DEEP WEB DATA INTEGRATION APPROACH BASED ON SCHEMA AND ATTRIBUTES EXTRACTION OF QUERY INTERFACES

Mr. Gopalkrushna Patel*

Anand Singh Rajawat**

Mr. Satyendra Vyas***

Abstract:

The deep web is becoming a collection of informative resource unlike the conventional web information that allow retrieval of the contents into the Deep Web only accessible through source query interfaces. However, there may be many query interfaces for any domain of interest that users need to access them in order to get the desired information. Accesing the information is a time consuming processes and requires building an integrated query interface over the sources. The first important task in our research is schema extraction from source query interface and second is automatic extracting attributes from the query interface and automatically translating a query, is a solvable way for addressing the current limitations in accessing Deep Web data sources. In this paper, we present a effective Deep web data integration approach based on Schema and Attributes Extraction of Query Interfaces. Our approach can also avoid obtaining the incorrect subsets while grouping attributes and is highly effective on schema extraction of source query interfaces on the invisible web.

Keywords: Surface Web, Invisible web, Schema and Attribute.

* Research Scholar, JJTU University, Rajasthan, M.TECH (IT)-IET, Alwar.

** Institute of Engineering (CSE), JJT University, jhunjhunu, CSE Department, SVITS.

*** Lecturer, CS&IT Department, IET-Alwar, Rajasthan

I. INTRODUCTION:

Mainly of the information on World Wide Web can not be directly accessed by the static link. Users must type some keywords before getting the information hidden in the Web records. Invisible web contains 400-500 times more information and 15% larger visit capacity than that of Surface Web. Additionally, the quality of data is also comparatively higher [11]. The study on Data Extraction from Invisible web pages is becoming a popular area. The study is to assist people access automatically and use freely the information distributed on the Invisible web. [2] and [3] focus on the technology of the search of Deep Web database on world wide web [4], [5] and [6] study the methods of extracting and the attribute from the query interface and the ones of construction a standardized pattern, which contribute to integrate several Invisible web. Data extraction is another important aspect of Deep Web research, which involves in extracting the information that users are interested in from semi-structured or unstructured Web pages and saving the information as the XML document or relationship model. [7], [8] and [9] have done a lot of work in this field. Additionally, in some papers, such as [10] and [5], researchers have paid extra concentration to the influence of semantic information on Invisible web about the schema extraction problem on the Invisible web. Let’s get to know a few resources which will be our deep diving vessel for the Invisible Web. Some of these are invisible web search engines with specifically indexed information, most popular deep web or invisivile web search engine, the WWW Virtual Library, Intute, CompletePlanet, Infoplease, DeepPeep, IncyWincy, Deep Web Tech, TechXtra, Scirus With its myriad databases and hidden content, this deep Web is an important yet largely-unexplored frontier for information search– While we have understood the surface web relatively well, with various surveys (*e.g.*, [3, 4]), how is the deep Web different? This article reports our survey of the deep Web, studying the scale, subject distribution, search-engine coverage, and other access characteristics of online databases. We note that, while the 2000 study [2] opens interest in this area, it focuses on only the scale aspect, and its result from *overlap analysis* tends to underestimate (as [2] also acknowledges). In overlap analysis, we need to estimate thenumber of deep Web sites by exploiting two search engines.

