IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 18, 2021, accepted August 9, 2021, date of publication August 13, 2021, date of current version August 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3104524

WebQuIn-LD: A Method of Integrating Web Query Interfaces Based on Linked Data

JULIO HERNANDEZ[1], HEIDY M. MARIN-CASTRO[2], AND MIGUEL MORALES-SANDOVAL[1]
[1]Unidad Tamaulipas, Centro de Investigación y de Estudios Avanzados, Victoria, Tamaulipas 87130, México
[2]Facultad de Ingeniería y Ciencias, Cátedras-CONACYT, Universidad Autónoma de Tamaulipas, Victoria, Tamaulipas 87000, México

Corresponding author: Heidy M. Marin-Castro (heidy.marinc@gmail.com)

This work was supported by the Fondo Sectorial de Investigación para la Educación, Ciencia Básica Secretaria de Educación Pública–Consejo Nacional de Ciencia y Tecnología (CB SEP-CONACyT), Mexico, under Project 281565.

ABSTRACT The deep web is a huge source of domain-specific information (sale of houses, medical information, e-commerce, science, etc) stored in database servers accessible through HTML forms called web query interfaces (WQIs). Information in the deep web is retrieved by querying one database server at a time, which results inefficient. A more attractive approach is to create an integrated WQI (IWQI) that acts as single entry point to query several database servers at a time for a given domain. Schema matching and string (labels in WQIs) comparison have been the most popular techniques to create IWQIs. In this work, we propose a new method for the integration of web forms based on linked data and the VDIS (View-based Data Integration System) architecture. We present WebQuIn-LD, an alternative and novel approach relying on linked data principles to combine individual WQIs into a single IWQI for a given domain is presented. WebQuIn-LD follows a data integration system architecture, starting from the wrapping of domain-specific WQIs until the creation of the IWQI. A domain-independent ontology is created to describe WQI elements as linked data resources and to exploit semantic integration between the WQI's elements. WebQuIn-LD was evaluated on performance metrics (precision, recall, and F1) using the state-of-the-art WQIs datasets for different domains (airfares, books, autos, jobs, music, movies, hotels, jobs). The obtained results demonstrate the effectiveness of the linked data approach presented in this work for the WQI integration problem.

INDEX TERMS Web query interface, automatic integration, semantic web, linked data, deep web.

I. INTRODUCTION

The deep web is a dynamically generated Web whose content is retrieved from different data sources such as databases or file systems [1], only accessible through web query interfaces (WQIs). A WQI [2] is an HTML form out of reach of conventional search engines, such as Google or Bing [3], [4]. The deep web was estimated to be at least 500 times larger than the surface Web, and it continues to grow at an accelerated rate [5].

A WQI has an intermediary role between an end-user and a deep web database. Initially, a user submits a query to the WQI, considering the semantics of each WQI element, the metadata of each element, and how the WQI is organized [4]. Since there are several WQIs for a given domain (e.g., e-commerce), information retrieval from the deep web requires filling in each WQI, sending the query, gathering, inspecting and integrating the results. Thus, information retrieval in the deep web is a very time-consuming task. To alleviate that problem, an integrated WQI represented a set of related WQIs in a given domain could receive the query and automatically transform and submit query conditions into each individual WQIs [6] (one for each database server).

According to Furche *et al.* [7], the automatic processing of WQIs to create a single integrated WQI in a given domain presents the following challenges:

- The different type of WQI's elements (text box, check box, radio button, etc.) used to represent the same concept in the domain. For example, in the flights domain, the *Age* label can be represented by a textbox or by a selection button.
- The different labels to describe the same concept in different WQIs. For example, *City*, *Town*, or *Address* can all them refer to *Location*.
- Existence of ambiguous search criteria. For example, *Tenure* might refer to the choice of buying or renting something.

The associate editor coordinating the review of this manuscript and approving it for publication was Mansoor Ahmed.

The WQI integration problem has been approached by different authors mainly considering the schema matching technique [2], [8], [9]. These works consider the following interface matching problem: given a large set of sources in a domain, find semantic correspondences called mappings between the attributes of the query interfaces of the sources [10], [11]. Interface matching plays an important role in the integration of WQIs, involving three major tasks [6]: interface modeling, schema extraction, and schema matching.

A. INTERFACE MODELING

A WQI typically consists of multiple attributes, where related attributes are placed near each other, forming a group. Closely related attribute groups may be further grouped into a supergroup. Attributes and attribute groups are intuitively ordered into the WQIs.

B. SCHEMA EXTRACTION

A WQI is typically rendered as an HTML form. The form is concerned with the visual representation of the attributes. However, the form does not explicitly specify the attribute-label and attribute-attribute relationships in the WQI. The structural aspect of the WQI needs to be inferred from its visual representation via schema extraction.

C. SCHEMA MATCHING

Given a set of WQI schemes, it is necessary to accurately determine the mappings of attributes among the different WQIs. There are two types of mappings: simple and complex. A simple mapping is a 1:1 semantic correspondence between two attributes. A complex mappings, e.g., 1-*m* mappings, is a mapping where an attribute in a WQI semantically corresponds to multiple attributes in another WQI.

This paper presents a method named *WebQuIn-LD* that relies on linked data technologies for creating Integrated Web Query Interfaces. Linked data [12] is a way to publish structured data based on the fundamentals of the World Wide Web. It makes use of open W3C standards such as RDF and SPARQL [13] to construct meaningful information.

WebQuIn-LD is a method that facilitates WQI integration, flexible to be applied to any domain. The idea of using a linked data approach is to achieve an easy exploration, retrieval and comparison of labels extracted from each web form. This approach allows the analysis of content in WQIs through a structured format and then to achieve a better identification of specific labels such as dates or ranges. SPARQL language was used for retrieving the largest possible number of elements contained by each form (text boxes, drop-down, check box, among others).

WebQuIn-LD is based on a domain-independent ontology, that is, an ontology suitable for any domain of interest, including the exploitation and use of semantic information by means of linked data technologies. This ontology identifies the general elements of a WQI. The proposed ontology considers a set of general rules to identify WQIs elements such as radio buttons, text box, drop-down list, etc., and map them to elements in a structured format based on W3C standards. The advantage to represent a WQI as a linked data structure is the discovery of relationships between elements from different WQIs based on string (label of a WQI's element) similarity which favors the integration process. The linked data structure can be queried by means of the SPARQL language, which is a standardized query language for linked data.

The main contributions of this work are listed below.

- A new WQI integration method based on linked data technologies and on the VDIS (View-based Data Integration System) architecture.
- A domain-independent ontology to map a WQI schema.
- A novel integration measure to determine the relevance of each element in individual WQIs to the integrated WQI.

