Extracting Web Information using Representation Patterns

Juan C. Roldán, Patricia Jiménez, and Rafael Corchuelo
{jcroldan, patriciajimenez, corchu}@us.es
University of Sevilla, ETSI Informática
Avda. Reina Mercedes, s/n
Sevilla E-41012, Spain

ABSTRACT

Feeding decision support systems with Web information typically requires sifting through an unwieldy amount of information that is available in human-friendly formats only. Our focus is on a scalable proposal to extract information from semi-structured documents in a structured format, with an emphasis on it being scalable and open. By semi-structured we mean that it must focus on information that is rendered using regular formats, not free text; by scalable, we mean that the system must require a minimum amount of human intervention and it must not be targeted to extracting information from a particular domain or web site; by open, we mean that it must extract as much useful information as possible and not be subject to any pre-defined data model. In the literature, there is only one open but not scalable proposal, since it requires human supervision on a per-domain basis. In this paper, we present a new proposal that relies on a number of heuristics to identify patterns that are typically used to represent the information in a web document. Our experimental results confirm that our proposal is very competitive in terms of effectiveness and efficiency.

KEYWORDS

Web information extraction, open information extraction, web representation patterns, semi-structured documents, scalability.

1 INTRODUCTION

In the Web, there are many semi-structured documents that provide information that might help humans make decisions. Information extraction techniques are required to analyse those documents and transform them into structured records that can be fed into typical information systems [4, 15, 20]. Nowadays, the web information extraction field is evolving towards developing proposals at web scale. A proposal is considered to work at web scale, when it requires as little user intervention as possible and can extract information from as many sites as possible.

Typical information extractors are machine-learnt from a set of web documents, known as learning set. Many proposals require the learning set to be gathered from a single web site and can work on documents from that site only; many also require the user to annotate the learning set with labels that endow the information to be extracted with semantics, known as supervised approaches. These supervised approaches are not scalable to the Web because they require too much user intervention to annotate the learning set. There are some unsupervised proposals that do not require the user to annotate the learning set [2, 5, 11, 18]; furthermore, they can be applied to a variety of sites and are not bound with a pre-defined data model. This makes them quite appealing to extract information at web scale, but they have an important drawback: they focus on extracting the information that varies from a document to the rest, but do not take their structure into account. In other words, they can extract attributes and group them into records without any semantics, but do not attempt to analyse the actual structure of the information so that the resulting records make sense. For instance, think of a typical item-description table in which the first column has labels that endow the values in the second column with semantics. The previous proposals discard the first column and extract the list of values in the second column without a hint label on whether they correspond to different attributes or a single multi-valued attribute; more than that: if an attribute is optional and makes the first column of a document different from the rest, they commonly mistake the labels for data that is typically returned as a list of values. Simply put: they focus on data and forget about their structure and the labels that endow them with semantics. The proposals in the field of Open Information Extraction attempt to solve the previous problems [7]. Unfortunately, few authors in this field have dealt with the problem in the context of semi-structured web documents.

Open Information Extraction is a research field that addresses the previous problems. An immense majority of proposals in this field focus on web documents in which the information to be extracted is rendered in natural language passages [1, 6, 21]; they typically require the text to be POS tagged in order to identify patterns from which the information of interest can be extracted.

To the best of our knowledge, there has been only one attempt to devise an open information extractor for semi-structured documents [3]. Unfortunately, this proposal requires human effort in order to generate an extractor for a new domain or language. It is not clear whether it is general enough to be applied to a whole domain, instead of a subset of sites from that domain. That is, the technique seems to be domain- or site-dependent, which prevents it from working at web-scale. Furthermore, it cannot be applied to documents with multiple records because it is only intended to extract attributes without a structure. It has trouble dealing with optional attributes, cannot deal with multi-valued ones and it requires attributes to be fully-contained within the context of a DOM node. Nodes that do not appear in at least 50% of the documents are discarded, which is a strong assumption, since there are cases in which relevant attributes appear in less than a half of the documents. Last, but not least, it is unclear whether the proposal is resilient to changes.

