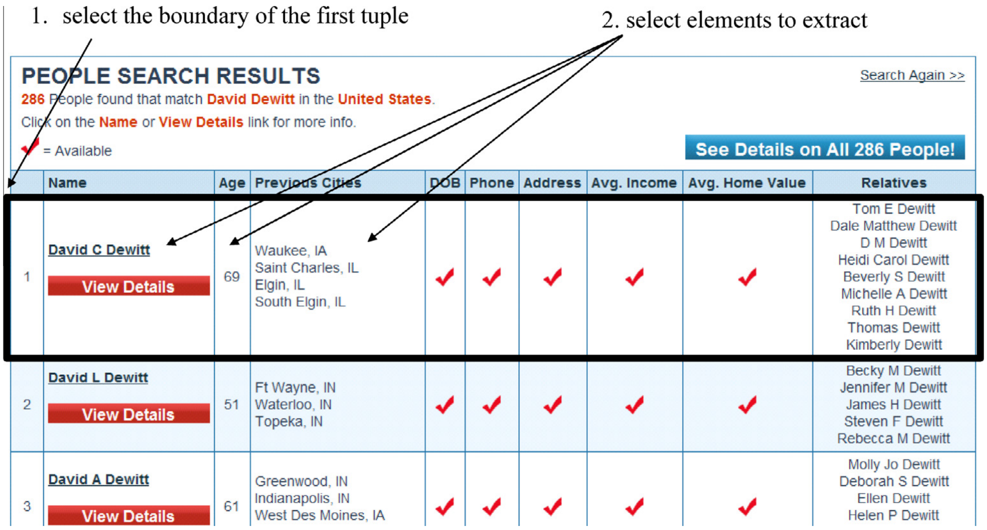
We propose a robust tuple extraction system that utilizes spatial relationships among elements rather than the XPath queries of the elements. Spatial information (e.g., 2-D coordinates) of elements can be obtained from the DOM tree when a web page is rendered in a browser. Our system regards elements in the rendered page as spatial objects in the 2-D space, and thus, executes spatial joins [38] to extract target elements. Since humans also identify an element in a web page by its relative spatial location, our system extracting elements by their spatial relationships could possibly be as robust as manual extraction. To the best of our knowledge, this is the first research to leverage spatial join to extract tuples from web pages.

To specify and execute spatial joins for tuple extraction from web pages, we propose RAQuery which is based on topological relationships between any two bounding boxes in the 2-D space. Since writing a complex RAQuery for each target attribute is tedious and error-prone, we also propose a high-level visual query composition tool called VisRAQuery, with which just a few number of mouse clicks and drags enable users to compose the complex RAQuery.

We then propose spatial join algorithms that efficiently process the RAQuery for tuple extraction. The algorithms use novel notions of *group match* and the *prunable relation group*. Since the spatial join for tuple extraction is a self-join over a DOM tree, we can combine all different scan operations into one scan using the group match. We exploit the prunable relation group to aggressively prune nodes which are to be disqualified during the join.

Once target elements are extracted, tuples must be constructed from them. We develop a tuple construction algorithm based on a novel notion of *maximal disjoint bounding tuple* along with sophisticated resolution techniques. With the maximal disjoint bounding tuple and the resolution techniques, we can construct tuples even when there are no boundary HTML elements specified.

Our contributions are summarized as follows: (1) We propose the first framework that leverages spatial join for robust tuple extraction from web pages. (2) We propose a query language RAQuery that supports various matches as well as


1. select the boundary of the first tuple
2. select elements to extract

PEOPLE SEARCH RESULTS

Search Again >>

286 People found that match David Dewitt in the United States.

Click on the Name or View Details link for more info.

✓ = Available

See Details on All 286 People!

	Name	Age	Previous Cities	DOB	Phone	Address	Avg. Income	Avg. Home Value	Relatives
1	David C Dewitt View Details	69	Waukee, IA Saint Charles, IL Elgin, IL South Elgin, IL	✓	✓	✓	✓	✓	Tom E Dewitt Dale Matthew Dewitt D M Dewitt Heidi Carol Dewitt Beverly S Dewitt Michelle A Dewitt Ruth H Dewitt Thomas Dewitt Kimberly Dewitt
2	David L Dewitt View Details	51	Ft Wayne, IN Waterloo, IN Topeka, IN	✓	✓	✓	✓	✓	Becky M Dewitt Jennifer M Dewitt James H Dewitt Steven F Dewitt Rebecca M Dewitt
3	David A Dewitt View Details	61	Greenwood, IN Indianapolis, IN West Des Moines, IA	✓	✓	✓	✓	✓	Molly Jo Dewitt Deborah S Dewitt Ellen Dewitt Helen P Dewitt

Fig. 1. An example page from the Yahoo's people search.