Utility Maximization Model for Deep Web Source Selection and Integration

Xuefeng Xian[1,2], Zhiming Cui[1,2*], Pengpeng Zhao[1,2], Yuanfeng Yang[1,2] and Guangming Zhang [2]
[1]JiangSu Province Support Software Engineering R&D Center for Modern Information Technology Application in Enterprise, Suzhou, China
[2]The Institute of Intelligent Information Processing and Application, Soochow University, Suzhou, China
Email: xianxuefeng@jssvc.edu.cn; szzmcui@suda.edu.cn

***Abstract*— The World Wide Web is witnessing an increase in the amount of structured content--vast collection of structured data are on the rise due to the deep web. Such Internet-scale deep web data integration tasks are becoming increasingly more common. In Internet-scale deep web data integration tasks, a primary challenge is to determine in which web database to be included in the integration system. This paper presents a utility maximization model for resources selection of deep web data integration. This new model shows an efficient and effective way to estimate the approximate utility of the web database bringing to a given status of an integration system by integrating it. The utility of the web databases is synthesized by positive and negative utility. With the estimated utility information, web database selection can be made by explicitly optimizing the goal of high-utility(include as much and important data as possible in the selected databases, and the query cost of which as low as possible) in an iterative manner, where web databases are integrated incrementally. We experimentally demonstrate that our approach is efficient and finding high-utility data integration solutions.**

***Index Terms*—deep web; data integration; utility maximization model; web database selection**

I. INTRODUCTION

An ever increasing amount of information on web is available through search interfaces. This information is often called the hidden web or deep web[1] because the search engine crawlers rely on hyperlinks to discover new contents, there are very few links that point to hidden web pages and crawlers have poor ability to fill out arbitrary html forms. Since the majority of web users rely on traditional search engines to discover and access information on the web, the deep web is practically inaccessible to most users and hidden from them. Even if users are aware of a certain part of the deep web, they have to go through the painful process of issuing queries to all potentially relevant deep web databases and investigating the results manually. On the other hand, 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 order to assist users in accessing the information 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[3,4,5,6,7]. 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[6,8,9,10,11,12,13]. For example, Cori[8] applies inference networks for collection selection. It has been reported as the most effective method in many papers, but there are question marks over its effectiveness[6]. Redde[9] ranks the collections based on the estimated number of relevant documents they contain. Redde has been shown to be very effective on some testbeds. Si and Callan[10] presented their Unified Utility Maximization (UUM) framework for collection selection. UUM performs slightly better than Redde on some testbeds. These works are mainly on text databases, not the structured databases.

In this paper, we mainly focus on the deep web data source selection problem for building the deep web data integration system. Ideally, to provide comprehensive query results in the integration system, the system should ask user to integrate most or even all web databases in a particular domain. In Internet-scale data integration tasks, however, this approach is not feasible. The main reason is that deep web is so enormous in scope that there may be hundreds or thousands of web databases providing data of relevance to a particular domain. Furthermore, the user may not want to include all available web databases in the integration system being defined, especially if there is significant overlap in the data in different web databases. Moreover, there are networking and processing costs

doi:10.4304/jcp.5.7.995-1002

associated with including a web database in the integration system. These are the costs to retrieve data from the database while executing queries, to map this data to the global mediated schema and so on. The more sources we have, the higher these costs are. So an integration system cannot possibly involve in all of them, The problem of web database selection has been a primary challenge to Internet-scale deep web data integration.

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[14]. The solution is computationally expensive so it does not apply to Internet-scale data integration. In [15], 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. Moreover, these strategies are to select top-m web databases by once for building integration system, the overlap between the data in the top-m web databases is not to be considered.

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, and avoiding the significant overlap in the data in integration system effectively and reducing the cost of query.

We describe a detailed experimental evaluation on real deep web databases shows that the selected and integrated result of web databases produced by our approach yields an integration system with more utility than other strategies.

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 $S=\{s_1,s_2,...,s_n\}$. 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 $0\le\{w_1,w_2\}\le 1$ 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 volume of new data that add to the integration system by integrating s_i, denoted by $D_{s_i}^{1+}$ and the importance of new data, denoted by $D_{s_i}^{2+}$. 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 and they are involved in more queries with greater importance, s_i bring more positive utility to a given status of integration system D.

