

Structured Database Integration over the Web

Xuefeng Xian^{1,2}, Yuanfeng Yang^{1,2}, Yinghuang Liang¹, Ligang Fang¹, and Zhiming Cui^{1,2*}

¹JiangSu Province Support Software Engineering R&D Center for Modern
Information Technology Application in Enterprise, Suzhou, China

²The Institute of Intelligent Information Processing and Application, Soochow University, Suzhou, China
Email: xianxuefeng@jssvc.edu.cn; szzmcui@suda.edu.cn

Abstract— With the rapid development of Web, there are more and more structured web database available for users to access. At the same time, domain searchers often have difficulties in finding the right web database. In this paper, we study how we can build an effective web database integration system with the aim of making the system contain as much important data as possible and the least degree of query cost. In this paper, we presents a method for integration of structured web database. We experimentally results on real web databases indicate that our method is highly efficient .

Index Terms—web database; data integration; source selection

I. INTRODUCTION

More and more databases are becoming Web accessible. This through form-based search interfaces. We is often called this kind of web data “deep web”. the deep web is believed to be possibly larger than the surface web, and typically has very high-quality contents [1]. According to the survey [2] released by UIUC in 2004, there are more than 300,000 deep web sites and 450,000 query interfaces available at that time, and the two figures are still increasing rapidly. In this paper, our research focuses on structured databases on the Web, which return structured objects with attribute-value pairs(e.g., a Book source like amazon.com returns books with author,title, etc.).

In order to assist users in accessing the Web data in the deep web, many efforts have focused on building the deep web data integration system(such as metasearch engine) that mediates many deep web databases and provides a single access point for users[3,4]. Given a user's query, the integration system determines which databases are the most likely to be relevant, directs the user's query to those databases and collects the search results back to the user. Given this scenario, we note that an effective integration system needs to do web databases selections twice in Internet-scale deep web data integration tasks.

1.In Internet-scale deep web data integration tasks, where there may be hundreds or thousands of web databases providing data of relevance to a particular domain. An integration system cannot possibly involve in all of them, so a few sets of web databases must be selected to build an integration system.

2.Based on the user's query, the integration system has to select a set of databases which are most relevant from all integrated web databases, so it can direct the query to those databases. Recently, main efforts have been focused

on automatically selecting the most relevant databases to a user's query[3,5].

There has been a few researches on the problem of web database selection for building deep web integration system. The problem of source selection is modeled as an optimization problem and solved by using the data envelopment analysis technique[6]. The solution is computationally expensive so it does not apply to Internet-scale data integration. In [7], data source is selected by the user depending on several subjective and objective criteria. Because it depends on some subjective preferences of the user, it is difficult to automate web database selection.

This paper presents a utility maximization model to the resource selection problem of deep web data integration by treating them as optimization goals. The model is to estimate the utility of the web database bringing to a given status of an integration system by integrating it. With the estimated utility information, we select and integrate web databases in an iterative manner, where web databases are integrated incrementally. This approach selects a maximal utility web database from the set of candidate web databases to integrate each time. After each web database is integrated, we update the status of integration system and recompute the next maximal utility web database to integrate. The integration system obtains maximal utility by using the incremental integrate manner.

We describe a detailed experimental evaluation on real deep web databases shows that our method is highly efficient.

The remaining of this paper is organized as follows. In section 2, describes the new utility maximization model. Section 3 describes how to use utility maximization model for selecting web database to integrate. Section 4 presents our experimental results for web database selection and integration. We conclude in section 5.

II. UTILITY MAXIMIZATION MODEL

The utility maximization model is based on estimating the utility of the web database bringing to a given status of deep web integration system by integrating it. In this section, we describe how the utility of web database is estimated.

Suppose we are given an integration system D and a set of candidate web databases $S = \{s_1, s_2, \dots, s_n\}$. Everything is a double-edged sword, given a candidate web database s_i , if the system integrates s_i , the integration system D would be affected by the positive

and negative utility of s_i . In this paper, the positive and negative utility of s_i bringing to D by integrating s_i are respectively denoted by $D_{s_i}^+$ and $D_{s_i}^-$.

Hence, the utility of s_i bringing to D by integrating s_i can be expressed as the following difference:

$$Utility(D, s_i) = D_{s_i}^+ w_1 - D_{s_i}^- w_2 \quad (1)$$

Where $0 \leq \{w_1, w_2\} \leq 1$ and $w_1 + w_2 = 1$.

In next two subsection, we show how we measure $D_{s_i}^+$ and $D_{s_i}^-$ respectively.