WebQuIn-LD was validated and evaluated using well known datasets for the WQI integration problem, the ICQ and Tel-8 datasets for the domains of airfare, automobile, book, job, hotel, movies, music, and car rental. The obtained results, particularly for the label selection evaluation are competitive with the state-of-the-art works, which demonstrate the effectiveness of the linked data solution approach for the WQI integration problem.

The rest of the document is divided as follows. Section 2 presents the related work. Section 3 details the design of WebQuIn-LD. Section 4 explains the experiments and results obtained from the performance evaluation of WebQuIn-LD. Finally, Section 5 presents the conclusion of this work and outlines future work directions.

II. RELATED WORK

The approaches in the state-of-the-art for integrating domain-specific WQIs [2], [4], [7]–[9] are mainly based on schema extraction, by serializing the WQIs content in a structured data format such as XML. The schema extraction process traverses all the elements from the different WQIs to determine their similarities by means of mapping rules.

The WQIs integration problem has been generally tackled by describing the features of each WQI element, e.g., label, name, hierarchy position, etc., and then finding similarities of such element to elements in other WQIs. Wang *et al.* [15] proposed a domain ontology construction to describe the most representative WQI elements and a schema matching process to integrate them. Wu *et al.* [6] proposed a schema extraction system called ExQ to integrate WQIs. In both approaches, the integration task is based on a schema matching process. While ExQ relies on visual representation, the proposal by Wang *et al.* relies on building a domain ontology. Unlike these approaches, WebQuIn-LD defines a domain-independent ontology to describe the WQI elements. For the integration process, WebQuIn-LD uses linked data technologies instead of schema matching.

Another approaches [4], [7], [14] parse and/or classify WQIs to integrate their elements based on statistical models or machine learning algorithms. DeepPeep [4] is a web search

TABLE 1. Related works in the literature for the WQIs integration problem.

Year	Ref.	Label identification	Integration	Schema serialization	Dataset	Evaluation measure
2019	Chichang Jou [2]	Yes	No	XML	ICQ, TEL-8	F-measure
2018	Marin et al. [8]	Yes	Yes	XML	Crawled	Precision, Integrity, and Efficiency
2016	Chichang Jou [9]	Yes	No	XML	WQIs from dmoz.org	Correctness, Recall, and Precission
2013	Furche et al. [7]	Yes	Yes	N/A	ICQ, TEL-8	Precision, Recall, and F-measure
2013	Sue et al. [14]	Yes	Yes	N/A	ICQ, TEL-8, WISE, CNW	F-score
2010	Barbosa et al. [4]	Yes	No	N/A	TEL-8	N/A
2009	Wue et al. [6]	Yes	Yes	N/A	ICQ, TEL-8	Precision, Recall, and F-measure
2021	This work	Yes	Yes	RDF	ICQ, TEL-8	Precision, Recall, and F-measure

engine to discover WQIs from the deep web, web databases and web services. However, it only identifies and classifies WQIs, the integration is not considered. StatParser [14] is a query interface parser based on the maximum-entropy principle to learn from parsed WQIs. StatParser starts with a small set of parsed WQIs to create a statistical model which increases its size by parsing new WQIs. OPAL [7] is based on a domain ontology to derive a WQI schema. It combines structural, textual and visual features to map labels to elements. The integration of WQIs is based on a domain schema classification.

Jou [2], [9] proposed a schema matching and merging based on string similarity and synonyms of labels [9] to merge elements from different WQIs. In [2], a schema extraction method is proposed based on heuristic rules, considering their previous work. VSearch [8] is a vertical search tool for the deep web that crawls WQIs from the web to create a local repository used to classify and integrate WQIs.

WebQuIn-LD is the first approach that uses linked data technologies to construct a strategy for addressing the WQI integration problem. It uses string similarity to associate related WQI elements and merges the rules defined in [2] and in [8] to identify and extract element's label and element's features, respectively. Table 1 summarizes the most relevant approaches to the WQI integration problem.

III. METHODOLOGY

WebQuIn-LD follows a pipeline process inspired in a view-based data integration system (VDIS) architecture (see Fig. 1). VDIS was inspired by multi-databases and federated systems [16], providing the grounds to study the problem of specifying the correspondence between the sources and the unified view. VDIS provides a single point of access to all heterogeneous data sources. It is based on four modules: i) source, ii) wrapper, iii) mediator and iv) applications. The first module (source) is constituted by structured data (XML files, web pages and data bases) and unstructured data (text file). The second module (wrapper) maps source's content into a schema representation, solving the heterogeneity from each source. The local schema information is used to generate a mediator (third module) which represents a global version of the wrapped data. The last module (applications) represents the mediator's applications. A query to the unified view (mediator) retrieves integrated results from all sources and shows them in the application module.

WebQuIn-LD is based on the first three modules of the VDIS architecture: source, wrapper and mediator. The process begins (source module) by collecting the source files, that is, the HTML files representing a set of heterogeneous WQIs for a given domain. The wrapping module maps (1-1) a WQI into an Ontology Representation (OR-WQI), where each WQI element (input text, radio button, etc.) is described as a linked data resource. The mediator module integrates the information provided by each OR-WQI, resulting in an integrated WQI (IWQI). The integration process is based on a Global As View (GAV) approach (see Fig. 2). The global schema, represented by the GAV approach, is described in terms of the local information from each OR-WQI.

The steps followed by the wrapper and mediator are explained in the following sections.

A. WRAPPER MODULE

The wrapper module identifies and extracts all the WQI's elements with their corresponding labels. These elements are described as a linked data resource. The extraction process follows the rules proposed by Castro *et al.* [8]. The authors

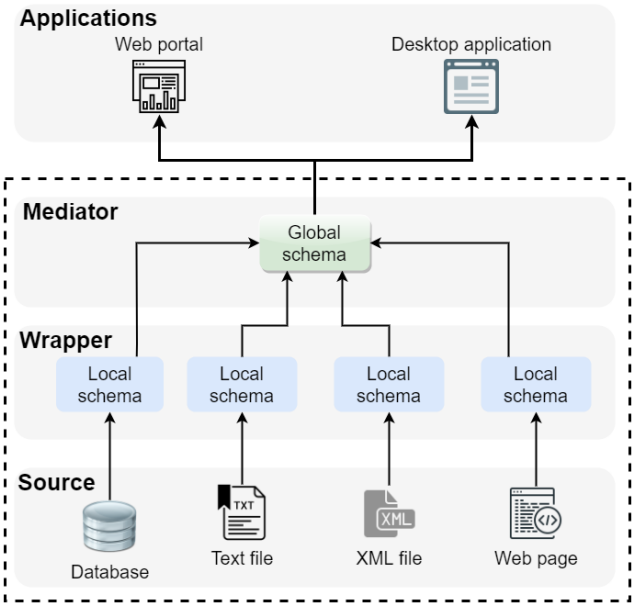

FIGURE 1. View-based data integration system (VDIS) architecture.