Thus, $D_{s_i}^{+}$ is defined as:

$$D_{s_i}^{+}=D_{s_i}^{1+}w_1+D_{s_i}^{2+}w_2 \quad (2)$$

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

1) Estimating $D_{s_i}^{1+}$ In this paper, $D_{s_i}^{1+}$ can be defined as follows.

Definition 1 ($D_{s_i}^{1+}$): Given a candidate deep web database s_i and the status of integration system D, the $D_{s_i}^{1+}$ is expressed by the volume of new data that add to the integration system by integrating s_i. Simply speaking, $D_{s_i}^{1+}$ is the number of data that contained in s_i, but not in D. The $D_{s_i}^{1+}$ can be expressed by the following equation.

$$D_{s_i}^{1+}=|D\cup s_i|-|D| \quad (3)$$

Where $|D|$ is the volume of data of unions of web databases in D, not counting duplicate data in D. $|D\cup s_i|$ is the volume of data after D integrates s_i.

Broadly speaking, $D_{s_i}^{1+}$ can be measured by analyzing all the data in D and s_i. The analysis of all data makes a solution that requires fetching all the data from web databases prohibitively expensive. Hence, in next subsection we show how we approximate $D_{s_i}^{1+}$. Our experimental evaluation shows that despite our approximations, our approach is to select and integrate web databases effectively.

As discussed above, we can not possibly analyze all the data in D and s_i. So we estimate approximate $D_{s_i}^{1+}$ by analyzing partial data which are obtained by sampling small amount of data from D and s_i randomly with query-based sampling.

Queries and Workloads: Queries are the primary mechanism for retrieving information from web database. Given a query q, when querying web database s_i, We denote the result set of q over s_i by $q(s_i)$. In this paper, a query workload Q is a set of random queries: $Q=\{q_1,q_2,...,q_m\}$. As the result set is retrieved by random queries, query-based results indicate the objective content of the web database.

To estimate approximate $D_{s_i}^{1+}$, we analyze the result set of the query workload Q over s_i and D representing all data in s_i and D.

In what follows, we show how to estimate approximate $D_{s_i}^{1+}$. The approximate $D_{s_i}^{1+}$ can be expressed by the following equation.

$$D_{s_i}^{1+}=\frac{|Q(D)\cup Q(s_i)|-|Q(D)|}{|Q(s_i)|}*size(s_i) \qquad (4)$$

Where $size(s_i)$ is the amount of data in s_i, $|Q(s_i)|$ and $|Q(D)|$ is separately the size of the result set of the query workload Q over s_i and D.

In this paper, $Q(s_i)$ is defined as the union of the result set for the queries in the workload Q on s_i:

$$Q(s_i)=\bigcup_{i=1}^{|Q|}(q_i(s_i)) \qquad (5)$$

With $Q(s_i)$ similar, $Q(D)$ is the union of the result set for the queries in the workload Q on the integration system D. Different from query on single web database, when querying the integration system D, the query processor utilizes all the integrated web databases. Merging result from all the integrated web databases into result set, eliminating all duplication of data at the same time, we denote result set by $Q(D)$. The high cost work to obtain $Q(D)$, in the next section, we will introduce an efficiency approach to obtain $Q(D)$.

Centralized sample database and Duplicate detection: Web databases, as we know, are heterogeneous in the web. In this paper, in order to obtain $Q(D)$ and $Q(D)\cup Q(s_i)$, we build a centralized sample database with consolidated single mediated schema that is set by the domain expert. We mapped the result set for the queries in the workload Q on each s_i in S to centralized sample database. $Q(D)$ and $Q(D)\cup Q(s_i)$ can easily be obtained, and duplication of data can also be detected by using a probabilistic approache[16] in centralized sample database. A probabilistic approache is proposed for solving the duplicate detection problem in [16], it can be used to match records with multiple fields in the database.