A. Positive Utility

In this paper, $D_{s_i}^+$ can be expressed by the importance of new data that add to the integration system by integrating s_i . In this paper, the importance of new data is expressed by correlation of the degree of these new data with greater importance query. So the more volume of new data are involved in more queries with greater importance, s_i bring more positive utility to a given status of integration system D .

1) *Estimating $D_{s_i}^+$* In this subsection, we mainly focus on the importance of new data that add to the integration system by integrating a web database.

Definition 2 ($D_{s_i}^+$): Given a candidate deep web database s_i and the status of integration system D , the $D_{s_i}^+$ is expressed by correlation of the degree of these new data with greater importance query.

So we generate a set of queries with weight to estimate importance of new data. A query workload QW is a set of pairs of the form $\{q, w\}$, where q is a query and w is a weight attributed to the query denoting its relative importance. Typically, the weight assigned to a query is proportional to its frequency in the workload, but it can also be proportional to other measures of importance, such as the monetary value associated with answering it, or in relation to a particular set of queries for which the system is to be optimized[8].

In this paper, we use a query generator to generate a set of queries. Each generated query refers to a single term and is representative of the set of queries that refer that term. For simplicity, the generator only produces keyword queries. The generator assigns a weight w to each query using a distribution to represent the frequency of queries on this term. Since the distribution of query-term frequencies on web search engines typically follows a long-tailed distribution [9], for w in our experiments we use values selected from a Pareto distribution [10].

In what follows, we show how to estimate approximate $D_{s_i}^+$. The approximate $D_{s_i}^+$ is defined as the weighted sum of the volume of new data for each the queries in the

workload QW .

$$D_{s_i}^+ = size(s_i) * \sum_{(q_j, w_j) \in QW} w_j * \left(\frac{|q_j(D) \cup q_j(s_i)| - |q_j(D)|}{|q_j(s_i)|} \right) \quad (2)$$

In order to actually compute the utility of a web database as defined in Equation 1. we Standardize $D_{s_i}^+$ one which has the range, 0–1.

B. Negative Utility

There are networking and processing costs associated with integrating a web database in the integration system. These are the costs to retrieve data from the database while executing queries, map this data to the global mediated schema and so on. Those cost aspects are the negative utility of web database and they may be just as important to users.

In this paper, we mainly consider time-cost as negative utility. Time-cost is expressed by response time that the time starts from user sending a query to the web database or integration system and ends at time they return the final result set of this query.

Response time contains time-cost to retrieve data from the database while executing queries, map this data to the global mediated schema, and resolve any inconsistencies with data retrieved from all sources and so on.

In what follows, we use a random query workload Q and a query with weighted workload QW that are used in above subsection to estimate approximate $D_{s_i}^-$. The approximate $D_{s_i}^-$ can be expressed by the following equation.

$$D_{s_i}^- = C_{s_i}^Q w_1 + C_{s_i}^{QW} w_2 \quad (3)$$

Where $0 \leq \{w_1, w_2\} \leq 1$ and $w_1 + w_2 = 1$. $C_{s_i}^Q$ is the increased average response time of a random query q in Q over after D integrating s_i , $C_{s_i}^{QW}$ is the increased average response time of a query q in QW over after D integrating s_i .

$C_{s_i}^Q$ can be expressed by the following equation.

$$C_{s_i}^Q = \frac{\sum_{j=1}^{|Q|} (q_j^{time}(D \cup s_i) - q_j^{time}(D))}{|Q|} \quad (4)$$

$C_{s_i}^{QW}$ can be expressed by the following equation.

$$C_{s_i}^{QW} = \frac{\sum_{j=1}^{|QW|} (q_j^{time}(D \cup s_i) - q_j^{time}(D)) * w_j}{|QW|} \quad (5)$$

where $q_j^{time}(D)$ is response time of q_j over D , $q_j^{time}(D \cup s_i)$ is response time of q_j over after D integrating s_i .

Based on the next section, in order to compute $D_{s_i}^-$, we need to execute queries on D m times, this is a time-cost

process and it is difficult to measure $q_j^{time}(D \cup s_i)$. So we simplify $C_{s_i}^Q$ and $C_{s_i}^{QW}$ respectively.

$C_{s_i}^Q$ can be simplified by average response time of a random query q in Q over s_i .

$$C_{s_i}^Q = \frac{\sum_{j=1}^{|Q|} (q_j^{time}(s_i))}{|Q|} \quad (6)$$

$C_{s_i}^{QW}$ can be simplified by average response time of a query q in QW over s_i .

$$C_{s_i}^{QW} = \frac{\sum_{j=1}^{|QW|} (q_j^{time}(s_i)) * w_j}{|QW|} \quad (7)$$

In order to actually compute the utility of a web database as defined in Equation 1. we Standardize $D_{s_i}^-$. one which has the range, 0–1.

