# Extracting Web Query Interfaces Based on Form Structures and Semantic Similarity

Jun Hong, Zhongtian He, David A. Bell

*School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT7 1NN, UK*
{j.hong,zhe01,da.bell}@qub.ac.uk

***Abstract*— Web databases are now pervasive. Such a database can be accessed via its query interface (usually HTML query form) only. Extracting Web query interfaces is a critical step in data integration across multiple Web databases, which creates a formal representation of a query form by extracting a set of query conditions in it. This paper presents a novel approach to extracting Web query interfaces. In this approach, a generic set of query condition rules are created to define query conditions that are semantically equivalent to SQL search conditions. Query condition rules represent the semantic roles that labels and form elements play in query conditions, and how they are hierarchically grouped into constructs of query conditions. To group labels and form elements in a query form, we explore both their structural proximity in the hierarchy of structures in the query form, which is captured by a tree of nested tags in the HTML codes of the form, and their semantic similarity, which is captured by various short texts used in labels, form elements and their properties. We have implemented the proposed approach and our experimental results show that the approach is highly effective.**

## I. Introduction

Web databases are now pervasive on the Web. These databases are 'hidden' on the Web as they are accessed via their query interfaces (usually HTML query forms) only. Queries to Web databases are made by filling out and submitting query forms. On receiving form-based queries, these databases return query results encoded in HTML, which are then displayed by a Web browser.

Many Web sites are supported by Web databases. For example, amazon.com provides a query form for searching its book database. In a specific application domain (e.g. flight booking, book sales), there are many database-driven Web sites that sell similar products or services. It is a daunting task for users to visit numerous Web sites individually to search for and compare services or products. Much research on Web data integration has been done to develop systems that provide integrated access to a multitude of Web databases, where users fill in a uniform query form only, and on receiving a user query the system will automatically make connections to different sites, fill in the local query forms on these sites, submit these forms, combine the query results, and return the combined results to the users.

There are a number of challenges in automating the process of integrating data from multiple Web databases [1]. The first challenge is to semantically understand query forms. Query forms are written in HTML, which are displayed for human use. To make them machine understandable, their formal representations need to be created by extracting query conditions in them. Semantic understanding of query forms enables queries to be made by filling in a query form automatically. It also provides a basis for the second challenge on reconciling semantic heterogeneity between query interfaces. The third challenge is to semantically understand query results that are returned from different Web sites in response to a user query. These challenges lead to four research problems: query form extraction [2], [3], [4], matching and filling [3], [5], [6], [7], [8], [9], and query result extraction [10]. In this paper, we focus on the first problem, i.e. query form extraction.

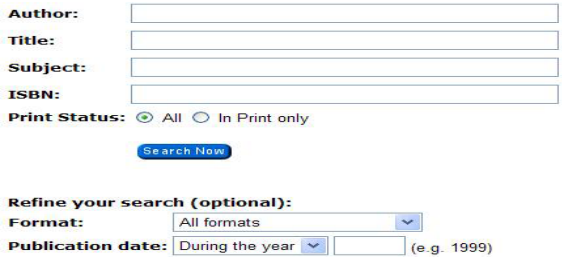


Fig. 1. A Web Query Interface: AllBookStores.com

Semantically a query form contains a set of query conditions, each consisting of an attribute and one or more operation (predicate) on the value of the attribute, which expresses one or more constraint on the attribute. For example, the query form shown in Fig. 1 may consist of 7 query conditions which can be created by filling in the corresponding query condition templates. The query condition on "Publication date" can be created by selecting one of the operators on the selection list and giving a specific year in the text box.

In a query form, a query condition template is encoded in HTML by a group of labels and form elements that play different semantic roles in a query condition. For example, the query condition template on "Publication date" as shown in Fig. 1 is represented by a label, a selection list and a text box: the label "Publication date" represents an attribute, the selection list represents a list of optional operators, and the text box receives a free value from the user for the selected operator. Query form extraction, therefore, involves recognizing the semantic roles that labels and form elements play in query conditions and hierarchically grouping labels and form elements that are related to individual query conditions

1084-4627/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDE.2009.215

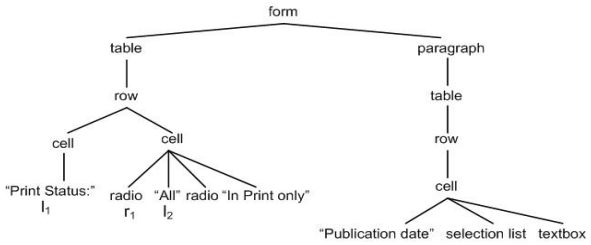


Fig. 2. Part of the Hierarchy of Structures in a Query Form - AllBookStores.com

into constructs of query conditions.

