International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:1, 2009

Deep Web Content Mining

Shohreh Ajoudanian, and Mohammad Davarpanah Jazi

***Abstract*—**The rapid expansion of the web is causing the constant growth of information, leading to several problems such as increased difficulty of extracting potentially useful knowledge. Web content mining confronts this problem gathering explicit information from different web sites for its access and knowledge discovery. Query interfaces of web databases share common building blocks. After extracting information with parsing approach, we use a new data mining algorithm to match a large number of schemas in databases at a time. Using this algorithm increases the speed of information matching. In addition, instead of simple 1:1 matching, they do complex (m:n) matching between query interfaces. In this paper we present a novel correlation mining algorithm that matches correlated attributes with smaller cost. This algorithm uses Jaccard measure to distinguish positive and negative correlated attributes. After that, system matches the user query with different query interfaces in special domain and finally chooses the nearest query interface with user query to answer to it.

***Keywords*—**Content mining, complex matching, correlation mining, information extraction.

I. Introduction

WITH the explosive growth of information sources available on the world wide web, it has become increasingly necessary for users to utilize automated tools in finding the desired information resources, and to track and analyze their usage patterns. Web has been deepened by online databases. In April 2004, online databases with the virtually unlimited amount of information sources and deep web [1], [2] estimated to be 450000. The deep web is clearly an important frontier for data integration. On the deep web, numerous online databases provide dynamic query based data access through their query forms or query interfaces. Fig. 1 shows the query interface of Book database of amazon.com web site. Users fill this form to access to the Book database. We must understand query interfaces to understand what query capabilities are supported by source or what information can be extracted from online databases.

Web content mining confronts this problem gathering explicit information from different web sites for its access and knowledge discovery. Basically, web mining is concerned with the use of data mining techniques to automatically discover and extract information from world wide web documents and services.

Shohreh Ajoudanian is with the Computer Engineering Department, Foulad Institute of Technology, Fouladshahr, Iran (e-mail: shajoudanian@gmail.com).

Mohammad Davarpanah Jazi is with the Electrical and Computer Engineering Department, Isfahan University of Technology, Iran (e-mail: mdjazi@cc.iut.ac.ir).

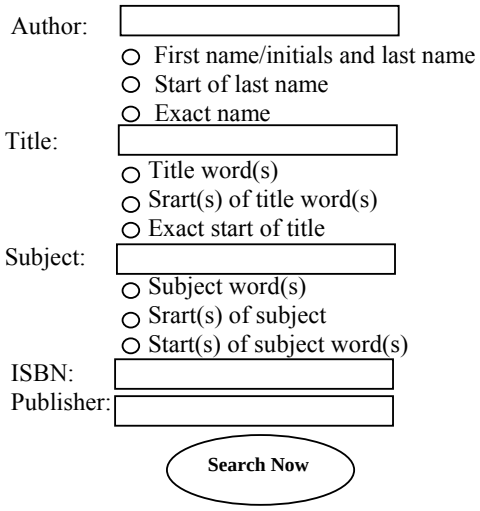

Fig. 1 Query interface of book domain of amazon.com web site.

This can help to discover global as well as local structure within and between web pages. For example, to match information between some web pages in the same domain, we can use web content mining techniques. Web mining techniques can be applied on different data structures, from fully structured data like database tables to unstructured data like free form text. This means that web mining is an invaluable help in the transformation from human understandable contents to machine understandable semantics.

In this paper we present a system that carries out the work in two essential steps. This system with crawling from one hyperlink to another, mines contents of query interfaces and after extracting information with clustering techniques, put them in special domains. System matches user query with different query interfaces in special domain and finally chooses the nearest query interfaces with user query.

This paper is structured as follows. In section II, we describe how to extract information from query interfaces, in section III we apply a clustering technique with a heuristic function to data which is extracted from section II and put them in correct cluster. In section IV we present a new algorithm that matches extracted information in different query interfaces in a special domain and chooses the best query forms to supply the best answer to the user query. In section V we discuss some related work and finally we conclude in section VI.

II. Web Content Mining

At this time crawlers cannot effectively query online databases, such data are invisible to search engines, and thus the Deep Web remains largely hidden from users. To enable effective access to databases on the web, since April 2002, some systems have been presented [3], [4], we continue their work with the system that is present in this paper.

To extract information from deep web that is a large collection of dynamic queryable databases, we need a system that can extract automatically. For this purpose we use web content mining techniques that uses XML version of HTML query interfaces. Web content mining is a form of text mining and can take advantage of the semi-structured nature of web page text. Query interfaces share similar or common query patterns. For instance, a frequently used pattern is a text followed by a selection list with numeric values.