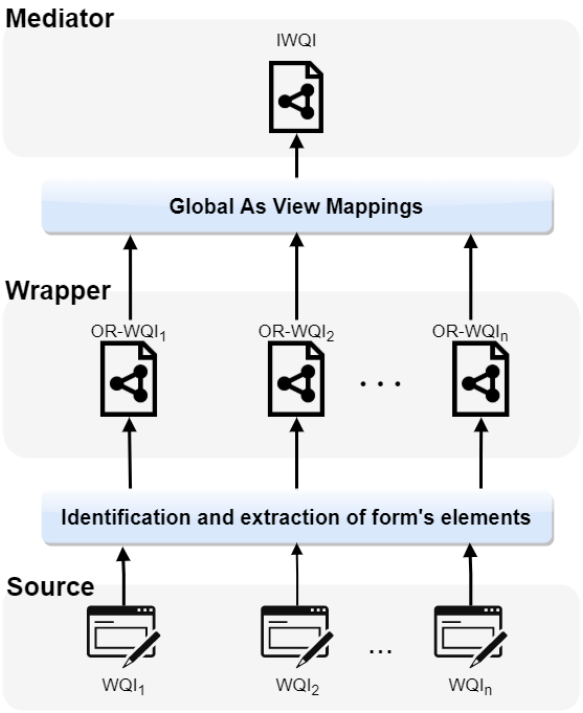

FIGURE 2. WebQuIn-LD, a view-based architecture.

defined a field (radio button, checkbox, text input box, selection list) as the basic semantic unit in a WQI, and how these elements are organized and the position of the element's label in the WQI.

The wrapper module is divided in two mapping tasks: i) HTML-JSON and ii) JSON-RDF. The first mapping task persists the structure of a WQI into JSON format (structured format). The second mapping task transforms the JSON format into a semantic representation RDF.

1) FROM HTML TO JSON

In general, a WQI is composed of several fields (see Fig. 3) to retrieve information from a domain. Each field describes a strategic piece of information to query the web database.

The WQI's fields are mapped to a JSON structure considering the following attributes (see Fig. 4):

- *elementId*: it is an integer number to define the position of an element in the WQI (right to left and top to bottom as defined in [8]).
- *elementLabel*: corresponds to the label associated with a WQI's element (text box, radio button, check box, etc.).
- *name*: corresponds to the name associated with an attribute, which is defined by the WQI designer.
- *value*: corresponds to the value associated with the WQI's element. Generally, a text box does not contain a value but a radio or checkbox element has an associated string value.
- *isChecked*: defines which radio or check box value is set as predefined. Usually, the element's value (radio or checkbox) narrows the information provided in a text box field.
- *type*: defines if the WQI's element is an input or a select list element.
- *elementType*: defines if the WQI's element is a text box, a radio, a check box, an image, etc.

Additionally, the provenance information from the source of the WQI such as page title, source code, and the issued date is part of the JSON structure. Once the WQI's elements information is collected and serialized as a JSON schema, the next step adds semantics to the data by mapping the JSON data into a RDF representation.

2) FROM JSON TO RDF

The HTML to JSON mapping process persists the WQI elements as a set of key-value data. The JSON structure is considered as a middle step to persist WQIs as a structured data, describing each element's feature as an isolated key-value information. WebQuIn-LD defines a semantic representation to describe the WQI's elements as resources based on a domain-independent ontology (see Fig. 5). The semantic representation is based on a linked data structure and a query mechanism. The state-of-the-art approaches [2], [8], [9] are focused on mapping WQI's content as XML format, thus providing a limited semantic representation. WebQuIn-LD is based on a rich semantic representation where WQI's elements can be linked by means of semantic relations.

In the WebQuIn-LD semantic representation, each WQI element is a resource. A resource is defined by a unique URI and described by a set of properties. For example, the class *Element* (in Fig. 5) is defined by the prefix

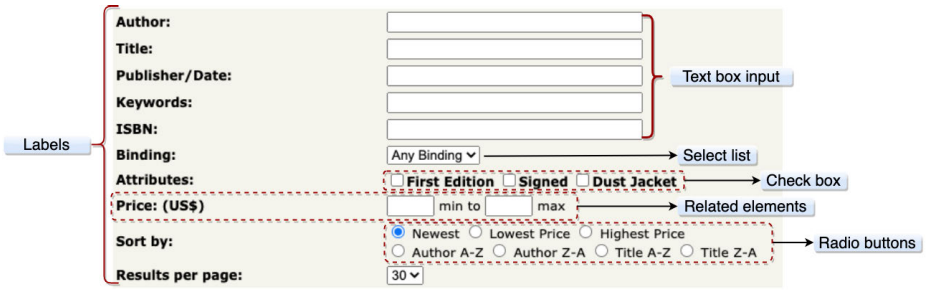

FIGURE 3. Example of a WQI in the 'Books' domain.

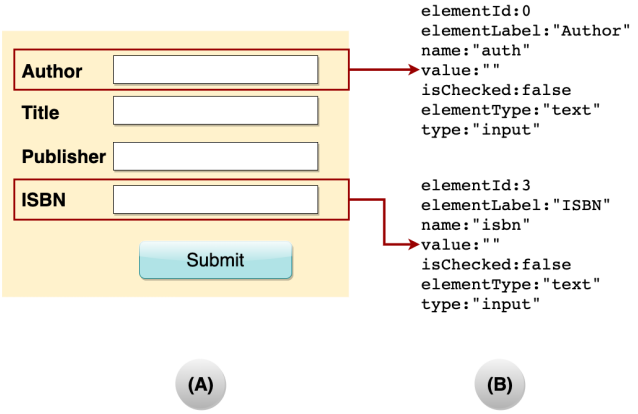

FIGURE 4. Mapping example of HTML components in a WQI (A) to JSON (B).

wqi:Element with seven associated properties. The proposed domain-independent ontology is constituted by the following classes:

- *Element*: The main class. It defines the general properties of a WQI's element such as attribute name, label and type.
- *Input*: It defines an input element by means of three properties: 1) `_:attrValue` defines the element's value, e.g., the label of a checkbox; 2) `_:isChecked` applies only for radio and checkbox elements to denote a default value; 3) `_:parentLabel` defines a group label, e.g., a group label to describe the year, month and day.
- *Text, Radio, Check box, Hidden and Submit*: Sub-classes of the Input class.
- *Select*: It defines a drop-down list of options.
- *Option*: It is a sub-class of *Select*. It defines an option of the drop-down list.
- *HTML*: This class describes the title and URI of the WQI's web page.
- *Form*: It describes the general information of the WQI such as the submission string, the name of the WQI, the method (POST or GET) and its ID.
- *Provenance*: It defines the source of the WQI, e.g., an URI or an HTML file. Additionally, it defines the issued date and the provenance ID.