Estimate size of database: Based on equation 3, to compute approximate $D_{s_i}^{1+}$, we need to be able to compute the amount of data in web database s_i. The difficulty is computing the amount of data in web database, because (1) many sources do not allow unrestricted access to their data, and (2) even if the sources did allow access to the data, the sheer amount of data at the sources makes a solution that requires fetching all the data from the sources prohibitively expensive[15]. Thus, we need a way to estimate the amount of database in web database with a few accessing the data. Ling et al [17] propose based on the word frequency an approach to assess the size of web database. In this paper, we could use it to assess the size of web database. For instance, for s_i, $size(s_i)$ refers to the size of web database s_i.

2) Estimating $D_{s_i}^{2+}$ 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 ($D_{s_i}^{2+}$): Given a candidate deep web database s_i and the status of integration system D, the $D_{s_i}^{2+}$ 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[18].

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 [19], for w in our experiments we use values selected from a Pareto distribution [20].

In what follows, we show how to estimate approximate $D_{s_i}^{2+}$. The approximate $D_{s_i}^{2+}$ is defined as the weighted sum of the volume of new data for each the queries in the workload QW.

$$D_{s_i}^{2+} = size(s_i) * \sum_{(q_j, w_j) \in QW} w_j * (\frac{|q_j(D) \cup q_j(s_i)| - |q_j(D)|}{|q_j(s_i)|}) \quad (6)$$

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.

$$D_{s_i}^{-} = C_{s_i}^{Q} w_1 + C_{s_i}^{QW} w_2 \quad (7)$$

Where $0 \le \{w_1, w_2\} \le 1$, 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.

$$C_{s_i}^{Q} = \frac{\sum_{j=1}^{|Q|} (q_j^{time}(D \cup s_i) - q_j^{time}(D))}{|Q|} \quad (8)$$

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

$$C_{s_i}^{Qw} = \frac{\sum_{j=1}^{|Qw|} (q_j^{time}(D \cup s_i) - q_j^{time}(D)) * w_j}{|QW|} \quad (9)$$

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.

$$C_{s_i}^{Q} = \frac{\sum_{j=1}^{|Q|} (q_j^{time}(s_i))}{|Q|} \quad (10)$$

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

$$C_{s_i}^{QW} = \frac{\sum_{j=1}^{|QW|} (q_j^{time}(s_i)) * w_j}{|QW|} \quad (11)$$

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: $S = \{s_1, s_2, ..., s_n\}$, the status of integration system D, find

$$\arg\max_{s_i \in S} (Utility(D, s_i)) \quad (12)$$

In order to actually compute the utility of a web database as defined in Equation 1,2. we Standardize $D_{s_i}^{1+}$, $D_{s_i}^{2+}$ and $D_{s_i}^{-}$. one which have the range, 0–1.

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 or are associated with more data. 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($m \le |S|$)

Count=0;
while (Count $\le m$) **do**
$s = \arg\max_{s_i \in S}(Utility(D, s_i))$;//select a maximal utility of s_i form S
D=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.

Based on integration algorithm and selection algorithm, Selection m web databases from S to integrate, The equation 4 need to be called $\frac{1}{2}m(2|S|+m-1)$ times. $Q(D)$, $Q(s_i)$ and $size(s_i)$ are called m times repeatedly. We can see that $Q(s_i)$ and $size(s_i)$ are constant in m times calls, so they only need to be computed one time. In this paper, in initialization status, before web database selection, we create $Q(s_i)$, $|Q(s_i)|$ and $size(s_i)$ for each s_i in S , and the system stores them in lists. In equation 4, $Q(D)$ is changed with a new web database integrated into D, in order to obtain $Q(D)$ and $|Q(D)|$, we need to repeat executing query workload Q over D. The high cost of retrieving data from integration system while executing queries. In what follows, we show how to obtain $Q(D)$ and $|Q(D)|$, but need not repeat executing query workload Q over D . We assume integration system has integrates k web databases, denoted D_k . $Q(D_k)$ can be expressed by the following recursive formula.

$$Q(D_k) = Q(D_{k-1}) \cup Q(s_k) \quad (13)$$

Where D_{k-1} is integration system with k-1 web databases, s_k is the first k-web database that is integrated into system.