The HTML tags of today's web pages, and even more so the XML markup of tomorrow's web pages, bear information that concerns not only layout, but also logical structure. HTML format might be invalid and cause problems in extracting information. In most of previous works [3] extracting information is performed from HTML pages and some of them firstly is converted invalid HTML pages to valid HTML pages and then extracting process is applied but in this paper we use XML format of web pages for extracting information. Extractor system which is presented in this paper gets XML pages as an input and can access to XML tags in documents with XML DOM API. DOM[1] is a standard language that gets a web page as an input and shows it in a structured tree from interfaces, objects and relations between them as an output. A sample DOM tree shows in Fig. 2 that is the extracted form of a sample query interface.

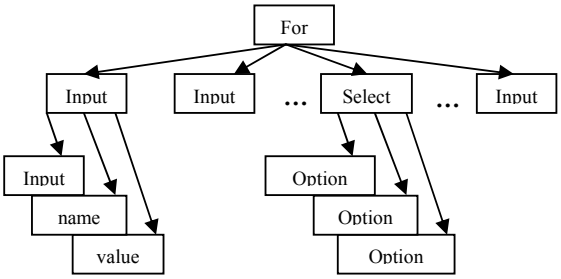

Fig. 2 A sample DOM tree

All of the extracted information from query interfaces is stored in a database that is used in the next step of system.

III. Clustering Extracted Data

Clustering of web search results has been in focus of the information retrieval community since the early days of the web [5], [6]. Web pages belong to the same cluster if they are similar in content. After extracting information from query interfaces in step 1, we must put them in true subject domains. We do clustering with an algorithm that uses a heuristic function that estimate the amount of similarity between web pages. First we suppose each file that is created in step one to be a separate cluster, with the use of heuristic function if the system recognizes that two clusters according to their contents are similar, merge them and put their query interface URLs in one cluster. System continues this process until clusters remain in a steady state. Algorithm that has been used for clustering is shown in Fig. 3.

Two types of heuristics can be proposed in the web domain are topology-driven and content-driven. Topology-driven heuristics are based on the layout of the web graph, while content-driven heuristics are based on features extracted from the interior of web pages. Heuristic function that we use in this paper increment the value of priority variable by one, if it finds two similar fields in the query interfaces, in this situation the chance of being in one cluster of these two query interfaces is increased.

Input:
$S = \{s_1, s_2, ..., s_n\}$ – URLs of source pages
Output:
Clusters $C_1, \ldots . C_k$
Initialize singleton clusters $C_i \leftarrow \{s_i\}$
While Clusters aren't steady
  For each pairs (C_i, C_j)
    If ($C_i \neq C_j$) **AND** Heuristic(C_i and C_j) >=4 **Then**
      $C_i \leftarrow C_i \cup C_j$

Fig. 3 Algorithm of clustering query interfaces

Two query interfaces compare completely. The experience shows that if the value of priority variable be equal or greater than 4, we can put them in one cluster. To examine the accuracy of presented heuristic function we use a data set that has 150 query interfaces in different domains. Since every online database prepares structured information in one domain, their query interfaces are very similar and this heuristic function works with a good accuracy about them. You can see part of the result of examination in Table I.

TABLE I
Part of the Results of Applying Heuristic Function on 150 Query Interfaces in Different Domain

Cluster	No. of putting QI	Errors	Domain
C1	20	0	Book
C2	21	2	Car
C3	20	0	Airfare
C4	17	1	Hotel

IV. Matching Extracted Information

After extracting information from online databases, to reach to the knowledge between query interfaces it is essential to match this information until finding matched attributes between them. Comparing match attributes, we reach to the desired knowledge and can response to the user query. Suppose that a user is looking for a book with "Web Mining" name and she is also looking for a store that can buy this book with the cheapest price. This system, after extracting

[1] Document Object Model

information from some query interfaces in book domain, finds matching information between them and after comparing them sends the best answer to the user query.

For complex matching (matching m attributes in a query interface with n attributes in another query interface) extracted data that are in the same domain we use this idea that query forms have frequently patterns (co-occurrence patterns) and for this reason we use solutions based on that data mining techniques.

Basically there are two types of attributes in query forms. Grouping attributes and synonym attributes. Grouping attributes are attributes that usually come with each other in a query forms. For example we commonly see First name and Last name attributes come together in a query form. Synonym attributes are attributes that are rarely come with each other in a query form like Number of Tickets and Number of Passengers in airfare domain.

A. Correlation Mining Approach

For matching information between some query interfaces we use correlation mining approach. In past years correlation mining approach is used for matching attributes [1], [10], [11], [12]. In this paper we use this approach with a new algorithm that finds correlated attributes in query interfaces faster than the old version algorithm.

For decreasing cost of correlation mining algorithm, before applying this algorithm on extracted information, in the previous step, attributes that are completely identical are recognized and is notified as correlated attributes. This attributes will be removed from the list of attributes that we send them to the algorithm.