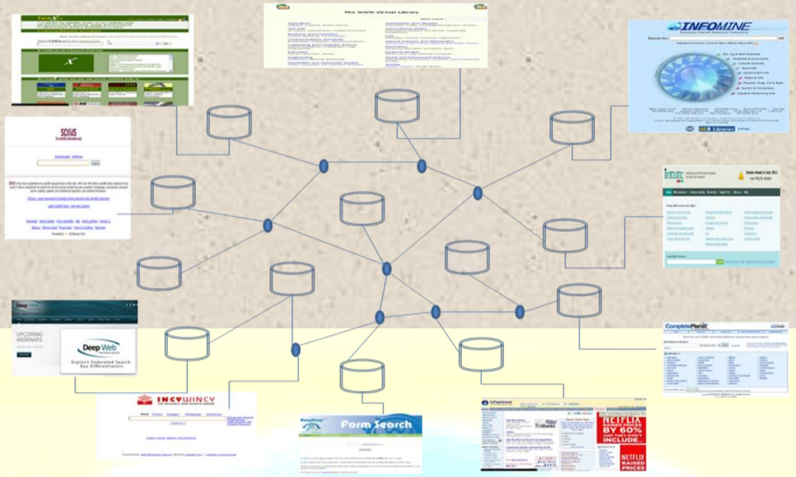


Fig 1.Deep web source

We use our customized schema extraction algorithm in literature [5] to extract the schema of entity on the Deep Web. In this paper, we will focus on studying an effective approach to identifying the corresponding entities on the Deep Web. The present entities matching methods always identify the corresponding entities based on predefined rules, which is poorly adaptive to different application domains. Furthermore, the weight of each attribute of entity cannot be estimated accurately by predefined rules. Neural network is trained, not programmed. The neural network learns the patterns directly from instances without prior knowledge of regularities and this characteristic make it possible to solve the entities matching effectively on the Deep Web which has more uncertainty than relational database. Based on our analysis, we propose a neural network approach to identifying the matching entities on the Deep Web.

II. METHODOLOGY:

Interface schema extraction to formulate it is easier for customer to understand and use query interface, a designer usually integrates various types of visual characteristics into the interfaces, such as the layout, appearance and main location. Thus, our system uses the pre-clustering the attributes of query interface for schema extraction. Initially, we will use the approach in literature [5] to extract the schema of query interface on the Deep Web by using our proposed grouping patterns with good clustering capability. Then determine the label block by clustering the similarity of multiple text blocks according to their location, lyout and appearance. At last, we will give the attributes of each label structure storage. Given multiple data sources with interfaces, the goal of our algorithm is to find the semantic matching between input attributes and output attributes among multiple data sources. Thus, our approach has two main components. The first component aims at finding valid instances for input attributes. Having such instances allows us to obtain instances of output attributes, and thus to find output schemas. Once we have instances of input and output attributes, the second component identifies the semantic correspondence between the set of all input and output attributes across different data sources.

I. Interface Integration: Through attribute analysis query interfaces is the main way of interface integration. Schema matching is a critical problem for integrating heterogeneous information sources. In recent ten years, a great deal of researches had been done in the area of schema matching [[5] [6] .In this paper, these grouping patterns will be used in the first step of our schema extraction algorithm when pre-clustering the attributes of query interface. First, we find frequent attributes in the input attribute groups. Second, Group discovery: We mine positively correlated attributes to form potential attribute groups. Third, partition the attributes into concepts; cluster the concepts by calculating the similarity of each two concepts. At last, we develop a strategy to reasonably rank to discover matching, and then use a greedy matching selection algorithm to select the final matching results. The similarity of concepts, based on two component similarities, the linguistic similarity and the domain similarity can be calculated.

II. Query Process Translation: After building a comprehensive interface, users only require submitting their requirements in it, and the query will be dispatched to all the local interfaces to retrieve manuscript information. We should select suitable web databases to query translate at first, thus we’ll spend as little as possible cost of access, but get small enough redundancy and specific query results. Query Process Translation involves many issues, such as, constrained mapping, schema matching, query rewriting, and so on. Predicate correspondence often exists in the same type of predicate templates, and this feature has no related to the specific web database and specific domain. Because of the difference of schema, constraint and query ability, the query translation can only be an approximate process. To maximize the use of the database query capabilities, we consider three main factors: attribute constraint, predicate mapping and query ability.