III. RESOURCE SELECTION AND INTEGRATION USING THE UTILITY MAXIMIZATION MODEL

In this section, we describe how to use the utility maximization model, which optimizes the resource selection problems for deep web data integration. The goal of the resource selection algorithm is to build an integration system contains m web databases(e.g.,20 databases) that contains as high utility as possible, which can be formally defined as an optimization problem:

Given a candidate source set: $S = \{s_1, s_2, \dots, s_n\}$, the status of integration system D , find

$$\arg \max_{s_i \in S} (Utility(D, s_i)) \quad (8)$$

The database selection decision is made based on the approximate utility of the web database.

Our approach is to select and integrate web databases in an iterative manner, where web databases are integrated incrementally. We select a maximal utility web database s_i to integrate from S each time. This approach takes advantage of the fact that some web databases provide more utility to the status of integration system than others: they are involved in more queries with greater importance. Similarly, some data sources may never be of interest, and therefore spending any effort on them is unnecessary.

The selection and integration algorithm using the utility maximization model as follow:

.....
Algorithm to web database selection and integration:

Integration Algorithm($D = \phi$; S : Set of candidates web databases; m is the maximum number of sources that the user is willing to select($m \leq |S|$)

Count=0;

while (Count $\leq m$) **do**

$s = \arg \max_{s_i \in S} (Utility(D, s_i))$;//select a maximal

utility of s_i form S

$D = \text{integrate}(D, s)$; ;//integrate(D, s) is integrate s into D , the status of integration system D is updated

$S = S - s$; ;//Set of candidates web databases S is updated

Count++;

end while

return D ;

.....

Integration algorithm call selection algorithm for selecting a most benefit web database to integrate each time. In initialization status $D = \phi$, while a web database is integrated, the status of integration system and the set of candidate web databases will change, at the same time, $Utility(D, s_i)$ will also change for each web database in the set of candidate web databases. So when selecting next web database to integrate, Selection Algorithm recomputes any web databases whose benefit value may have changed. Selection algorithm then returns the most benefit web databases for user integration. Finally, if the number of integrated web database equals to threshold m , it has finished; if not, it continues.

IV. EXPERIMENT EVALUATION

In this section we present a detailed experimental evaluation on real-world datasets of the approach presented in the previous section.

A. Experimental Setup

Candidate web databases. We evaluate our approach using real data sets from movie domain in the web. we get 80 web databases that we can obtain all data from back-end as a set of candidate web databases for integration.

Queries workload. We use four queries workload in the experiment. two random queries workload($RQ1$ and $RQ2$) and two queries with weight workload($WQ1$ and $WQ2$). We use a query generator to randomly generate 500 keywords as $RQ1$ and 300 keywords random queries as $RQ2$. We also generate a 500 keywords queries with weight as $WQ1$ and 300 keywords with weight queries as $WQ2$ by using the method in the 2.2 subsection. $RQ1$ and $WQ1$ is used to estimate the utility of web database and $RQ2$ and $WQ2$ is used on experiment evaluation.

Weight. Based on user's interest, all the weight can be set by user. In this paper, In equation 1, the default weights of $D_{s_i}^+$ and $D_{s_i}^-$ are 0.7 and 0.3, respectively. In equation 3, the default weights of $C_{s_i}^Q$ and $C_{s_i}^{QW}$ are 0.4 and 0.6, respectively.

In order to validate the effectiveness of our approach, we compare our approach with quality-based[7]. In this paper, the quality of web database is measured only depending on objective criteria in [7]. Each strategy selects m web databases to build integration system. Benefit-based: m web databases are selected and integrated with our approach.

B. The importance of data in the integration system

We now turn our attention to evaluating the importance of data in the integration system. In this paper, the importance of data in the integration system is defined as the following equation.

$$importance = \frac{\sum_{(q_j, w_j) \in QW} w_j * q_j(D)}{|QW|} \quad (9)$$

For this experiment, we compare the importance of data in the integration system that are produced by our approach and quality-based approach. The results are shown in Figure.3. Here we can see the more importance of data in integration system that is produced by our approach than quality-based approach.