We define correlation mining problem as follows: with a set of schemas that are in same domain we want to find correlated attributes. Correlation mining approach needs a measure that computes correlated measure between some items. In the previous works measures like Lift, χ^2, confidence and cosine [12] are used and each one has some advantages to others. In the presented algorithm in this paper, we use Jaccard measure. In general Jaccard measure is capable to measure similarity degree between some items. Jaccard measure is also capable to recognize positive and negative correlations between attributes. Grouping attributes have positive correlation and synonym attributes have negative correlation.

B. Correlation Mining Algorithm

In this algorithm notwithstanding to the commonly correlation mining algorithm, a Priori, that uses sets theory and has a complex implementation, we store input schemas in arrays. All of the attributes are posited in the columns and some frequent schemas posited in the rows. If an attribute is in a frequent schema, a 1 is stored in related cell and otherwise a 0 is stored in it. Finally each two columns (for example c1 and c2 columns) are compared with each other. If two peer cells in a row is one, we add one unit to n(c1∩c2) and if two or one of them is one, we add one unit to n(c1∪c2). After all, with the use of Jaccard measure in equation 1, we calculate correlation degree of the two attributes.

$$\text{Jaccard} = \frac{n(C1 \cap C2)}{n(C1 \cup C2)} \quad (1)$$

This equation returns a number between 0 and 1 as a result. Returning one means that peer attributes are grouping attributes and commonly come with each other in a query interface. Returning zero means that peer attributes are synonym attributes and rarely come with each other in a query interface. If it returns a number between 0 and 1 it actually returns the probability that two attributes come with each other in a query interface. If this number is near to zero shows negative correlation and if it is near to one, it shows positive correlation. This algorithm is shown in Fig. 4.

1. **Begin**
2. /*enter inputs in array*/
3. **for each** attributes in input schemas (as columns c_i)
4. **for each** frequent schemas (as rows r_j)
5. **if** attributes ∈ frequent schema **then**
6. matrix[c_i,r_j] = 1
7. **else**
8. matrix[c_i,r_j] = 0
9. /*compare columns*/
10. **for each** attributes in input schemas (as columns c_i)
11. **for each** attributes in input schemas (as columns c_{i+1})
12. **for each** frequent schemas (as rows r_j)
13. **if** matrix[c_i,r_j] = 1 **AND** matrix[c_{i+1},r_j] = 1 **then**
14. $n(c1 \cap c2) = n(c1 \cap c2) + 1$
15. **else if** matrix[c_i,r_j] = 1 **OR** matrix[c_{i+1},r_j] = 1 **then**
16. $n(c1 \cup c2) = n(c1 \cup c2) + 1$
17. $J = \frac{n(c1 \cap c2)}{n(c1 \cup c2)}$
18. **if** J=1 **then**
19. **return** c_i and c_{i+1} are grouping attributes
20. **if** J=0 **then**
21. **return** c_i and c_{i+1} are synonym attributes
22. **End**

Fig. 4 New correlation mining algorithm

Algorithm inputs are a set of common schemas in a special domain, attributes of query interfaces that must be matched and Jaccard measure and the outputs of the algorithm are matching attributes in web query interfaces. This algorithm is capable to compare some attributes with each other too.

In the following you can see an example of this algorithm in book domain. As an input we send attributes of 5 frequently schemas in book domain to the algorithm. The attributes of this frequently query forms reside in array's rows. These attributes are stated below.

QI_1: www.americanbookcenter.com = {author, title, subject, ISBN, publisher, reader age, language}
QI_2: www.abbeys.com = {author, title, category, price range, ISBN, publication date}
QI_3: www.abebooks.com= {author, title, subject, publisher, keywords, ISBN, price, attributes}
QI_4: www.bookgallery.com = {Last name, First name, title, other keywords, ISBN, category}

QI_5: www.bookplace.com= {title search, author search, keyword search, ISBN search, publisher search}

As a second input we send extracted attributes of query interfaces that must be matched. These query interfaces are www.amazon.com and www.bookery.com. Attributes that extracted from them are as follows.

T1: www.amazon.com = {author, title, subject, ISBN, publisher, reader age, language}
T2: www.bookery.com = {Last name, First name, title, other keywords, category}

As stated in section IV. A, For decreasing the cost of correlation mining algorithm, before applying this algorithm to extracted information, attributes that are completely similar are recognized and notified as correlated attributes. This attributes will be removed from the list of attributes that we send them to the algorithm. In this example Title attribute is used in two query forms and thus we don't reside it in array's columns. The resulted array is as Table II. Other attributes in these two query interfaces reside in array's columns.