So $Q(D_k)$ can also be expressed by the following equation.

$$Q(D_i) = \bigcup_{j=1}^{|k|}(Q(s_j)) \quad (14)$$

Where s_j is the first j-web database that is integrated into system.

Through the equation 14, we are able to effectively obtain $Q(D)$ and $|Q(D)|$ avoiding the cost of executing query workload Q over D.

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 2, the default weights of $D_{s_i}^{1+}$ and $D_{s_i}^{2+}$ are 0.5 and 0.5, respectively. In equation 7, 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[16]. In this paper, the quality of web database is measured only depending on objective criteria in [16]. Each strategy selects m web databases to build integration system.Benefit-based: m web databases are selected and integrated with our approach.

B. The volume of data and the overlap degree of between data in the integration system

In this subsection, we study the volume of data and degree of overlap integration systems, which are separately obtained by our approach and quality-based[16]. Figure.1 shows the percentage which the

volume of data to choose 10 to 100 web databases to integrate from a universe of 100 candidate web databases. Here the volume of data does not count duplicate data in all web databases. First observe the curve for our approach. This approach selects the maximal utility candidate web databases to integrate, the curve is very steep for the early integration. As it integrates more web databases, the curve is flattens out as these web databases bring less new data to the integration system. Finally, it converges to 1, all candidate web databases are integrated into system. the slopes of the curves for quality-based strategy is much shallower. It takes more web databases to produce an integration system with a high percentage. The percentage of the volume of data has reached 94.5% when system integrates 30 web databases by using our approach. One of our goal is to obtain maximum data in integration system with as few web databases as possible, so our approach is efficient.

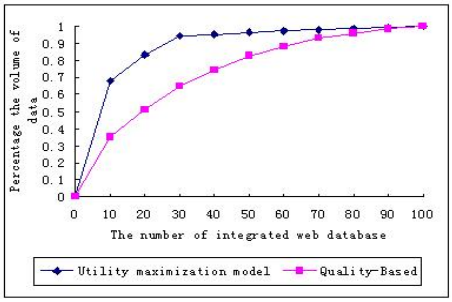

Figure 1. .Percentage the volume of data

The degree of overlap between data in integration system is
expressed by the following equation.

$$Overlap = \frac{\sum_{s_i \in D} |s_i|}{|D|} \quad (15)$$

Where $|D|$ is the volume of data of unions of web databases in D, not counting duplicate data in D. $|s_i|$ is the volume of data s_i.

To simplify the calculation of equation 15, in our experiment, equation 15 can be simplified by the following equation.

$$Overlap = \frac{\sum_{s_i \in D} |Q(s_i)|}{|Q(D)|} \quad (16)$$

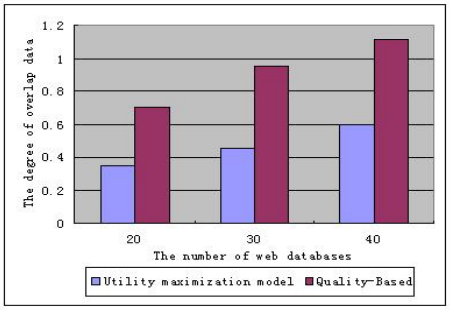

Figure 2. The degree of overlap data in the integration systems

Figure.2 shows the degree of overlap data to respectively choose 20,30,40 web databases to integrate from a universe of 100 candidate web databases. Our approach performs better than which is quality-based. The degree of overlap data by our approach is lower than that of quality-based approach. Our approach selects and integrates web databases in an iterative manner, where web databases are integrated incrementally. it is avoids the significant overlap of between data in integration system effectively.

C. *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.

$$importance = \frac{\sum_{(q_j, w_j) \in QW} w_j * q_j(D)}{|QW|} \quad (17)$$

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.