Several approaches [3], [4], [6] have been developed for query form extraction. While significant progress has been made in these approaches, none of them has explored use of semantic similarity between labels and form elements to group labels and form elements though it is commonly used by humans. Another problem with these approaches is that they use heuristics and grammatical rules that are domain-specific and have to be manually acquired and represented for each application domain.

We have observed that semantic similarity between labels and form elements provides an important measure on their associations. In addition, we have observed that query forms are often hierarchically structured by the form designer, as shown in Fig. 2. The form designer uses HTML tags to represent the hierarchy of structures in a query form. Because this hierarchy of structures reflects the form designer's intention on grouping labels and form elements into constructs of query conditions, relationships between labels and form elements in this hierarchy provide strong clues on their associations.

This paper presents a novel approach to extracting query forms, which shows a strong correlation to human intuition and the form designer's intention. First, to group labels and form elements in a query form, our approach makes use of two types of clues: semantic similarity between labels and form elements, and their structural proximity in the hierarchy of structures in the query form. Second, we propose a method for computing semantic similarity between short texts. Third, we parse the HTML codes of the query form into a DOM tree to determine the structural proximity of labels and form elements in the hierarchy of structures in the query form. Fourth, we define a generic set of query condition rules to tag the semantic roles of labels and form elements in a query condition and hierarchically group them into constructs of query conditions. Fifth, we report the experimental evaluation of our approach on a large set of Web query interfaces that shows that our new approach has led to much improved quality of query form extraction and is highly effective.

## II. A Novel Approach to Query Form Extraction

### A. Labels and Form Elements

In HTML a query form is composed of both labels and form elements. Four types of form elements are used to receive input from the user, including radio buttons, check boxes, text boxes, and selection lists. They allow the user either to choose from a set of preset options or to enter a free value. A form element represents a list of optional attributes, operators or preset values, or a free value to be entered. Labels in a query form are descriptive texts, each representing an attribute, an operator or a component of a complex attribute.

### B. Defining Query Conditions

A query form contains a set of query conditions that are semantically equivalent to search conditions in a SQL query. A query condition consists of a set of labels and form elements that are hierarchically grouped into constructs of query conditions. How a query condition construct is composed and the semantic roles of labels and form elements in a query condition can be defined for each type of query conditions. We create a generic set of query condition rules to define each type of query conditions. Query condition rules are domain independent and not specific to individual query forms because of their two distinctive characteristics respectively. First, they are semantically equivalent to search conditions in SQL. Second, they define only what semantic roles labels and form elements play in a query condition and how they are hierarchically grouped into a query condition.

Query condition rules are defined in the form of the EBNF rules. Each query condition rule has the structure: `head ::= body`. There are two types of query condition rules. In the first type of query condition rules, the head represents a query condition construct while the body represents either a label or a form element. The meaning of the rule is that the label or form element represented by the body can play the semantic role represented by the head in a query condition. For example, we have two rules

```
<Free Value> ::= "Text Box"
<Preset Value List> ::= "Selection List"
```

where the first means that a free value in a query condition can be provided in a text box by the user while the second means that a list of preset values can be provided by a selection list for the user to choose.

In the second type of query condition rules, the head represents a query condition construct while the body represents a pair of query condition constructs. The meaning of the rule is that the pair of query condition constructs represented by the body of the rule can be grouped into a query condition construct represented by the head of the rule. For example, the rule

```
<Comparison Query Condition> ::=
            <Attribute><Comparison Operation>
```

means that a comparison query condition is composed of an attribute and a comparison operation. A query condition defines only what types of labels and form elements can be grouped into a query condition construct. There is no requirement on the sequential or spatial relations between the query condition constructs in the body of the rule. In the above rule, it is not required that a comparison query condition must start with an attribute followed by a comparison operation. We instead require that those query condition constructs in

the body of the rule to satisfy two general constraints on hierarchical proximity and semantic similarity.

### C. Structural Proximity and Semantic Similarity

**Structural proximity**: A query form is often designed as a hierarchy of structures represented by the appropriate HTML tags. Labels and form elements that form a query condition construct are often positioned in close proximity in this hierarchy, that is, hierarchically they are close to their lowest common ancestor structure in the hierarchy (hierarchical proximity), and if they are in the same parent structure, they are close to each other within the parent structure (sibling proximity).

For example, as shown in Fig. 2, radio button $r_1$ is in closer hierarchical proximity with label $l_2$ ("All") than label $l_1$ ("Print Status"), in the hierarchy of structures in the query form even though when the query form is displayed, as shown in Fig. 1, two labels $l_1$ and $l_2$ appear to be in the same distance from radio button $r_1$. Therefore label $l_2$ ("All") rather than label $l_1$ ("Printed Status") should be associated with radio button $r_1$.

