

Semantic Completely Preprocessing for Deep Web Queries Translation

Hao liang^{1,2}, and Fei Ren³

1. College of Computer Science and Technology, Jilin University, Changchun 130012, China;

2. Department of Information, Changchun Taxation College, Changchun 130117, China

3. China Development Bank, Center of Operations, Beijing, 100037, China

Email: liangh434@163.com

Abstract—Local schemas embedded in query interfaces of Deep Web are partial of database schemas. However, local schemas are extracted incompletely from query interfaces coded by semi-structure programming language. Some key attributes missed lead to form incompletely semantic queries. To tackle the problem, a semantic completely preprocessing for queries is presented based on Ontology instrument and instance information in the query interface. Instance information are extracted and attached to right attributes and can be used to instantiate some semantic meaningless attributes playing an important roles in schemas. By generating semantic completely queries, the information in Deep Web can be accessed. The precision and recall of query translation is improved by instance information in a certain extent.

Index Terms—Local schema, Deep Web, Ontology, instance information, meaningless

I. INTRODUCTION

The World Wide Web should be divided into the Surface Web and the Deep Web^[1]. The Surface Web consists of billions of searchable pages, while the Deep Web is hidden behind the Surface Web remaining unsearchable. A survey in April 2004 estimated there were more than 450,000 online databases^[2]. Myriad information may not be accessed through static URLs because they are presented as results after users submitted the query. The Deep Web databases require manual query interfaces and dynamic programs to access their contents, thus preventing Web crawlers from automatically extracting their contents and indexing them, and therefore not being included in search engine results.

Search engines and web crawlers can not access the Deep Web directly. The workable way to access the hidden database is through query interfaces. To access the Deep Web is an inverse transaction of Web service construction. Automatic extracting attributes from query interfaces and translating queries is a solvable way for addressing the current limitations in accessing Deep Web. However, even in the same domain, different query interface designers have different develop and design model. To generate a share and global schema based on local schemas is a challenge. Some methods can be concerned such as type-based search-driven translation framework by leveraging the “regularities” across the implicit data types of query constraints. He B, et al. found that query constraints of different concepts often share similar patterns, and encoded more generic translation

knowledge for each data type^[3, 4, 5]. They provided an extensible search-driven mechanism.

In the previous work, attributes of the query interfaces were obtained manually and the co-occurrence of attributes was used to evaluate the domain information^[6]. In our framework, we measure the relevance of attributes not only with the exact matching, but also with the semantic similarity. The automatic attribute extraction is the indispensable previous pretreatment of schema matching, schema merging, and the carrying out of some correlated research fields depended on the result of it, such as discovering, categorizing, indexing, and query capability modeling Deep Web sources and extracting domain knowledge^[7-9]. However, the automatic attribute extraction has always been proven to be a difficult task^[10]. Yoo, et al. have put forward an automatic attribute extraction algorithm to automatically determine the attributes of Deep Web query interfaces by utilizing WordNet^[11]. Two types of attributes, programmer viewpoint attributes and user viewpoint attributes, were defined. The final attributes of a query interface were determined by checking the overlapping areas between programmer and user viewpoint attribute sets. The most obvious difference between the methods of Yoo's and ours is that we extracted the instances information during the attributes extraction procedure. We believed instances are a major part of describing the semantic attributes and sometimes provide instancial information in query interfaces to depict the meaningless words, such as “From”, “To”, “ISSN”, et al. The semantic of query interfaces is not complete with some meaningless words semantically un-instantiated. The recall of query translation is lowered by the semantic incomplete query instances.

Schemas of Deep Web are composed of attributes in query interfaces, so the validation and effectiveness of attribute extraction is the most important factor during accessing to Deep Web. We try to extract abundant attributes, which describe the concepts, and the semantic relationships between attributes. The most efficient and effective Ontology technique of detecting the semantic relation between words is the WordNet. During attributes extraction, the instance information is a more important part, by which some meaningless attributes are instantiated by semantic related instance information in the query interface to generate a semantic completely query.

II. HELPFUL HINTS

A. Different level attributes

To describe an object in the sense of a designer, each characteristic of object is equal to an attribute. A schema is composed of different attributes depicted the same object. In the information system, information should be stored as a record according to schemas. During this step, the attribute is extended by adding description and restriction information. To share the information stored in databases, Web pages with Web services are designed by translating schemas into query interfaces. In Fig 1, we can see the attribute extended two times from logic description to query interface. The most obviously extended stage is from database schema to query interface.