We bet on heuristic-based proposals because they are the closest to a truly web scalable proposal since they do not require any learning phases, data models or schemas, and labelling the information that they extract is getting easier thanks to automatic semantic typers [19]. In this paper, we present an unsupervised heuristic-based technique that extracts information building on common web representation patterns, which is a term that we use to refer to the typical regular templates used to render information in web documents, e.g., an attribute-value table, a variable list, HTML meta-information, and so on. Since the heuristics are used to extract the information instead of using a rule, a learning step is not required. Our technique does not require any human intervention since it extracts the information as it processes the web documents, which makes it appropriate to perform open information extraction from semi-structured documents at web scale. Furthermore, it can make a difference between data themselves and the labels used to endow them with semantics.

The rest of this paper is organised as follows: Section 2 describes the details of our proposal; Section 3 reports on the results of our experiments; finally, Section 4 concludes our work.

2 OUR PROPOSAL

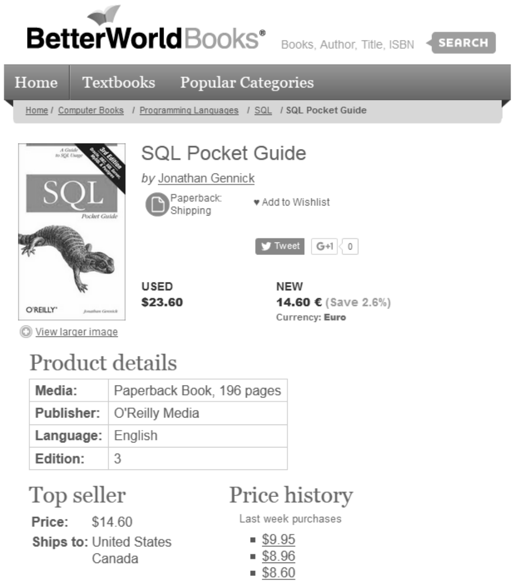
Our proposal works on sets of structurally-similar documents within a web site. It applies a number of heuristics that we have compiled building on our previous experience at developing several web information extractors [8–10, 12, 14, 16–18]. Each heuristic is intended to identify a representation pattern. Our heuristics have proven to be general enough so as to be applicable to a variety of different web sites since they focus on identifying representation patterns.

In this section, we first present some preliminary concepts and the heuristics themselves; we use the sample book store in Figure 1 to illustrate our description. We use the JSON format to represent the information that is extracted.

2.1 Preliminaries

Definition 2 .1. (Documents and nodes) Documents are character strings that adhere to the HTML syntax and can be represented as nodes. Given a node n, we denote its number of children as $|n|$, its tag as $tag(n)$, its HTML attributes as $attr(n)$, and its own text as $ownText(n)$. The tag path of a node is a path from the root of the document to that node.

For instance, Figure 2 depicts a simplified DOM tree from a web document. Given node n_3, it has $|n_3| = 2$, $tag(n_3) =$ "body", and $attr(n_3) = \{\text{onload} \mapsto \text{"RunTicker()''}\}$. The text of n_3 is "SQL Pocket Guide by Jonathan Gennick", which is obtained by transversing its children and concatenating their $ownText$ attributes. The tag path of n_6 is html/body/div/a.


BetterWorldBooks®

Books, Author, Title, ISBN

SEARCH

Home | Textbooks | Popular Categories

Home / Computer Books / Programming Languages / SQL / SQL Pocket Guide

SQL Pocket Guide

by Jonathan Gennick

Paperback: Shipping

♥ Add to Wishlist

Tweet G+1 0

USED
$23.60

NEW
14.60 € (Save 2.6%)
Currency: Euro

View larger image

Product details

Media:	Paperback Book, 196 pages
Publisher:	O'Reilly Media
Language:	English
Edition:	3

Top seller

Price:	$14.60
Ships to:	United States Canada

Price history

Last week purchases

- $9.95
- $8.96
- $8.60

Figure 1: A sample document.

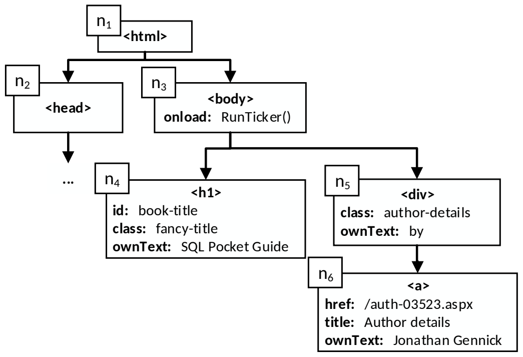
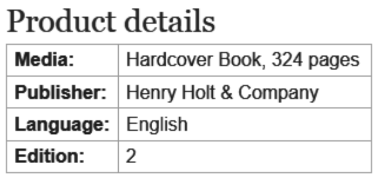


Figure 2: A simplified document DOM tree.