**Semantic similarity**: Labels and form elements that form a query condition construct are often semantically similar. A label is a short text that has semantic meaning. A form element has a set of short texts associated with it, which can be extracted from its properties (i.e. name and value) and the labels used in it (i.e. the labels of the options in a selection list, the labels associated with radio buttons/check boxes). When we say that labels and form elements are semantically similar we mean that the sets of short texts associated with them are semantically similar.

### D. Query Form Extraction

We propose a novel approach to query form extraction, which takes into account both structural proximity and semantic similarity, mimicking the hierarchy of structures in a query form and human intuition. Given the HTML codes of a query form, query form extraction is carried out in four stages. First, the HTML codes are parsed into a DOM tree which represents the hierarchy of structures in the query form. Second, the labels and form elements in the query form are extracted. Third, the extracted labels and form elements are parsed into such query condition constructs as operators, selectors, attributes, values, attribute lists and operator lists using the appropriate query condition rules, which represent the semantic roles they can play in query conditions. Fourth, different types of query condition rules are used to parse appropriate query condition constructs into a higher level of query condition constructs if the labels and form elements in them satisfy both structural proximity and semantic similarity constraints. This parsing process continues level-by-level until labels and form elements have been eventually parsed into query conditions.

## III. Experimental Results

We have implemented the algorithms developed in our approach in a prototype query interface extractor. To evaluate the performance of these algorithms, we have used a data set that consists of 104 query interfaces from 4 application domains, including Books (38), Music Records(17), Movies (28) and Automobiles (21). These query interfaces are part of the TEL-8 Query Interfaces data set in the UIUC Web Integration Repository[1]. The experiments are carried out on individual query interfaces, in each of the domains and across the domains. Our experimental results show that our proposed approach to query form extraction is highly effective.

We use three performance metrics: precision, recall, and F-measure. A query condition extracted by our extractor is taken as correct if it is one of the query conditions manually extracted. Precision is the ratio between the number of correct query conditions extracted by our extractor and the total number of query conditions extracted by our extractor. Recall is the ratio between the number of query conditions correctly extracted by our extractor and the total number of query conditions manually extracted. F-measure is the incorporation of precision and recall, which is defined as $(2pr)/(p + r)$, where $p$ and $r$ represent precision and recall respectively.

Fig. 3(a), 3(b) and 3(c) show interface distributions over precision, recall and F-measure respectively in each of the domains and across the domains. For instance, in the domains of books, automobiles, movies, music records and across the domains, 78.9%, 71.4%, 64.3%, 70.6% and 72.1% of interfaces have 1.0 precision. Fig. 3(d) and 3(e) illustrate the experimental results on the average rates of precision, recall and F-measure per query form and the overall rates of precision, recall and F-measure, in each of the domains and across the domains.

## IV. Related Work

Several approaches [3], [4], [6] have been developed for query form extraction, which have the commonality that information about the layout of a form as it is displayed is used to associate labels and form elements with each other. Some of these approaches use various heuristics on the association of labels and form elements in terms of their positions in form layout while the others use a set of grammar rules to represent common patterns for various constructs of query conditions.

The main differences between our approach and these approaches are threefold. First, instead of using information about form layout as it is displayed, we use information about the hierarchy of structures in a query form created internally in the HTML codes by the form designer. Second, we use information about the semantics of labels and form elements. Third, we use query condition rules to represent the semantic compositions of query conditions in a query form, which semantically conform to the corresponding set of SQL search conditions.

The approach reported in [4] is the most related to ours, in which query interfaces are viewed as a visual language and the visual compositions of query conditions in a query interface as conforming to a hidden grammar that has to be