For example, Fig. 6 shows a WQI (part A) and how the first input text element is described by means of a RDF subject (part B), predicates and objects (part C), following the linked data rules. Fig. 7 shows a graph representation for the WQI.

The mapping processes (HTML to JSON and JSON to RDF) define the wrapper module of the integration architecture to describe all WQIs as linked data resources. This representation enables the integration of all WQIs through SPARQL queries based on the type of each element, their associated label and their provenance information.

B. MEDIATOR

The mediator step integrates the WQIs based on the GAV approach [17], [18]. GAV defines a global schema in terms of the elements from the local schemas (wrappers), described by Eq. 1.

$$V_i \rightarrow I(R_i) \tag{1}$$

where V_i is the view (query) over the combined schemes of the sources. R_i is a relation in the global schema and $I(R_i)$ is the identity query over R_i, i.e., a query that returns all the attributes of R_i.

In this work, the constructed queries are SPARQL queries over RDF files. $I(R_i)$ is the join of all the provenance information instead of a join of tables, as in the original definition of GAV.

The integration process is based on label matching, which has been already used in the WQI problem. However, in this work label matching is not a critical part in the solution, but a tool that is benefited from the inclusion of a preprocessing strategy (not previously used) that discovers relevant nouns. This preprocessing is useful because the label matching effort is reduced. Label matching is also improved by including SPARQL to help discover labels with particular meaning, as for example dates (from-to) or ranges (before-after). Furthermore, as pointed out by Nguyen *et al.* [19], labels can be placed in many different positions in relation to their associated elements due to miscellaneous design methodology. In [2], author mentioned that the content of a label could be different for each WQI, containing explanatory information to describe the required information, e.g., *'author name'* label in WQI_1 and *'insert the author's name'* label in WQI_2. Also, six rules are defined in [2] to extract an element's label. In this

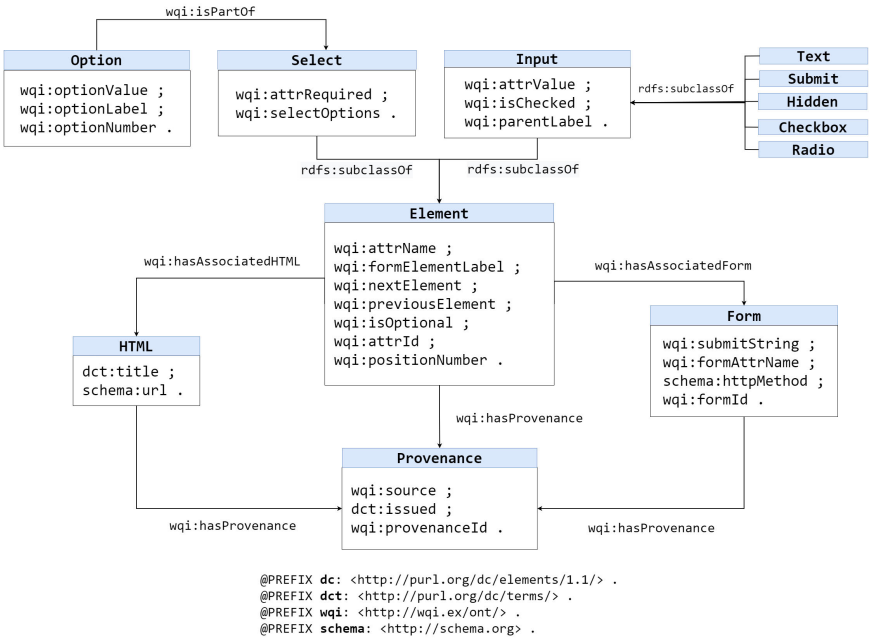

FIGURE 5. The WebQuIn-LD schema.

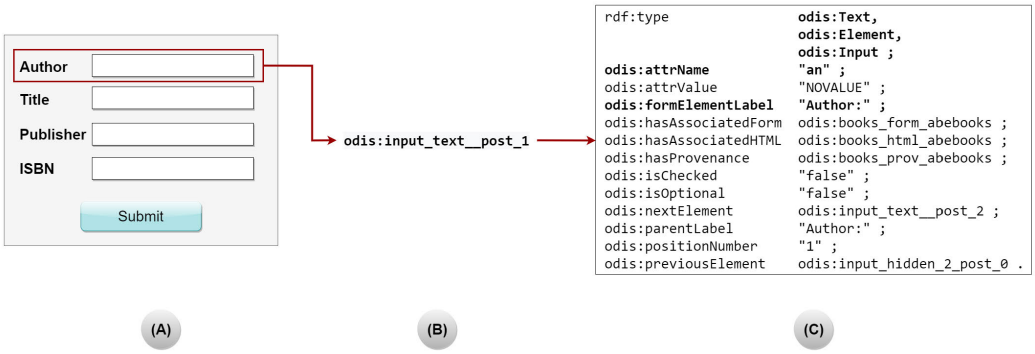

FIGURE 6. (A) WQI from the book domain. (B) The URL to identify the input text element (RDF subject). (C) Attributes of the input text element (RDF predicates and objects).

work, label extraction follows the steps proposed in [2]. Also, a label matching step is used, which selects the most relevant noun in a label. The noun selection step is as follows:

1) Pre-processing: Features such as part-of speech, tokenization and lemmatization are extracted for each label.
2) Sorting: Original labels are sorted by length.
3) Selection: A gold label (constituted by only one noun) is selected.
4) Clustering: A cosine similarity measure is applied over gold labels and noun label's tokens.
5) Rewriting: If a label is miswritten, it is rewritten with a gold label based on a cosine similarity measure.

The pre-possessing step defines the label's features. The sorting step arranges the labels based on their length (number of words). The selection step defines a set of descriptive labels. The clustering step groups labels with similar content based on the set of gold labels. The final step, rewriting, helps to fix any word misspelling based on the cosine similarity.

The cosine similarity measure is extensively used in the state-of-the-art [20], [21] to determine if two strings are similar. In this work, it is defined a threshold of 0.8 based on a basic experimentation to determine the similarity between two strings. The experiment compares a noun against the same noun without the first or last letter and without the first or last two letters. For one deleted letter, the results are closed to 0.9 and for two deleted letters, the results were closed to 0.8. The compared nouns were lemmatized and lowercased to provide a fair comparison between them.