Definition 2.2. (Representation pattern) A representation pattern is the way in which the information to be shown in a document is formatted and rendered so that it is appealing for users. People are familiar with these patterns, which provide structural and behavioural common features across different sites. In our descriptions, terms "first" and "second" or "left" and "right" must be interpreted relative to the direction of writing that is specified in the corresponding document, i.e., left-to-right or right-to-left.

For instance, one of the most recurrent patterns is the key-value table. This pattern shows some properties of an item in a table. In Figure 3, the first column represents the names of the properties and the second one their values. Each row corresponds to a different property of the corresponding item.



Product details

Media:	Hardcover Book, 324 pages
Publisher:	Henry Holt & Company
Language:	English
Edition:	2

Figure 3: A sample key-value table.

Definition 2.3. (Heuristic) A heuristic is a method that helps identify and/or extract a representation pattern from a document.

For instance, to identify key-value tables one can fetch the tables with the same tag path and check whether the values in the first column are quite stable (the keys) and the values in the second column are not (the values).

Definition 2.4. (Records and attributes) A record instance is a piece of text that encapsulates the information regarding an entire item. Record instances might rely on sub-pieces of text with an atomic structure, namely, attribute instances.

For instance, when extracting information from the document shown in Figure 1, there is only one record instance, the book, which is identified as a set of attribute instances such as "Media", "Publisher", or "Language".

2.2 Extraction heuristics

Information nodes: We align nodes that have the same tag paths to identify the ones that are likely to carry some useful information, that is, those that contain information that varies from one document to another. These nodes are marked as information nodes, and their ancestors are marked as child-information nodes.

For instance, let us focus on node n_4 from our running example. We first align this node with the nodes that have the same tag path in other input documents. Since these nodes render the title of a book, it is very unlikely that they all are the same; so, this heuristic labels them all as information; this, in turn, results in the ancestor nodes being labelled as child-information.

Key-value tables: These tables present some properties of one or more items. The first column presents the names of the properties, the second column presents the values of the properties for the first item, the third column presents the second item, and so on (dealing with the transposed case is trivial). Since the input documents are structurally similar, we can collect every child-information table with the same tag path across different documents. Then, we can compute the variability ratio of the nodes of each column of the tables as n/k, where n denotes the total number of unique values, and k denotes the number of values. The interval of the variability ratio ranges from $(0.00 - 1.00]$, so that a value close to 0.00 means that the information does not vary from one node to another, and a value close to 1.00 means that the information of the nodes is highly variable. In a key-value table, the variability ratio of the first column must be smaller than the variability ratio of the other columns.

For instance, the key-value table in our running example has two columns. The variability ratio of the nodes of the first column of every table with the same tag path must be notably smaller than the second column, whose values change on every book. The table in our example is extracted as {"Media:": "Paperback Book, 196 pages", "Publisher:": "O'Reilly Media", "Language:": "English", "Edition:": "3"}. Note that it is very easy to remove the trailing colons to produce more natural field names.

Description lists: This structure is modelled using HTML description lists. This heuristic is similar to the previous one since a description list can be seen as a two-column table in which the terms constitute the first column and the descriptions the second column. So, it also relies on computing and checking variability ratios. The name of a property is identified using dt tags, and their values are identified using dd tags.

For instance, the description list in our running example is <dl> <dt> Price: </dt> <dd> $ 14.60 </dd> <dt> Ships to: </dt> <dd> United States </dd> <dd> Canada </dd> </dl>. It is extracted as {"Price:": "$ 14.60", "Ships to:": ["United States", "Canada"]}, identifying the second property as multivalued. Again, the trailing colons can be very easily removed.

Global information: There are some nodes that are used to provide global information of the document, such as title, h1, and meta nodes. They typically contain information about more than one property, so they are aligned across the documents and split according to the set of tokens that does not vary from document to document.

For instance, if we extract the title "SQL Pocket Guide by Jonathan Gennick - Abebooks.com" from our sample document and "The Call of Cthulhu by H. P. Lovecraft - Abebooks.com" from another document, the resulting extraction is ["SQL Pocket Guide", "Jonathan Gennick"] and "["The Call of Cthulhu", "H. P. Lovecraft"]" by splitting by the common tokens and removing them latter. In these case, the tag path is used as attribute name.