Deep Web is composed of kinds of diversity web databases scattered on the whole Web. In the databases provided some domains Web services, the original

attributes named by database designers, describe some objects and constitute the schemas of the databases. Web designers extend the database schema attributes during the procedure of designing query interfaces, because the different programming environment provides different controls for Web designers. Take the “time” attribute as an example, some designers maybe need three control fields to describe “year”, “month” and “day”, so attributes of query interface are the results of second extending original attributes. General Web users can use interfaces provided by the Web designers to carry out some query function. The interface is composed of attributes which are Web designer attributes and some text description information. It is an extending procedure of the attributes from logic description to query interface. After general Web users creating new query instances and carrying out the search function operation, the matching procedure between the instances and attributes is like an inductive procedure, which carried out from specialization to generalization.

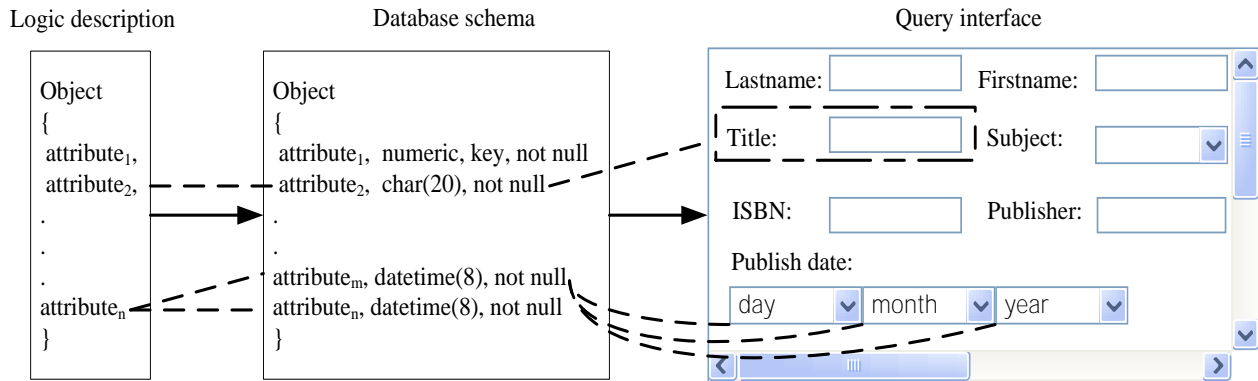


Figure 1. The attribute extended from logic description to query interface.

Query interfaces of Deep Web are coded by HTML according to W3C specification. The query interface contains some form elements, such as label, textbox, radio button, checkbox and selection list, which allow users to enter or choose search information and form the query instances. A descriptive text label is usually associated with an element to describe the semantic meaning of the element and sometimes the instance text has the same function. Logically, elements and their associated labels together form different attributes of the underlying database. Using such a schema-based interface, users can specify complex and precise queries and pass them to Deep Web.

In the query form representation, there are some associated elements describing an attribute, which sometimes showing the semantic information of the attribute. Each label is considered to be the name of an attribute of the underlying schema. Each element has a format which is the input format of the element. There are generally four types of formats: text box, radio button, check box and selection list. Some element also has a domain that defines the set of values that can be used to instantiate the element when forming a query instance. Textbox allows users to input whatever text they want and the box and radio button have one associated value.

Multiple checkboxes or multiple radio buttons are used together to accomplish the same function as a selection list. In addition, each element or a group of elements may have its or their default value, which is used to help forming a query instance when users do not make a different selection. For each attribute, there is a type for its values, such as numeric, date, time, and currency. In some query forms, numeric type instance indicates that the attribute is used for identification purpose. The type information can be obtained through analyzing attribute name and the pattern or format of attribute values. For example, \$15 is an instance of currency type and 18:26 for time. When the value type is difficult to determine, a default value type, i.e., char, is used. In addition, the scale/unit of the attribute values is also concerned. Because during the query translation, the attributes of different query forms maybe share the same value type, but different scales and unit. To ensure the translation accuracy, some additional information is indispensable. Finally, each attribute has its layout position in the interface. The position value is determined by the layout order of attributes in an interface. More important attributes are usually arranged ahead of less important ones. In addition to the label of an attribute, each element

of the attribute may have its own label. Such label helps define the semantic meaning of the element.