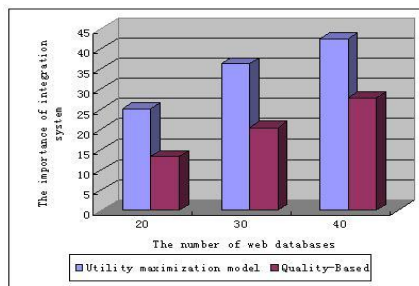


Figure 1. The importance of data in the integration system

C. Time Effective

Our final set of experiment studies the time-cost of a query workload over integration system that are produced by our approach and quality-based approach. Figure.4 shows the average response time for a query in RQ2 over integration systems that choose 20,30,40 web databases to integrate from a universe of 80 candidate web databases. It is obvious that response time of our approach is low, and time-cost is slow growth with the increase in the number of database.

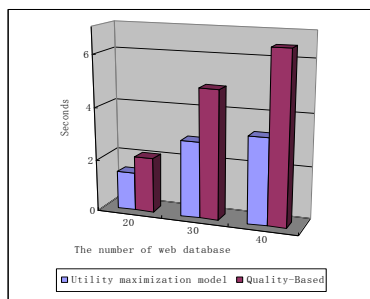


Figure 2. The average response time for a query in RQ1

V. CONCLUSION

We have presented a utility maximization model to the problem of the deep web database selection. Utility maximization model estimate the utility of web database bringing to a given status of integration system. The utility of web database is estimated by two aspects: positive and negative utility. In this paper, the importance of new data that add to integration system by integrating

a web database are considered as positive utility; Negative utility of web database is measured by the average response time(time-cost) of a query on workload. With the estimated utility information, the paper selects and integrates web databases in an iterative manner, where web databases are integrated incrementally. We select a maximal utility web database to integrate from set of candidate web database each time, it obtains the maximal utility of integration system with as few web databases as possibly effective. Finally, we described a set of experiments on real datasets that validated the benefits of our approach.

ACKNOWLEDGMENT

This research was partially supported by The Natural Science Foundation of China under grant No. 60970015; The 2008 Jiangsu Key Project of Science Support and Self-Innovation under grant No.BE2008044; The Natural Science Foundation of Jiangsu under grant No.BK2009563. The Opening Project of Jiangsu Province Software Engineering R&D Center for Modern Information Technology Application in Enterprise under grant No.SX200904; The Scientific Research Foundation of Suzhou Vocational University under grant No.SZD08L24.

REFERENCES

- [1] B. Michael K. "The Deep Web: Surfacing Hidden Value." *The Journal of Electronic Publishing from the University of Michigan*, July,2001.
- [2] Chang KCC, He B, Li CK, Patel M, Zhang Z. "Structured Databases on the Web: Observations and Implications." *SIGMOD Record*, vol. 33. no. 5, pp.61-70, 2004.
- [3] D'Souza, J. Zobel, and J. Thom. "Is CORI Effective for Collection Selection an Exploration of parameters, queries, and data." *In Proceedings of Australian Document Computing Symposium*, pp.41-46, Melbourne, Australia,2004.
- [4] K. C.-C. Chang, B. He, and Z. Zhang. "Toward large scale integration: Building a MetaQuerier over databases on the Web." *In Proceedings of Biennial Conference on Innovative Data Systems Research*, pp.44-55, ACM Press, Asilomar,CA,2005.
- [5] Shokouhi,M., "Central-Rank-Based Collection Selection in Uncooperative Distributed information Retrieval." *In Proceedings of the 29rd European Conference on Infomation Retrieval*, pp.160-172, Rome, Italy,2007.
- [6] F.Naumann, J. C. Freytag, and M. Spiliopoulou. "Quality-driven Source Selection Using Data Envelopment Analysis." *In Proceedings of the 3rd Conference on Information Quality*, pp. 137-152, Cambridge, MA,1998.
- [7] Ashraf Abounnaga and Kareem El Gebaly. "µBE: User Guided Source Selection and Schema Mediation for Internet Scale Data Integration." *In Proceedings of the IEEE International Conference on Data Engineering*, pp.186-195, IEEE Press, Turkey ,2007.
- [8] Shawn R. Jeffery, Michael J. Franklin, Alon Y. Halevy: "Pay-as-you-go user feedback for dataspace systems." *SIGMOD Conference*: 847-860,2008.
- [9] Craig Silverstein, Monika Henzinger, Hannes Marais, and Michael Moricz. "Analysis of a very large altavista query log. Technical Report" 1998-014, *Digital Systems Research Center*,1998.
- [10] George Casella and Roger Berger. *Statistical Inference, Second Edition*, Duxbury,2002.