Breadcrumbs: A breadcrumb is a hint that helps users to keep track of their location within a site. The path consists of a sequence of hyperlinked words or short phrases that are separated by a glyph, plus a final word or short sentence that is not hyperlinked but is separated by the same glyph. We identify breadcrumbs by searching for such sequences and requiring the degree of variability of the components of the path to increase monotonically from left to right.

For instance, in the document shown in Figure 1, the breadcrumb is extracted as "breadcrumbs": ["Home", "Computer Books", "Programming Languages", "SQL", "SQL Pocket Guide"]. If we analyse more documents of the same site, we find that Home never changes, but the remaining components change more often as we move right wards.

Multi-valued lists: This structure is used to represent multiple values of a property, using HTML tags ol or ul, with values that varies across documents.

For instance, the list of prices of the example <ul> <li>$9.95</li> <li>$8.96 </li> <li>$8.60</li> </ul> is extracted as [$9.95, $8.96, $8.60].

Dataset	RoadRunner			FivaTech			Trinity			Our proposal		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Mean	**0.53**	**0.76**	**0.59**	**0.68**	**0.86**	**0.72**	**0.81**	**0.91**	**0.85**	**0.95**	**0.87**	**0.90**
Standard de iation	**0.21**	**0.28**	**0.24**	**0.26**	**0.14**	**0.25**	**0.10**	**0.08**	**0.07**	**0.07**	**0.14**	**0.08**
4 Jobs	0.10	1.00	0.18	0.87	0.60	0.71	0.82	0.90	0.86	0.98	0.82	0.90
6 Figure J bs	0.23	1.00	0.38	0.93	0.92	0.93	0.70	0.95	0.81	0.82	0.96	0.88
Job of Mine	0.61	0.66	0.63	0.53	0.52	0.53	0.67	0.99	0.80	1.00	1.00	1.00
Auto Trader	-	-	-	-	-	-	0.81	0.81	0.81	1.00	1.00	1.00
Car Zone	0.56	1.00	0.72	0.83	0.99	0.90	0.91	0.91	0.91	0.88	0.92	0.90
Classic Cars for Sale	0.36	0.46	0.40	-	-	-	0.61	0.84	0.71	0.90	0.86	0.88
Internet Autoguide	0.90	0.99	0.94	0.85	0.93	0.89	0.76	0.99	0.86	0.99	0.99	0.99
Aus Open Players	0.61	1.00	0.76	0.04	0.77	0.08	0.70	0.94	0.81	0.93	0.75	0.83
Haart	0.56	0.78	0.65	0.67	0.95	0.78	0.88	0.95	0.91	1.00	1.00	1.00
Web MD	0.22	0.04	0.07	0.54	0.95	0.69	0.96	0.94	0.95	0.96	0.64	0.77
Dr. Score	0.65	0.98	0.78	0.67	0.95	0.79	0.72	0.95	0.82	1.00	0.92	0.96
Linked In	0.38	0.49	0.43	0.78	0.87	0.83	0.89	0.86	0.87	1.00	0.54	0.70
RD Learning	0.73	1.00	0.85	0.86	0.74	0.80	0.75	0.94	0.83	0.72	0.95	0.82
Albania Movies	0.48	0.77	0.59	0.75	0.73	0.74	0.81	0.76	0.78	0.92	0.93	0.92
Abe Books	0.60	0.53	0.56	0.75	1.00	0.86	0.90	0.96	0.93	0.98	0.92	0.95
Awesome Books	0.70	0.43	0.54	0.80	0.96	0.87	0.91	0.86	0.88	1.00	0.65	0.79
Better World Books	-	-	-	0.91	0.93	0.92	0.71	0.70	0.70	1.00	0.91	0.95
Wate st nes	0.71	0.77	0.74	0.73	0.86	0.79	0.87	0.89	0.88	1.00	0.89	0.94
Player Profiles	-	-	-	0.08	0.93	0.14	0.81	1.00	0.89	1.00	0.71	0.83
Soccer Base	0.66	0.95	0.77	-	-	-	0.96	0.97	0.97	0.93	0.93	0.93

Table 1: Effectiveness results.