III. Data Merging: Consequences from different databases need to be merged and eliminated duplication, then feedback to users. We still use the Entities identification using back propagation Neural network on the deep web [2] [3]. Our system pre-forms an add semantic procedures ,which can learn the corresponding relations between the data and semantics, on a sample pages through machine learning, then builds mapping between each Web databases' schemas. These two relations in a complementary way are used to achieve the addition of semantic information. However, as the poor level of the structure of pages, the experimental results are not satisfactory. Currently, the research on entity identification is also immature, only launches on relation schema and semi-structured XML schema. The key problem is to establish the comparison between mapping relations and values of entity attributes. Our system uses an instance-based method to final results merging. This method includes data types, high frequency words related with attribute.

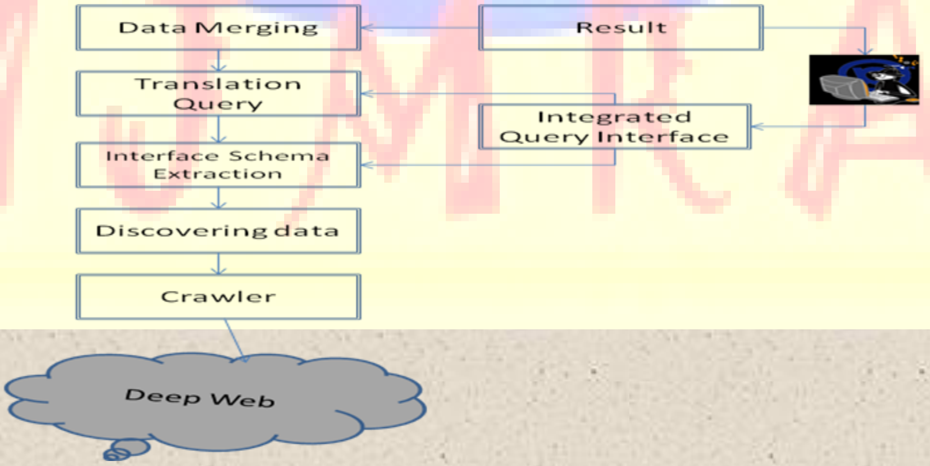


Fig 2: The architecture of Deep Web data integration system

III. SCHEMA EXTRACTION OF ENTITIES ON THE DEEP WEB:

The entity on the Deep Web represents the attribute of web database which has standard form of schema. In order to advance the performance of identifying the corresponding entities on the Deep Web, the schema of entities should be extracted firstly. We will use the approach in literature [5] to extract the schema of query interface on the Deep Web by using our proposed grouping patterns with good clustering capability. These grouping patterns are presented in turn according to their importance for clustering attributes of query interface as follows:

(I)Alignment-based patterns

(II)Control button-based patterns.

(III)Semantic-based patterns

(IV)Attribute value-based patterns

(V)Separator-based patterns

(VI)Indentation-based patterns

These grouping patterns will be used in the first step of our schema extraction algorithm when pre-clustering the attributes of query interface. The schema extraction algorithm is presented below. About the detailed algorithm description, please refer to literature [5]

```
Extr(S) →H:
Input: S, a set of attributes on a query interface
Output: H, a hierarchical clustering
Step1. Utilize our pre-clustering algorithm MPreCluster to obtain partial clustering:
        P←MPreCluster(S)
Step2. Form initial clustering
      C← {{f} |f ϵ S}
Step3. Repeat the following steps until all clusters in P are visited
```

```
        a. Select one unvisited cluster M in P
        b. Q←NCluster (M)
        c. Merge attributes in P according to Q
Step4. Merge clusters in C according to P:
        C’ ← Merge(C, P)
Step5. Merge all candidate singleton clusters in C’ via nway constrained merging operation
/Obtain complete clustering by steps 4 and step 5 */
Step6. H ← the hierarchical clustering output by step5
Step7. Return H
NCluster (M) →M’:
Input: M, a cluster
Output: M’, a merged cluster
Step1. Obtain a copy of M, denoted as M’
Step2. Repeat the following steps until no attributes with the minimum distance in M or M is null:
/* perform a n-way merging operation */
a. Find two attributes f1, f2 with the minimum distance via distance function of Definition 4 in one cluster N, N є M
b. Expand them into a proximity set: X←OBTAINPROXIMITYSET (f1, f2, N)
c. Merge clusters in M’ according to X
d. Remove the attributes in X from M
Step3. Return M’
```

By using this schema extraction algorithm, we can obtain the schema of entities on the Deep Web by extracting the schema of query interface on the HTML page.