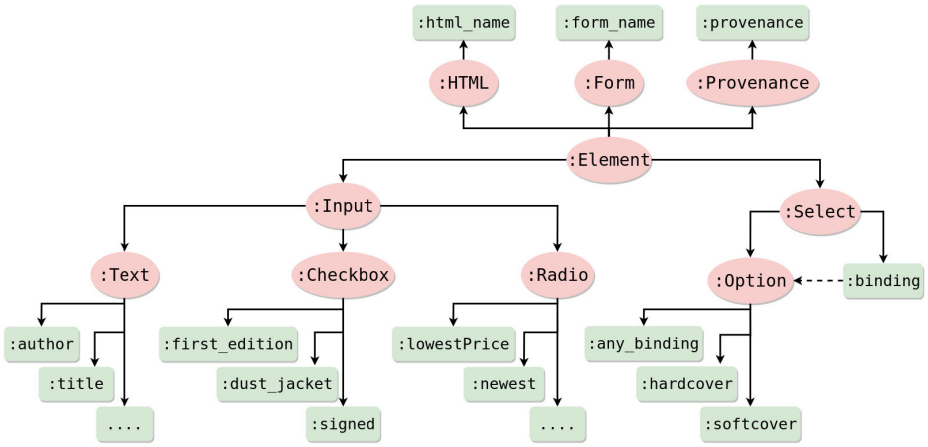

FIGURE 7. Graph representation of a WQI in Fig. 6.

The noun selection process is relevant for most of the WQIs. However, as it is pointed in [2], there are labels which denote a special meaning, e.g., dates, adverbs like *before* and *after* to define ranges. To denote this particular situation, a range rule is defined as follows:

1) RANGE RULE

If a label denoting beginning of a range is discovered in the set of labels from the WQI, then a label denoting the ending of that range must be part of the same set.

An advantage of representing a WQI as a linked data structure is the possibility to search any WQI's element in the structure. A SPARQL query is defined to apply the range rule (Listing 1).

```
PREFIX wqi: <http://wqi.ex/ont/ns#>

SELECT ?startRangeLabel WHERE {
 ?s a ?type ;
    wqi:formElementLabel ?
       ↪ startRangeLabel.
 FILTER (?type in (wqi:Input,wqi:Select)
    ↪ )
 FILTER (?startRangeLabel in ("from","
    ↪ before"))
}
```

LISTING 1. Query to search for a range start label.

The `startRangeLabel` variable will be empty if no a beginning of range label is found in the WQI's labels set. If the variable is not empty, the presence of the complementary label is validated through a new query (Listing 2).

The `ASK` query in Listing 2 returns true if the ending of a range label is found. The validation helps to discover any error in the mapping process.

```
PREFIX wqi: <http://wqi.ex/ont/ns#>

ASK {
  ?s a ?type;
    wqi:formElementLabel END_RANGE_LABEL.
       ↪
  FILTER(?type in (wqi:Input,wqi:Select))
       ↪
  FILTER(END_RANGE_LABEL in ("to","after
    ↪ "))
}
```

LISTING 2. Query to validate a range label.

A rule for the 'date' data type is also defined in this work. The basic structure of a date label is confirmed by three elements: day, month, and year.

Date rule: If a date label is found, then day, month and year labels must be part of the labels set. The validation of the `Date rule` follows the same steps as the `Range rule` considering the previous three elements.

The next section presents the experiments to validate and evaluate WebQuIn-LD.

IV. EXPERIMENTS AND EVALUATION

The experiments were conducted in a Linux machine with an Intel core i5 and 16 GB of RAM. Tel-8 and ICQ [22] datasets were used as experimental data as they are broadly used in the stated-of-the-art [2], [8], [9]. The ICQ dataset contains WQIs from four domains: airfare, automobile, book and job (see details in Table 2). The Tel-8 dataset is a collection of original and manually extracted query interfaces from eight representative domains, divided in three groups: Travel (Airfares, Hotels, and Car Rentals), Entertainment (Books, Movies, and Music Records), and Living (Jobs and Automobiles). See more details in Table 3.

TABLE 2. Statistic information of ICQ dataset.

Domain	#WQI	#Elements	#Input	#Select
Book	20	236	197	39
Auto	14	196	167	29
Job	14	77	44	33
Airfare	9	223	152	71
TOTAL	57	732	560	172

TABLE 3. Statistic information of Tel-8 dataset.

Domain	#WQI	#Elements	#Input	#Select
Automobiles	83	311	90	221
Movies	74	342	213	129
Music Records	65	239	184	55
Books	64	296	211	85
Jobs	49	338	104	234
Airfares	44	420	91	329
Car Rentals	24	179	63	116
Hotels	21	193	46	147
TOTAL	424	2318	1002	1316

The evaluation of WebQuIn-LD is divided in two main processes: label extraction and WQIs integration. The following subsection presents the results from these evaluations.

A. LABEL EXTRACTION EVALUATION

The label extraction process was evaluated considering the performance metrics of precision, recall and F-measure. Precision (Eq. 2) measures how many of the returned labels are correct. Recall (Eq. 3) measures how many of the labels that should have been returned are actually returned. Finally, the F1 metric (Eq. 4) is a balance between the quantity and the quality of labels. The label extraction process (Fig. 8) compares the label from the original WQI against the automatically extracted label from the OR-WQI using the metrics mentioned before. The comparison between the WQI's elements label and the OR-WQI resources label property (odis:formElementLabel) is made by means of the element's position, i.e., the WQI element's position is used to look for the corresponding resource in the OR-WQI through a SPARQL query. If the resource is found, the comparison between the resource label property and the WQI element's label is made.

$$Precision = \frac{TP}{(TP + FP)} \tag{2}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{3}$$

$$F - measure = \frac{2 * Precision * Recall}{(Precision + Recall)} \tag{4}$$

The evaluation results for the ICQ dataset (Table 4) demonstrate an overall F-measure above 96% for most of the cases. The Job domain contains the lowest number of WQI elements, which makes the label extraction process more difficult. On the contrary, the Book domain contains the highest number of WQI elements. The best F-measure was

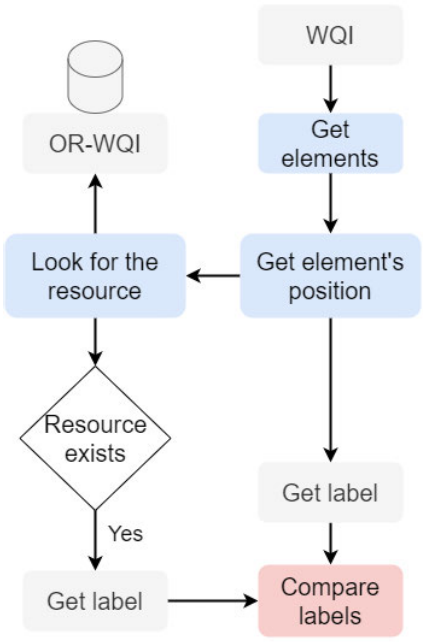

FIGURE 8. Label extraction evaluation process in WebQuIn-LD.