Measure	Sample ranking		Iman-Davenp rt	rgmann-Homm l
	Technique	Rank	P-Value	Rank
P	RoadRunne	3.71	$3.25\ 10^{-12}$	3
	FivaTech	2.76		2
	Trinity	2.41		2
	Our proposal	**1.12**		**1**
R	RoadRunner	2.65	0.69	1
	FivaTech	2.53		1
	Trinity	2.18		1
	Our prop sal	**2.65**		**1**
F_1	RoadRunner	3.53	$1.28\ 10^{-5}$	3
	FivaTech	2.71		2
	Trinity	2.18		2
	Our proposal	**1.59**		**1**

Table 2: Statistical analysis.

Remaining information nodes: After the previous heuristics are applied, some information nodes may remain. Although this remaining information is not structured according to a representation pattern, it might still be relevant for further analysis. They are extracted using their CSS selector as their field name and their own text as their values.

For instance, considering that span nodes were marked as information nodes after, then <div id="seller22"> <span>$14.60 </span> <div><div>Currency:</div> <span>Euro</span></div> is extracted as "general": {"#seller22 span1": "$14.60", "#seller22 div span2": "Euro"}.

3 EXPERIMENTAL RESULTS

Our experiments were performed on an computer with an Intel Core i7 3.50 GHz processor, 4GiB of RAM, Windows 7 Pro 64 bits, Java 1.8, and JSoup 1.8.3.

To evaluate our results, we compared our proposal with some unsupervised information extraction proposals that have been widely used in the literature for empirical comparison purposes [5, 11, 18]. We ran the information extractors on a validation dataset with 20 sites from the following domains: jobs, cars, real estate, doctors, conferences, movies, books, and sports. We randomly collected sets of 30 documents from each site.

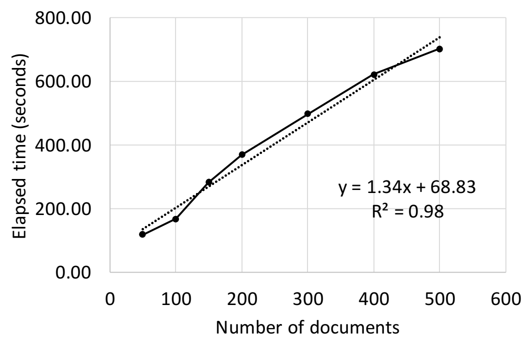


Figure 4: Performance results.

Table 1 reports on the effectiveness of the proposals that we have analysed. To compute precision, recall, and the F_1 score, we applied the proposals to the datasets, which were annotated for evaluation purposes only. The precision and recall of each site was computed as the average of the precision and recall of each value of an attribute that was extracted in each document from a site. The mean and standard deviation were also computed and they are shown at the top two rows.

Our first conclusion is that the precision of our proposal is better the other proposals. Despite the recall falls in some sites, the F_1 score stands out over the results of the other proposals. Since our proposal is based on a catalogue of heuristics, at the time of performing the experimentation, some representation patterns were not identified. We are currently working on devising additional heuristics to identify them so as to expand our catalogue and support other representation patterns.

To support that the differences that we found are statistically significant, we performed a statistical analysis [13]. We conducted a Shapiro-Wilk test on precision, recall, and the F_1 score at the standard significance level $\alpha = 0.05$ and we found that none of them behaves normally. As a conclusion, We then have to resort to non-parametric analysis techniques, namely: a) compute the rank of each proposal from the evaluation results; b) determine whether the differences in ranks are significant or not using the well known Iman-Davenport's test; c) if the differences are significant, compute the statistical ranking using Bergmann-Hommel's test on every pair of techniques. The results are shown in Table 2. Note that there are significant differences between precision and F_1 score, since the p-value does not exceed α so we can claim that we have not found any evidences to refute the hypothesis that our proposal ranks the first one regarding these measures. Despite our recall is smaller than the ones of the surveyed proposals, there is not a statistically significant difference regarding it, so there is no evidence that contradicts the hypothesis that our proposal is comparable to the others in terms of recall.

To measure the efficiency of our proposal, we applied it to a collection of 500 documents from the book domain. Starting with 50 documents, we increased the number of documents by 100 up to 500. The performance results are shown in Figure 4. Note that it performs linearly in the number of documents to be analysed.

In practice, our results suggest that our proposal can be used to extract information from semi-structured documents at web scale with good precision and recall, since it scales in a linear manner.