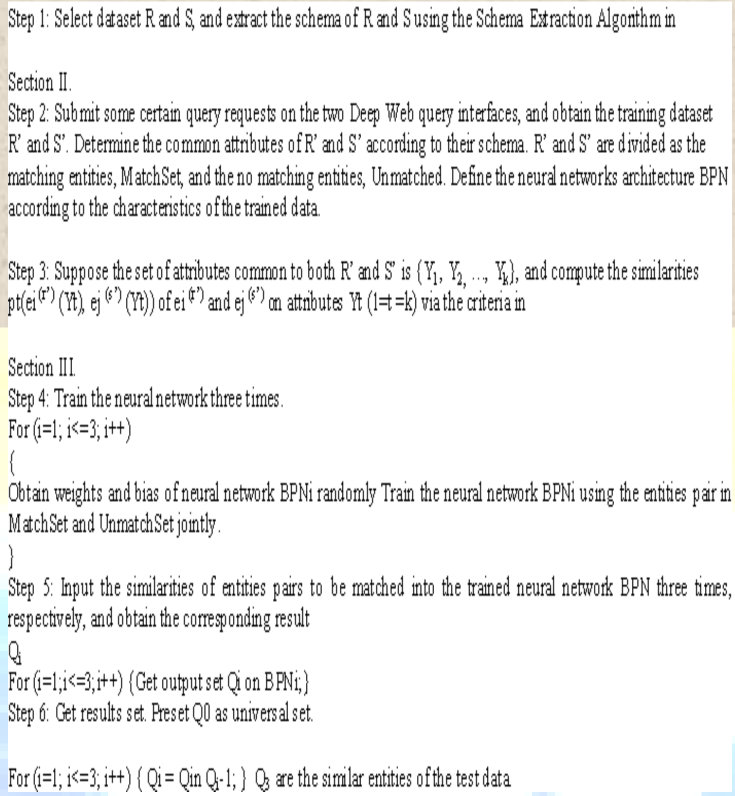
IV. ENTITIES IDENTIFICATION USING BACK PROPAGATION. NEURAL NETWORK ON THE DEEP WEB:

I. The neural network model for entities identification: In literatures [5, 7], we solved the attributes matching problem across heterogeneous databases by using neural network. Since the number of metadata describing the attributes usually is less than 15, the neural network training and identification process is effective. Unlike the attributes matching, the training process for entities matching will become impossible, if we use each entity in the training set to train the neural network because of the large number of entities on the Deep Web. At the same time, the string information cannot be quantified and the difference between strings such as “pressman” and “galvin” cannot be discriminated by using neural network directly. So we consider constructing the training set using the matching entities and some randomly chose nonmatching entities alternatively. Even if in two web databases with large number of entities belong to the same domain, the number of corresponding entities is much less than that of all entities. So we firstly compute the similarity on each attribute of every pair of matching entities and some nonmatching entities (the proportion is about 1:10), and then use these similarities to train neural network. The architecture of the neural network is presented. The input vector for neural network is composed of the similarities on attributes of every pair of matching and nonmatching entities, the outputs represent the similarity of entities. After the trained neural network obtained the predefined training precision, it can be used to evaluate the similarity of entities on the Deep Web.