B. Attribute extraction

We take EN short for element name and ET for element text. Both EN and ET are important information to get the semantic understanding of a query interface. In our demo, we extract EN set and ET set of a query interface to determine valid attributes, candidate attributes are stored in the refined EN set and ET set. The algorithm is as following:

- step 1. Get IIS (inner identifier set) and TIKW from web data source,
- step 2. Remove special symbols, generate more substrings. Special symbols (:, -, _, @, \$, &, #, ?, !, *, etc.)
- step 3. Remove duplicated in the IIS and TIKW.
- step 4. Extract all text between <option> and </option> form the instance set, INS for short.
- step 5. Pre-process function, PPF for short.
- step 6. Extend the key words of IIS, TIKW and INS into a set by utilizing WordNet.
- step 7. Generate hierarchy tree for EN and ET.
- step 8. Add instances into ET tree mark the relation between candidate attributes and instances; mark the relation between candidate attributes and instances.
- step 9. Refine the hierarchy tree and get EN and ET set in a hierarchy relation tree.

C. Instance extraction

At this step, we consider the texts of instance information in query interface and compute the visual distance with respect to each field of the form F. The

visual distance between a text instance ti and a candidate attribute ca is computed as follows:

1. We can use the APIs provided by the browser to obtain the coordinates of a rectangle enclosing ca and a rectangle enclosing ti . If ti is in a HTML table cell, and it is the unique text inside, then we can mark the correlative relation between them.
2. We also obtain the angle of the shortest line joining both rectangles. The angle is approximated to the nearest multiple of $\pi/4$.

For each candidate attribute, we try to obtain the texts instance semantically linked with it in the query interface. For selecting the best matching text information for a ca , we apply the following steps:

1. We add all the text instances with shortest distance with respect to ca into a list.
2. Those text instances having a distance lesser with respect to ca are added to the list ordered by distance. This step discards those text instances that are significantly further from the field.
3. Text instances with the same distance are ordered according to its angle. The preference order for angles privileges texts aligned with the fields. The main standards to measure the preference is that privileges left with respect to right and top with respect to bottom, because they are the preferred positions for labels in forms.
4. After extracting the ET, we also can get a tree hierarchy exhibition of the candidate attributes. In Fig 2, it shows an ET tree extracted from the HTML of query interface and the refined ET tree with added instances.

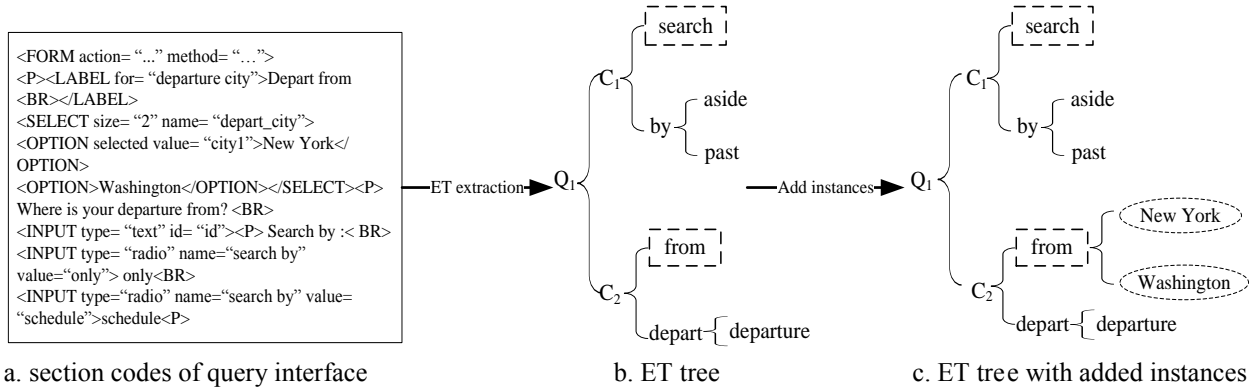


Figure 2. The attribute extended from logic description to query interface.

In Fig 2b, the “from” and “search” are in dashed blocks, because they are a little different from the other terms. We can get no information form WordNet about “from” and we also find that “search” is the more general word in the group of the same meaning words set. In this situation, “from” and “search” are called atom lemmas. So there are no extra information form the WordNet and they have no leave nodes comparing with “depart” and “by”. In Fig 2c, there are two text instances and showed in circle blocks. We believe that the instances information are belonged to both “depart” and “from”. It is easy to find that the instance “New York is an INSTANCE OF city, metropolis” and urban center” in the WordNet. In other words, “city” is the source lemma of “New York”.

We describe the relation between candidate attributes and instance like:

$S_{depart} = \{depart \parallel departure, INSTANCE ((New York, "instance of" (city, metropolis)), (Washington, "instance of" (city, metropolis)))\}$,

$S_{from} = \{from \parallel INSTANCE ((New York, "instance of" (city, metropolis)), (Washington, "instance of" (city, metropolis)))\}$.