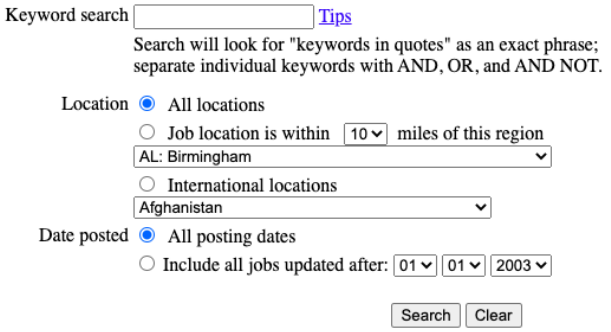

FIGURE 9. Example of a WQI from job domain.

obtained by the Auto domain and the worst F-measure was obtained by the Job domain. The main problem to select the correct label was the extraction of labels with explanatory text which affects precision and recall, e.g., Figure 9 shows an example of an extensive explanatory text before the *Location* label.

The evaluation results for TEL-8 dataset (Table 5) demonstrate good F-measure in the extraction label process with an overall result above 97%. The Music Records domain gets the lowest precision value and Airfares gets the highest precision value. The explanatory text problem shown in ICQ is also presented in this dataset, e.g., explanatory text is added to the correct label. However, the correct label is part of the extracted label. This problem does not impact the integration results since the correct label is part of the solution. The noun selection method, applied in the integration process, helps to alleviate this problem by selecting the correct label.

TABLE 4. Label extraction using the ICQ dataset.

Domain	Elements	TP	FP	FN	Precision	Recall	F-1
Book	236	220	12	4	94.83%	98.21%	96.49%
Airfare	223	210	10	3	95.45%	98.59%	97.00%
Auto	196	185	8	3	95.85%	98.40%	97.11%
Job	77	70	5	2	93.33%	97.22%	95.24%
TOTAL	732	685	35	12	94.87%	98.11%	96.46%

TABLE 5. Label evaluation using the TEL8 dataset.

Domain	Elements	TP	FP	FN	Precision	Recall	F-1
Airfares	420	412	5	3	98.80%	99.28%	99.04%
Movies	342	328	12	2	96.47%	99.39%	97.91%
Jobs	338	330	6	2	98.21%	99.40%	98.80%
Automobiles	311	305	4	2	98.71%	99.35%	99.03%
Books	296	281	12	3	95.90%	98.94%	97.40%
Music Records	239	210	18	11	92.11%	95.02%	93.54%
Hotels	193	187	4	2	97.91%	98.94%	98.42%
Car Rentals	179	170	5	4	97.14%	97.70%	97.42%
TOTAL	2318	2223	95	29	96.91%	98.50%	97.70%

TABLE 6. The most representative labels selected by WebQuIn-LD during the integration process for the ICQ dataset.

Label	Type	#Doc.	%Doc.	Label	Type	#Doc.	%Doc.
ICQ - Books (18)				ICQ - Autos (14)			
Author	Text	18	100%	Make	Select	10	71.43%
Title	Text	18	100%	Model	Text	9	64.29%
ISBN	Text	11	61.11%	Car type	Select	6	42.86%
Keyword	Text	10	55.56%	From	Text	6	42.86%
Publisher	Text	7	38.89%	To	Text	6	42.86%
Subject	Select	6	33.33%	Zip	Text	5	35.71%
Binding	Select	4	22.22%	Max	Text	4	28.57%
				Min	Text	4	28.57%
				State	Select	4	28.57%
				Year	Select	4	28.57%
ICQ - Airfare (9)				ICQ - Jobs (9)			
Adult	Select	6	66.67%	Keywords	Text	6	66.67%
From	Text	6	66.67%	Location	Select	5	55.56%
To	Text	6	66.67%	Job	Select	5	55.56%
Child	Select	4	44.44%	State	Select	4	44.44%

B. WQIs INTEGRATION EVALUATION

The integration process evaluation was carried out using the ICQ (Table 6) and TEL-8 datasets (Tables 7, 8, 9). The results show the label's frequency from the same domain, e.g., the Airfare domain is constituted by 9 documents and the label *Adult* appears in 6 of them (66.67%). For each label, some variants were considered referring to the same concept, e.g., the labels *Actor*, *Actress*, or *Star* refer to the same kind of job in WQIs from the Jobs domain.

The selected labels during the integration process were defined according to the average frequency calculated from all labels of the same domain. For example, in the Airfare domain the average frequency value is 5.5, then every label with a frequency greater than 5.5 was selected in the integration process. As a result, the most representative labels (highlighted rows) are part of the final result.

Table 6 shows the integration results of the ICQ dataset based on the automatic label selection. The elements selected for each domain correspond to the most frequent labels (highlighted rows in Table 6). For example, the most frequent label in the Autos domain is the *Make* label which appears in 10 of 14 documents. The frequencies shown in Table 6 demonstrate that WQIs from the same domain not always share the same fields (labels). The domain with the best integration results, based on the label frequency, is Books with two labels with a frequency of 100% (*Title* and *Author*).

The TEL-8 results are divided in three categories: Travel (Table 7), Entertainment (Table 8), and Living (Table 9). The integration results in travel category (Table 7) are based on the most frequent labels for Airfares, Car Rentals, and Hotels domains (highlighted rows in Table 7). The Airfares domain got the best integration results because many of its labels have

TABLE 7. The most representative labels selected by WebQuIn-LD during the integration process for TEL8 Travel category.

Label	Type	#Doc	%Doc	Label	Type	#Doc	%Doc	Label	Type	#Doc	%Doc
TEL8 - Airfares (44)				TEL8 - Car Rental (22)				TEL8 - Hotels (21)			
Departure Date	Select	44	100%	Pick up Location	Text	19	86.36%	Adults	Select	10	47.62%
Depart From	Select	44	100%	Pick up date	Select	15	68.18%	Rooms	Select	8	38.10%
Return Date	Select	44	100%	Pick up time	Select	12	54.55%	Children	Select	7	33.33%
Destination	Selection	44	100%	Drop off date	Select	12	54.55%	City	Text	5	23.81%
Adult	Select	31	70.45%	Car type	Select	12	54.55%	Hotel name	Text	5	23.81%
Children	Select	29	65.91%	Drop off time	Select	9	40.91%	Country	Select	4	19.05%
Infant	Select	14	31.82%	Drop off location	Text	8	36.36%				
Passenger	Select	8	18.18%	Car company	Select	7	31.82%				
Arrival	Select	7	15.91%	Flight number	Text	5	22.73%				
Airline	Select	6	13.64%	Airline	Select	4	18.8%				

TABLE 8. The most representative labels selected by WebQuIn-LD during the integration process for TEL-8 Entertainment category.