II. Entity identification algorithm is based on back propagation neural network. In this section, we give the criteria to evaluate the similarity of attribute values and present the entities identification algorithm.

a) Evaluation for similarity of attribute values**:** With the existence of different naming conventions coding schemes, or precision levels used in different applications, it is difficult to estimate the similarity of instance values for attributes. Now we establish the criteria to evaluate similarity of attributes according to the data types of attributes. For numeric data types, we evaluate the similarity of attribute values by considering the relative distance between the two attribute values. The distance between two values s1 and s2 can be expressed as D=|s2-s1 |/ max (s1, s2), and the similarity of these two values is 1-D. For binary data types, the value is either 1 or 0. If the values of binary data types match, the similarity is 1, otherwise 0. For nominal data types, the similarity of values can be defined beforehand due to their value domains are definite. Usually, the distance between two values of nominal data types is elicited from the users and the domain experts. For character data types, the similarity between two strings may be expressed as the maximum substring over the total number of characters in the longer string. For example, the similarity of names Martin T. and Howard B. is 7/9 —the number of characters in maximum substring is seven (include blank space) and the number of characters in the longer string is nine.

b) Entities identification algorithm using neural network on the Deep Web- The main ideas of our algorithm are: The similarities on each attributes of every pair of matching and nonmatching entities in the sample train set are computed. Then the neural networks are trained two or three times using the vector composed of the above similarities [2]. Finally, the similarities of each pair of entities on the Deep Web to be matched are input into the neural network and the similarity of entities is evaluated according to the output. Firstly, we choose two Deep Web sites belonging to the same domain as the training dataset. Let R represent the entity set from one Deep Web, and S represent the entity set from another Deep Web. R and S belong to the same domain. Let ei(r) represent the entity in R, and ej(s) represent the entity in S. The similarity of e_i(r) and e_j(s) is denoted by pt (e_i(r), e_j(s)).



Step 1: Select dataset R and S, and extract the schema of R and S using the Schema Extraction Algorithm in

Section II.

Step 2: Submit some certain query requests on the two Deep Web query interfaces, and obtain the training dataset R' and S'. Determine the common attributes of R' and S' according to their schema. R' and S' are divided as the matching entities, MatchSet, and the no matching entities, Unmatched. Define the neural networks architecture BPN according to the characteristics of the trained data.

Step 3: Suppose the set of attributes common to both R' and S' is $\{Y_1, Y_2, \ldots, Y_k\}$, and compute the similarities pt(ei$^{(r')}$ (Yt), ej$^{(s')}$ (Yt)) of ei$^{(r')}$ and ej$^{(s')}$ on attributes Yt (1=t =k) via the criteria in

Section III.

Step 4: Train the neural network three times.

For (i=1; i<=3; i++)

{

Obtain weights and bias of neural network BPNi randomly Train the neural network BPNi using the entities pair in MatchSet and UnmatchSet jointly.

}

Step 5: Input the similarities of entities pairs to be matched into the trained neural network BPN three times, respectively, and obtain the corresponding result

Q_i

For (i=1;i<=3;i++) {Get output set Qi on BPNi;}

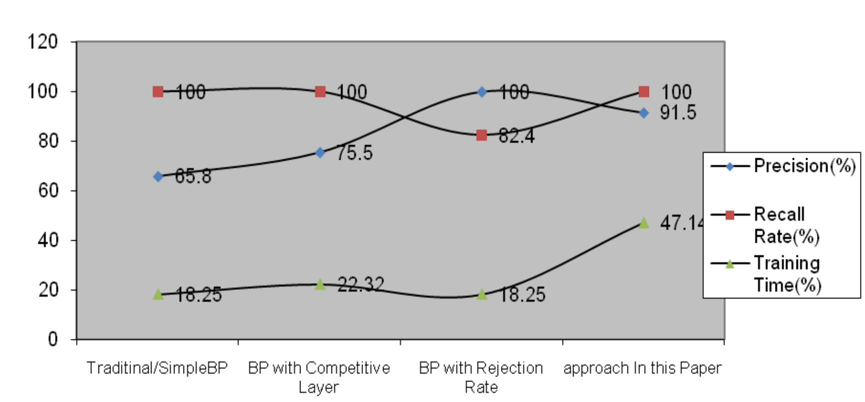
Step 6: Get results set. Preset Q0 as universal set.