[1] http://metaquerier.cs.uiuc.edu/repository/

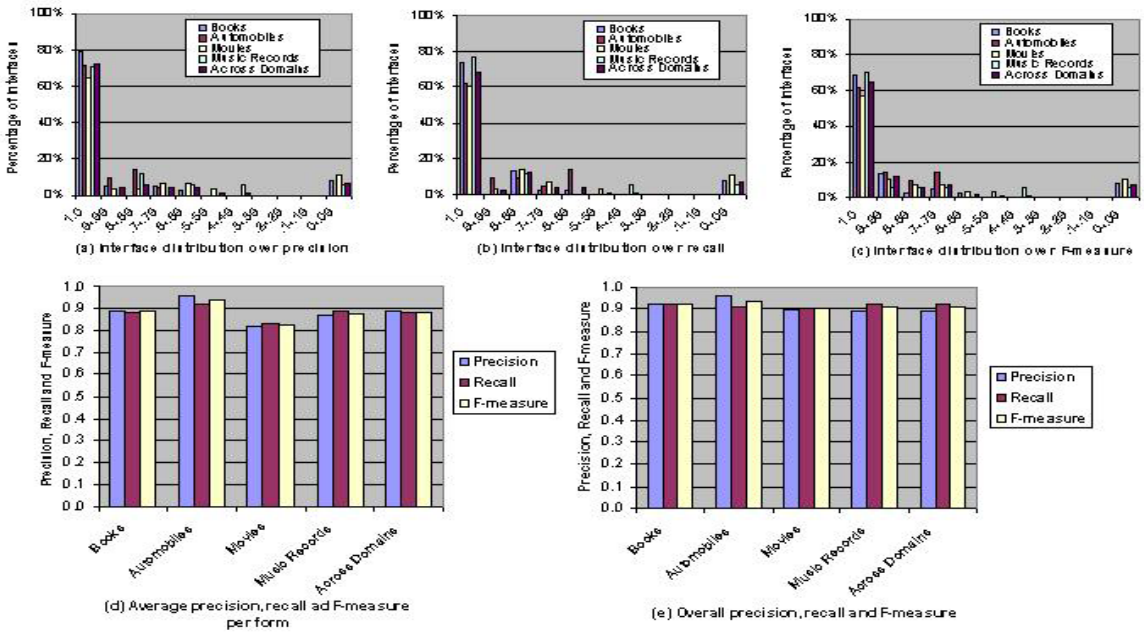


Fig. 3. Experimental Results on Precision, Recall and F-measure

derived. The extraction of query conditions, as the inverse, is a parsing process, which uses a set of derived grammar rules to hierarchically group labels and form elements and tags their semantic roles. In their approach, the visual compositions of query conditions as they are displayed are grammatically defined.

In our approach, query interfaces are viewed as a semantic model and the semantic compositions of query conditions as semantically conforming to a set of query conditions that are equivalent to the corresponding set of SQL search conditions. While a set of grammar rules in [4] have to be derived and manually generated for different domains, our query condition rules define a generic set of query conditions that are domain independent, not specific to individual query forms, and semantically correspond to a set of SQL search conditions.

## V. Conclusions

In this paper, we considered the problem of query interface extraction. We proposed a novel approach to extracting query forms, which shows a strong correlation to human intuition and the form designer's intention. Our approach is based on the following observations: (a) query conditions in a query form are equivalent to search conditions in SQL; (b) labels and form elements in a query condition are in structural proximity in the hierarchy of structures in the query form; (c) There exists semantic similarity between labels and form elements in a query condition; (d) The semantic composition of a query condition can be represented by a set of query condition rules and such semantic composition plus two general constraints on labels and form elements in the query condition can be used together to extract the query condition. The experimental evaluation of our approach on a large set of query interfaces shows that our new approach has led to much improved quality of query form extraction and our new approach is highly effective. In the near future work, we will be extending our approach to extract query conditions that contain complex attributes.

## References

[1] K. C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang, "Structured databases on the web: Observations and implications," *SIGMOD Record*, vol. 33, no. 3, September 2004.

[2] H. He, W. Meng, C. Yu, and Z. Wu, "Constructing interface schemas for search interfaces of web databases," in *Proceedings of the 6th International Conference on Web Information Systems Engineering (WISE 2005)*, New York City, New York, 2005.

[3] S. Raghavan and H. Garcia-Molina, "Crawling the hiddenweb," in *Proceedings of the 27th VLDB Conference*, Roma, Italy, 2001.

[4] Z. Zhang, B. He, and K. C.-C. Chang, "Understanding web query interfaces: Best-effort parsing with hidden syntax," in *SIGMOD Conference*, Paris, France, 2004.

[5] B. He and K. C.-C. Chang, "Statistical schema matching across web query interfaces," in *SIGMOD Conference*, San Diego, CA, 2003.

[6] H. He, W. Meng, C. Yu, and Z. Wu, "Wise-integrator: An automatic integrator of web search interfaces for e-commerce," in *VLDB Conference*, Berlin, Germany, 2003.

[7] W. Wu, A. Doan, and C. Yu, "Webiq: Learning from the web to match deep-web query interfaces," in *ICDE Conference*, Atlanta, Georgia, 2006.

[8] W. Wu, C. Yu, A. Doan, and W. Meng, "An interactive clustering based approach to integrating source query interfaces on the deep web," in *SIGMOD Conference*, Paris, France, 2004.

[9] Z. Zhang, B. He, and K. C.-C. Chang, "Light-weight domain-based form assistant: Queryingweb databases on the fly," in *Proceedings of the 31st VLDB Conference*, Trondheim, Norway, 2005.

[10] J. Wang, J.-R. Wen, F. Lochovsky, and W.-Y. Ma, "Instance-based schema matching for web databases by domain-specific query probing," in *Proceedings of the 30th VLDB Conference*, Toronto, Canada, 2004.