Label	Type	#Doc	%Doc	Label	Type	#Doc	%Doc	Label	Type	#Doc	%Doc
TEL8 - Music Records (65)				TEL8 - Movies (73)				TEL8 - Books (64)			
Artist	Text	65	100%	Title	Text	60	82.19%	Title	Text	47	73.44%
Title	Text	30	46.15%	Actor	Text	28	38.36%	Author	Text	41	64.06%
Label	Text	18	27.69%	Director	Text	23	31.51%	ISBN	Text	35	54.69%
Catalog	Text	8	12.31%	Genre	Select	16	21.92%	Keyword	Text	21	32.81%
Category	Select	8	12.31%	Format	Select	14	19.18%	Subject	Text	13	20.31%
Song	Text	8	12.31%	Keyword	Text	14	19.18%	Publisher	Text	8	12.50%
All	Text	7	10.77%	Category	Select	12	16.44%	Sort by	Select	8	12.50%
Format	Select	6	9.23%	Price	Select	11	15.07%	Format	Select	7	10.94%
Keyword	Text	6	9.23%	Rating	Select	11	15.07%	Category	Select	6	9.38%
				Release date	Text	10	13.70%				
				Studio	Select	6	8.22%				
				Cast/Crew	Text	5	6.85%				
				Language	Select	5	6.85%				
				Name	Text	5	6.85%				
				Theater	Select	5	6.85%				

a high frequency in most of the documents. The labels *Destination*, *Departure/Return date*, and *Depart* were part of all WQIs. The Hotels domain has the lowest integration results due to the low homogeneity of labels in this domain, being the labels *Adults*, *Children* and *Rooms* the most important with a frequency below %50.

The results of the integration process in the Entertainment category are obtained from the Books, Movies, and Music Records domains (Table 8). The domains from this category showed a high sparsity in their labels, being *Artist* the most frequent label in the Music Records domain.

The Living category results (Table 9) are obtained from the Automobile and Jobs domains. The Automobile domain contains the shortest set of labels, where the labels *Make* and *Model* are the most relevant. The Jobs domain contains a high diversity of labels, being the label *Keywords* the most representative.

The integration results show the most representative labels for each domain. The frequency obtained for each label demonstrated the diversity in some domain's labels, e.g., the *Artist* label, from the Music Records domain, and the *Destination* label, from the Airfares domain, were part of all input WQIs, meanwhile the *Children* label, from the Hotels domain, and the *Genre* label, from the Movies domain, appears in a few WQIs.

The results of WebQuIn-LD in different domains demonstrate its feasibility as an alternative solution to the WQI integration problem. The ontology proposed in this work, on which WebQuIn-LD is based, was determinant to improve the analysis and selection of the most valuable labels for the WQI integration task. We provide and make available to the community the semantic models (RDF files) from the URL https://www.kaggle.com/nhernandeztorres/webquinld.

C. DISCUSSION

According to the evaluation process, the label extraction considers the source and wrapper modules from the view-based architecture of WebQuIn-LD. The results for both datasets, ICQ and TEL-8, got a performance higher than 92%. In contrast to the state-of-the-art approaches, WebQuIn-LD represents a WQI as a semantic structure, mapping each WQI element into a linked data resource. The semantic representation facilitates the label extraction evaluation providing a mechanism to easily recover the target resource through SPARQL queries. Additionally, the proposed label extraction process takes advantage of SPARQL queries for the noun selection step and to apply the rules defined in [2] and [8].

The integration process considers the mediator module from the view-based architecture of WebQuIn-LD. According to the results, labels with explanatory text represent a

TABLE 9. The most representative labels selected by WebQuIn-LD during the integration process for TEL-8 Living category.

Label	Type	#Doc	%Doc	Label	Type	#Doc	%Doc
TEL8 - Automobile (4)				TEL8 - Jobs (47)			
Make	Select	34	82.93%	Keywords	Text	27	57.45%
Model	Select	32	78.05%	State	Select	20	42.55%
To	Select	24	58.54%	Job Category	Select	17	36.17%
Year	Select	17	41.46%	City	Text	15	31.91%
Zip	Text	11	26.83%	Location	Select	13	27.66%
				Country	Select	10	21.28%
				Industry	Select	9	19.15%
				Job type	Select	9	19.15%

challenge during the integration phase. The noun selection step helps to reduce this problem by selecting the most representative nouns from each label. In contrast to domain-dependent approaches, the experimental results demonstrate that WebQuIn-LD can be used as a domain-independent solution based on the GAV approach, generating a global solution considering the local information from the semantic representation of the WQIs (OR-WQI).

The integration results demonstrate the lack of consistency in the use of some WQI elements labels. For example, the Music Records category from the TEL-8 dataset contains several labels which are used by a few WQIs, e.g. the label *Song* appears only in 8 of 65 WQIs. In this case, the use of synonym dictionaries or domain ontologies could improve the results in a specific domain. These resources could help to improve the semantic of the labels in each domain by defining synonyms as rules between different concepts. WebQuIn-LD is able to determine the relevance of each WQI's element according to its label and independently to the type of element (radio, text box, etc.). Additionally, it is possible to query any kind of information from the WQIs, thus providing information for statistical purposes. This statistical information demonstrates the high diversity of domain's labels.

V. CONCLUSION

The domain-specific WQIs integration process tries to simplify the task of querying several WQIs with the aim of providing a unique input and an integrated output. The schema matching technique is the most common way to tackle this problem. This technique is based on the comparison between two or more WQI structures. The schema matching considers special features (e.g., the position of an element in the WQI) and text features (e.g., the name and label of an element) as the most important attributes during the integration of WQIs.

This work presented WebQuIn-LD, a linked data-based method for the integration of WQIs. It is based on a view-based data integration system architecture, divided in three main components: a source, a wrapper and a mediator module. The source module is constituted by the WQI dataset. The wrapper modules map each WQI into an Ontology Representation of the WQI (OR-WQI). The last module (mediator) integrates the content from all OR-WQIs to produce a unique WQI. The WQIs integration process considers a label selection step to identify the most representative noun from a label, especially for labels with explanatory content. This step is based on a WQI noun selection process and a noun frequency calculation. WebQuIn-LD includes a mechanism to automatically select the most relevant labels and elements from a set of domain-specific WQIs.

The label selection evaluation gets an overall result above 93%, which is competitive with the state-of-the-art works. In comparison with other solutions for the domain-specific WQIs integration problem, mainly based on structured formats like XML, WebQuIn-LD provides a semantic solution based on linked data. According with the obtained results, WebQuIn-LD selects the most representative WQI elements from a domain-specific dataset, providing a label analysis based on their frequency.