For (i=1; i<=3; i++) { Qi = Qin Q_i-1; } Q_3 are the similar entities of the test data

Algorithm

V.<u>RESULT AND DISCUSSION:</u>

Initially, the Section III in Schema extraction algorithm is applied to extract the schema of the query interfaces which reflect the schema of the entities on the invisible web. We can get the universal attributes {Author, Title, Publisher, Keywords, Price} for the two query interfaces from their schema. Next, the training dataset can be obtained by submitting query as Author='qiang' and Binding Type="Hardcover" on the book domain. There are 32 pieces of result entities and 341 pieces of result entities on the first and second query interfaces, respectively. The 373 pieces of entities are divided into MatchSet and UnmatchSet. The neural network is trained using datasets MatchSet and UnmatchSet. Due to the obvious difference of the two Web databases, it can testify our algorithms effectively. Finally, we submit query on another query interface in the book domain dataset and get the entities. Then the entities identification algorithm using neural network on the Deep Web is applied to determine the corresponding entities between this query interface and the first query interface or the second query interface. The effectiveness of Schema Extraction Algorithm on query interface has been evaluated in literature [5]. Here we evaluate the Entities Identification Algorithm Using neural network mainly by comparing the training time, the precision and recall of entities matching. The experimental results are presented on the chart. From the comparison of different approaches, it shows that our proposed approach is very effective. The average precision is improved about 20% while the recall keeps stable. Although the training time increases due to the three-time training process, it can be compensated by the higher accuracy. Our approach is very effectiveness for entities identification due to the following strategy: for the information consistent with the training data, the output results are always keeping steady accordingly on different trained neural networks. On the other hand, for the information inconsistent with the training data, it will produce the inconsistent output results on different trained neural networks. Our proposed approach can remove the inconsistent output effectively.



Experimental Result

VI. CONCLUSION:

Compared with the asymmetrical data format on the surface web, the data on the invisible web has certain schema. In order to improve the accuracy for entities identification on the invisible web, we firstly consider extracting the schema of the query interface which indicates the schema of the web database. Then we proposed entities identification algorithm using neural network to determine the corresponding entities on the invisible web. Unlike traditional rules, neural network is trained, not programmed, which is very suitable to entities identification on the invisible web with some uncertainty. The experimental results show that our anticipated algorithm is very effective.

References:

- Jufeng Yang Guangshun Shi Yan Zheng Qingren Wang," Data Extraction from Deep Web Pages" 0-7695-3072-9/07 2007 .
- Baohua Qiang, Chunming Wu, Long Zhang," Entities Identification on the Deep Web Using Neural Network" International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery-2010.
- Xin Zhong,Yuchen Fu, Quan Liu,Yan Wang,Zhiming Cui," A Deep Web Data Integration System For Book Searching Domain" Workshop on Intelligent Information Technology Application-2007.
- M. Bergman. The Deep Web: Surfacing the hidden value. BrightPlanet.com(http://www.brightplanet.com/technology), 2000.
- W. Wu, C. Yu, A. Doan, and W. Meng. An interactive clustering-based approach to integrating source query interfaces on the Deep Web. In Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data (SIGMOD'04), pages 95–106, 2004.
- Bao-hua Qiang, Jian-qing Xi, Ling Chen. Effective Schema Extraction of Query Interfaces on the Deep Web. The 4th International Conference on Natural Computation and the 5th International Conference on Fuzzy Systems and Knowledge Discovery, Jinan, China, Aug. 25-28, 2008. IEEE, in press
- UIUC Web integration repository: http://metaquerier.cs.uiuc.edu/repository/.
- Marcus P. Zillman. Deep Web Research 2008, Published on November 24, 2007.
- K. Chang, B. He, C. Li, M. Patel, and Z. Zhang. Structured databases on the Web: Observations and implications. ACM SIGMOD Record, 33(3): 61–70, 2004.

- H. He, W. Meng, C. Yu, and Z. Wu. WISE-Integrator: An automaticintegrator of Web search interfaces for e-commerce. In Proceedings of the 29th International Conference on Very Large Data Bases (VLDB’03), pages 357–368, 2003.
- BrightPlanet.com. The deep web: Surfacing hidden value. Accessible at http://brightplanet.com, July 2000