III. EXPERIMENTS AND RESULTS

Our experiment is to translate queries from one query interface to another in the same domain. In this paper, we carry out experiment on Airfares, Automobiles, Books,

Music and Movies domains. The experiment results and the comparison with Yoo's method are shown in Table 1. The parameters are defined as following:

- QN: the number of query interfaces in specific domain;
- AEP: the average precision of attribute extraction in specific domain;
- TR: the recall of query translation in specific domain;
- TP: the precision of query translation in specific domain;
- Yoo-TP: the precision of query translation of Yoo's algorithm.

The translating precision of our method is 2% higher than Yoo's on average. Especially in the Airfares domain, we take full advantage of the text instance information to improve the attribute extraction precision and translating precision. However, there are some lacks of our method in the Books and Music domains. The semantic container is difficult to capture in the two domains and the default processing of the semantic container is not effective enough.

TABLE I. THE RESULTS OF AIRFARES, AUTOMOBILES, BOOKS, MUSIC AND MOVIES DOMAINS

Domain	QN	AEP	TR	TP	Yoo-TP
Airfares	35	85.3%	81.7%	76.7%	67.8%
Automobiles	36	87.3%	82%	74.5%	70%
Books	25	93%	85%	79%	100%
Music	25	87%	83%	72.2%	77%
Movies	25	95%	87%	82.7%	60%
Average:	29	89.5%	83.7%	77%	75%

IV. FUTURE WORK

Our framework illustrates an automatic process of extracting attributes from query interfaces and query translating. We propose to get more valid attributes by using Ontology technology, each candidate attribute is extended into form a synonym set by WordNet and stored in a tree data structure. During the attributes extracting, the instance information is extracted together in the aim of being attached to semantic attributes and instantiating meaningless attributes. As we all know, attributes in the query interface are still part of the schema hidden in the Web database, so in our opinion the semantic completeness of the query interface is very important factor of determining precision and recall of query translation. After attributes extraction, query translation is carried out based on the semantic containers generated from query interfaces. The precision and recall of query translation are improved by attributes extracted and

instantiated by our algorithm, especially in the Airfares domain. However, the precision and recall are not effective enough, because the semantic restrictions in query interfaces are too hard to deal with only by parsing technique. The design style and purpose of query interfaces can not be modeled by anyone else except the designer.

REFERENCES

- [1] Bin He, Kevin Chen-Chuan Chang. Statistical schema matching across Web query interfaces. In Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, USA, June, 2003, pp: 217-228.
- [2] K. C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang, Structured databases on the web: Observations and Implications, ACM SIGMOD Record, September 2004. 33(3):61-70.
- [3] B. He, Zhen Zhang, and Kevin C.-C. Chang, MetaQuerier: Querying Structured Web Sources On-the-fly, In Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June, 2005, pp: 927-929.
- [4] B. He, K. C.-C. Chang, Automatic complex schema matching across web query interfaces: A correlation mining approach, ACM Transactions on Database Systems, Association for Computing Machinery, Vol.31(1), 2006, pp: 346-395.
- [5] B. He, Patel M, Zhang Z, K. C.-C. Chang, Accessing the deep web, Communication OF The ACM, Vol.50 (5), MAY 2007: 95-101.
- [6] G. Kabra, C. Li and K.C. Chang. Query Routing: Finding Ways in the Maze of the Deep Web. In Proceedings of the International Workshop on challenges in Web Information Retrieval and Integration, Tokyo, Japan, April, 2005, pp: 64-73.
- [7] Caverlee J, Liu L, Rocco D, Discovering Interesting Relationships among Deep Web Databases: A Source-Biased Approach, World Wide Web-Internet and Web Information Systems, Vol.9(4): 585-622, Springer, 2006.
- [8] Shu LC, Meng WY, He H, Yu C, Querying Capability Modeling and Construction of Deep Web Sources, In Proceedings of 8th International Conference on Web Information Systems Engineering, Nancy, France, Dec, 2007, pp: 13-25.
- [9] LIU Wei, MENG Xiao-Feng, MENG Wei-Yi, A Survey of Deep Web Data Integration. Chinese Journal of Computers, Vol.30, No. 9, Sept, 2007, pp: 1475-1489.
- [10] Sriram Raghavan, Hector Garcia-Molina. Crawling the hidden Web. In Proceedings of 27th International Conference on Very Large Data Bases, Roma, Italy, Morgan Kaufmann, September, 2001, pp: 129-138.
- [11] Yoo Jung An, James Geller, Yi-Ta Wu, Soon Ae Chun: Semantic deep web: automatic attribute extraction from the deep web data sources. In Proceedings of the 2007 ACM Symposium on Applied Computing (SAC2007), Seoul, Korea, March, 2007, pp: 1667-1672.