After that, each column is compared with other columns and according to their results, the status of attributes is declared. For example about first name and last name attributes Jaccard measure returns one as a result and it shows that they are grouping attributes and commonly come with each other in a query interface. For example about author and first name attributes it returns 0 that means they are synonym and rarely come with each other in a query interface.

This approach returns good semantic matching between attributes in online databases but it has some false information too. For instance in the above example in the book domain in addition to true result author = {First name, Last name}, we also encounter with the false result subject = {First name, Last name}. We call this situation conflict. For resolving this problem in the previous works, distribution relation has been used.

For example because both author and subject attributes are matched with {first name, last name}. Thus subject and author must be equivalent. But algorithm doesn't show this. Thus one of these matching is wrong and must be removed. We do this by means of function that select ones with upper Jaccard measure result, among matching attributes.

TABLE II
A SAMPLE ARRAY WITH INPUT INFORMATION

	author	Subject	ISBN	publisher	Reader age	Language	Last name	First name	Other keywords	category
QI1	1	1	1	1	1	1	0	0	0	0
QI2	1	0	1	0	0	0	0	0	0	1
QI3	1	1	1	1	0	0	0	0	1	0
QI4	0	0	1	0	0	0	1	1	1	1
QI5	1	0	1	1	0	0	0	0	1	0

V. RELATED WORK

In the face of that correlation mining technique that we use in this paper can find complex matching between attributes faster and more accurate than previous works that uses grammatical methods but for improving the result, we can improve the algorithm. The heart of this algorithm is its measure. With the use of suitable measure, the result algorithm tends to be better.

In this paper we use Jaccard measure in the algorithm. This measure is capable to find correlation between attributes accurately. In [1] a new measure is introduced that can find correlated attributes in a good manner but it needs a threshold to find matching attributes. Defining such threshold accurately usually is very difficult. Jaccard measure doesn't need to define a threshold. In the future work with a more accurate measure we can have a better algorithm.

For resolving conflict between 1:1 attributes matching, we can use test samples [7]. For example in two query interfaces that algorithm recognizes that subject = author, we can verify that what they return if user ask similar query. If the result is equal then they are matched attributes, otherwise they are not equal and one of them must be removed.

VI. CONCLUDE

In this paper a system that extracts and matches information in the deep web is presented. This system does the task of matching attributes in online databases with a new correlation mining approach. This system does its work in two essential steps automatically. In the first step it extracts information from query interfaces and in the second step it matches them. In general it does deep web content mining and complex matching between attributes that are extracted from query interfaces. It uses web content mining approach to extract information from web based databases. After that for clustering web pages in subject domains, it uses clustering technique with a heuristic function for extracting attributes. Finally with the use of correlation mining algorithm we match the extracted attributes in a special domain. We use Jaccard measure in this algorithm to find grouping and synonym attributes in a faster and more accurate manner.

REFERENCES

[1] Bin He, Kevin chen-chuan chang; "Automatic complex schema matching across web query interfaces: A correlation mining approach"; ACM Transactions on Databases Systems; Vol. 31; No.1; Pages 1-45; March 2006.
[2] Michael K. Bergman; "The Deep Web: Surfacing Hidden Value"; www.BrightPlanet.com; Pages 1-5; 2001.

[3] Kevin chen-chuan chang; "Toward Large Scale Integration: Building a Metaquerier over databases on the web"; VLDB Journal; 2005.
[4] Zhen Zhang; "Light-weight Domain–based Form Assistant: Querying web databases on the fly "; 31st VLDB Conference; Trondheim Norway; 2005.
[5] M. A. Hearst and J. O. Pederson; "Reexamining the cluster hypothesis: Scatter/gather on retrieval results"; In Proceedings of SIGIR; Pages 76-84; 1996.
[6] O. Zamir and O. Etzioni; "Web document clustering: a feasibility demonstration"; In Proceedings of SIGIR; 1998.
[7] Sh. Ajoudanian, M. Davarpanah Jazi, and M. Saraee; "Discovering Knowledge from Deep Web Databases using Correlation Mining Approach"; IDMC Conference; Iran; 2007.
[8] Bin He, Kevin chen-chuan chang; "Statistical schema matching across web query interfaces"; In SIGMOD Conferences; 2003.
[9] E. Rahm, P. A. Bernstein;"A survey of approaches to automatic schema matching"; VLDB Journal; no 10; Pages 334-350; 2001.
[10] Agrawal R., Imielinski T., Swami A. N.; "Mining association rules between sets of items in large databases"; In SIGMOD Conference; 1993.
[11] Y-K Lee, W-Y Kim, Y. D. Cai; "Efficient mining of correlated patterns"; In SIGMOD Conference; 2003.
[12] S. Brin, R. Motwani, C. Silverstein; "Beyond market baskets: generalizing association rules to correlations"; In SIGMOD Conference; 1997.