As future work, the WebQuIn-LD method will be studied and evaluated into a user-oriented system to query the deep web. Such system considers the stages of WQI collection, WQI integration, querying the deep web using the resulting IWQI, and the collecting, ranking and visualization of results. Furthermore, WebQuIn-LD could be evaluated in a user-oriented system for consulting integrated web forms. Label matching can be further improved by providing useful algorithms that reduce labels with the same meaning but written differently (e.g., children and infant). A possible solution could be to establish a metric for the most used labels by using a semi-supervised method, trying to create "same as" relationships between resources sharing the same meaning (for example *Children* sameAs *Infant*) and to determine the most used for each domain of study.

REFERENCES

[1] M. K. Bergman, "White paper: The deep web: Surfacing hidden value," *J. Electron.*, vol. 7, no. 1, pp. 1–17, Aug. 2001.

[2] C. Jou, "Schema extraction for deep web query interfaces using heuristics rules," *Inf. Syst. Frontiers*, vol. 21, pp. 163–174, Feb. 2018.

[3] K. C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang, "Structured databases on the web: Observations and implications," *ACM SIGMOD Rec.*, vol. 33, no. 3, pp. 61–70, Sep. 2004.

[4] L. Barbosa, H. Nguyen, T. Nguyen, R. Pinnamaneni, and J. Freire, "Creating and exploring web form repositories," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, New York, NY, USA, Jun. 2010, pp. 1175–1178.

[5] J. Madhavan, S. Cohen, X. L. Dong, Y. Alon Halevy, R. S. Jeffery, D. Ko, and C. Yu, "Web-scale data integration: You can afford to pay as you go," in *Proc. 3rd Biennial Conf. Innov. Data Syst. Res.*, Asilomar, CA, USA, Jan. 2007, pp. 342–350. [Online]. Available: https://www.cidrdb.org

[6] W. Wu, A. Doan, C. Yu, and W. Meng, *Modeling and Extracting Deep-Web Query Interfaces*. Berlin, Germany: Springer, 2009, pp. 65–90.

[7] T. Furche, G. Gottlob, G. Grasso, X. Guo, G. Orsi, and C. Schallhart, "The ontological key: Automatically understanding and integrating forms to access the deep web," *VLDB J.*, vol. 22, no. 5, pp. 615–640, Oct. 2013.

[8] H. M. M. Castro, V. S. Sosa, and M. A. N. Maganda, "Automatic construction of vertical search tools for the deep web," *IEEE Latin Amer. Trans.*, vol. 16, no. 2, pp. 574–584, Feb. 2018.

[9] C. Jou, "Deep web query interface integration based on incremental schema matching and merging," in *Proc. 3rd Multidisciplinary Int. Social Netw. Conf. Social Inform., Data Sci.*, New York, NY, USA, 2016, pp. 1–7.

[10] B. He and K. C.-C. Chang, "Statistical schema matching across web query interfaces," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, New York, NY, USA, 2003, pp. 217–228.

[11] Z. He, J. Hong, and D. Bell, "Schema matching across query interfaces on the deep web," in *Sharing Data, Information and Knowledge*, A. Gray, K. Jeffery, J. Shao, Eds. Berlin, Germany: Springer, 2008, pp. 51–62.

[12] C. Bizer, M.-E. Vidal, and H. Skaf-Molli, *Linked Open Data* New York, NY, USA: Springer, 2018, pp. 2096–2101.

[13] S. Sakr, M. Wylot, R. Mutharaju, D. Le Phuoc, and I. Fundulaki, *Linked Data—Storing, Querying, Reasoning*. Cham, Switzerland: Springer, 2018.

[14] W. Su, H. Wu, Y. Li, J. Zhao, F. H. Lochovsky, H. Cai, and T. Huang, "Understanding query interfaces by statistical parsing," *ACM Trans. Web*, vol. 7, no. 2, pp. 1–22, May 2013.

[15] Y. Wang, T. Peng, W. Zuo, and F. He, "Automatic integration of deep web query interfaces based on ontology," in *Proc. 4th Int. Conf. Comput. Sci. Converg. Inf. Technol.*, 2009, pp. 1654–1659.

[16] T. Landers and R. L. Rosenberg, *An Overview MULTIBASE* Norwood, MA, USA: Artech House, 1986, pp. 391–421.

[17] Y. A. Levy, *Logic-Based Techniques in Data Integration*. Boston, MA, USA: Springer, 2000, pp. 575–595.

[18] M. Lenzerini, "Data integration: A theoretical perspective," in *Proc. 21st ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst.*, New York, NY, USA, 2002, pp. 233–246.

[19] H. Nguyen, T. Nguyen, and J. Freire, "Learning to extract form labels," *Proc. VLDB Endowment*, vol. 1, no. 1, pp. 684–694, Aug. 2008.

[20] C. Charu Aggarwal and C. X. Zhai, *Mining Text Data*. New York, NY, USA: Springer-Verlag, 2012.

[21] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Intell. Res.*, vol. 37, pp. 141–188, Feb. 2010.

[22] The UIUC Integration Repository. (2003). *Computer Science Department, University of Illinois at Urbana-Champaign*. [Online]. Available: http://metaquerier.cs.uiuc.edu/repository

JULIO HERNANDEZ received the B.S. degree from the Meritorious University of Puebla, Mexico, in 2008, the MSc. degree in computer science from the National Institute of Astrophysics, Optics and Electronics, Mexico, in 2012, and the Ph.D. degree in computer sciences from the Center for Research and Advanced Studies, National Polytechnic Institute, Mexico, in 2019. He is currently a Postdoctoral Fellow with the Center for Research and Advanced Studies, National Polytechnic Institute. His current research interests include semantic web, natural language processing, and machine learning.

HEIDY M. MARIN-CASTRO received the B.Sc. degree from the University of Puebla, in 2004, the M.Sc. degree in computer science from the National Institute for Astrophysics, Optics and Electronics (INAOE), in 2008, and the Ph.D. degree in computer science from the Center for Research and Advanced Studies of the National Polytechnic Institute, Mexico (Cinvestav), in 2014. She is currently a Conacyt Researcher with the Information Technology Department, Autonomous University of Tamaulipas, Mexico. Her research interests include web data management, databases, data mining, and information retrieval.

MIGUEL MORALES-SANDOVAL received the B.Sc. degree from the University of Puebla, in 2002, and the M.Sc. and Ph.D. degrees from the National Institute for Astrophysics, Optics, and Electronics, Mexico, in 2004 and 2008 respectively. He is currently a computer science researcher with special interest in information systems, data security, cryptography, and embedded systems. He is currently focused on the development of software engineering, encrypted data retrieval mechanisms for security applications in the Internet of Things domain, and in the cloud. He actively participates in graduate programs in computer science and information technology and in the development of research projects. He has served as a reviewer for several journals and conferences and as an associated editor.