4 CONCLUSIONS

In this paper, we have presented an information extractor that is based on web representation patterns, which are used to display contents in a semi-structured manner in typical web documents. This proposal is unsupervised, web scalable and open, since it does not require any human supervision, it is not bound with any web site or domain, and it does not require a data model to extract the information in a structured way. Our experimental results confirm that our proposal is very promising regarding both effectiveness and efficiency.

During our experimentation, we found a number of additional web representation patterns for which we have to develop new heuristics in future, namely: improving the quality of the extractions by using pre-processing heuristics to polish, and homogenise the input documents, removing ads by using ad blocker filters and identifying structures which are not tables but are rendered as so.

REFERENCES

[1] E. Agichtein, L. Gravano, J. Pavel, V. Sokolova, and A. Voskoboynik. *Snowball: A prototype system for extracting relations from large text collections*. In *SIGMOD Conference*, page 612, 2001.

[2] A. Arasu and H. Garcia-Molina. *Extracting structured data from web pages*. In *SIGMOD Conference*, pages 337–348, 2003.

[3] A. Carlson and C. Schafer. *Bootstrapping information extraction from semi-structured web pages*. In *ECML/PKDD (1)*, pages 195–210, 2008.

[4] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan. *A survey of web information extraction systems*. *IEEE Trans. Knowl. Data Eng.*, 18(10):1411–1428, 2006.

[5] V. Crescenzi, G. Mecca, and P. Merialdo. *Roadrunner: automatic data extraction from data-intensive web sites*. In *SIGMOD Conference*, page 624, 2002.

[6] O. Etzioni, M. J. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. *Web-scale information extraction in KnowItAll: (preliminary results)*. In *WWW*, pages 100–110, 2004.

[7] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. *Open information extraction: The second generation*. In *IJCAI*, pages 3–10, 2011.

[8] P. Jiménez and R. Corchuelo. *On learning web information extraction rules with TANGO*. *Inf. Syst.*, 62:74–103, 2016.

[9] P. Jiménez and R. Corchuelo. *Roller: a novel approach to web information extraction*. *Knowl. Inf. Syst.*, 49(1):197–241, 2016.

[10] P. Jiménez, R. Corchuelo, and H. A. Sleiman. *ARIEX: Automated ranking of information extractors*. *Knowl.-Based Syst.*, 93:84–108, 2016.

[11] M. Kayed and C.-H. Chang. *FiVaTech: Page-level web data extraction from template pages*. *IEEE Trans. Knowl. Data Eng.*, 22(2):249–263, 2010.

[12] A. M. Reina, P. Jiménez, and R. Corchuelo. *A novel approach to web information extraction*. In *BIS*, pages 152–161, 2015.

[13] D. J. Sheskin. *Handbook of parametric and nonparametric statistical procedures*. Chapman & Hall/CRC, edition 4, 2007.

[14] H. A. Sleiman and R. Corchuelo. *An information extraction framework*. In *PAAMS (Workshops)*, pages 149–156, 2012.

[15] H. A. Sleiman and R. Corchuelo. *A survey on region extractors from web documents*. *IEEE Trans. Knowl. Data Eng.*, 25(9):1960–1981, 2013.

[16] H. A. Sleiman and R. Corchuelo. *TEX: An efficient and effective unsupervised web information extractor*. *Knowl.-Based Syst.*, 39:109–123, 2013.

[17] H. A. Sleiman and R. Corchuelo. *A class of neural-network-based transducers for web information extraction*. *Neurocomputing*, 135:61–68, 2014.

[18] H. A. Sleiman and R. Corchuelo. *Trinity: On using trinary trees for unsupervised web data extraction*. *IEEE Trans. Knowl. Data Eng.*, 26(6):1544–1556, 2014.

[19] M. Taheriyan, C. A. Knoblock, P. A. Szekely, and J. L. Ambite. *Learning the semantics of structured data sources*. *J. Web Sem.*, 37:152–169, 2016.

[20] J. Turmo, A. Ageno, and N. Català. *Adaptive information extraction*. *ACM Comput. Surv.*, 38(2), 2006.

[21] A. Yates, M. Banko, M. Broadhead, M. J. Cafarella, O. Etzioni, and S. Soderland. *TextRunner: Open information extraction on the Web*. In *HLT-NAACL (Demonstrations)*, pages 25–26, 2007.

ACKNOWLEDGEMENTS

Supported by FEDER funds through the Spanish R&D&I programme (grants TIN2016-75394-R and TIN2013-40848-R).