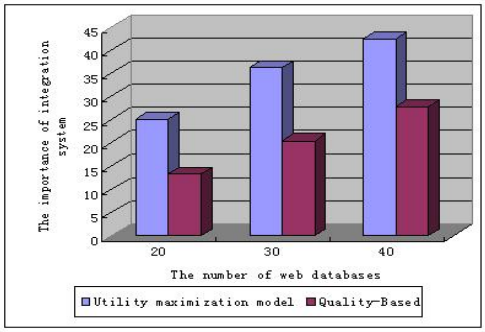

Figure 3. The importance of data in the integration system

D. *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. The degree of overlap between data in integration system, which is produced by quality-based, is very high. So the time-cost is high for retrieving a large number of duplicate data from the source while executing queries, and result merging also takes more time.

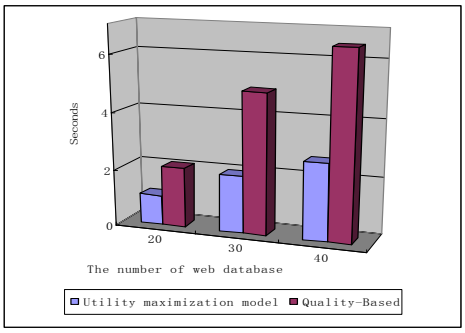

Figure 4. The average response time for a query in RQ1

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 volume and 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 for the Young Teachers, Suzhou Vocational University under grant No.SZDQ09L08.

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] C.Yu, W.Meng, W.Wu, K.Liu. “Efficient and Effective Metasearc for Text Databases Incorporating Linkages among Documents.”*In Proceedings of SIGMOD*, pp.187-198, ACM Press, CA,2001.

[4] P.G. Ipeirotis, L. Gravano. “Distributed Search over the Hidden Web: Hierarchical Database Sampling and Selection.”*In Proceedings of VLDB*, pp.394-405, Hong Kong ,2002.

[5] W. Wu, C. T. Yu, A. Doan, and W. Meng. “An interactive clustering-based approach to integrating source query interfaces on the Deep Web.” *In Proceedings of ACM SIGMOD International Conference on Management of Data*, pp.95-106, ACM Press, Paris ,2004.

[6] 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.

[7] 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.

[8] Callan, Z. Lu, and W. B. Croft, “Searching Distributed Collections with Inference Networks,” *Proceedings of the ACM SIGIR Conference(SIGIR 1995)*, pp.21-28, July ,1995.

[9] L. Si and J. Callan. “Relevant Document Distribution Estimation Method for Resource Selection,” *Proceedings of ACM SIGIR Conference(SIGIR2003)*, pp.298-305, Aug.,2003.

[10] Luo Si,J.P.C., “Unified Utility Maximization Framework for Resource Selection,” *In Proceedings of ACM CIKM Conference*, pp.32-41, ACM Press, Washington,2004.

[11] Shokouhi,M., “Central-Rank-Based Collection Selection in Uncooperative Distributed information Retrieval.” *In Proceedings of the 29rd European Conference on Infromation Retrieval*, pp.160-172, Rome, Italy,2007.

[12] Zhenyu Liu, Cl.,Junghoo Cho, Weslesy W. Chu, “A Probabilistic Approach to Metasearching with Adaptive Probing.” *In Proceedings of the international Conference on Data Engineering*, pp.547-559, IEEE Press, Boston, USA,2004.

[13] P.lpeirotis, L.G., “When one Sample is not Enough: Improving Text Database Selection Using Shrinkage.” *In Proceedings of the ACM SIGMOD International Conference On Management of Data*, pp.767-778, ACM Press, Paris, France,2004.

[14] 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.

[15] Ashraf Aboulnaga 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.

[16] Ahmed K. Elmagarmid, et al., “Duplicate Record Detection: A Survey.” *IEEE Transactions on. Knowledge and Data Engineering*, Vol. 19, No. 1, pp. 1-16, 2007.

[17] Ling Yan-Yan, Meng Xiao-Feng, Liu Wei. “An Attributes Correlation Based Approach for Estimating Size of Web Databases.” *Journal of Software*, 19(2), 224-236,2008.

[18] Shawn R. Jeffery, Michael J. Franklin, Alon Y. Halevy: "Pay-as-you-go user feedback for dataspace systems." *SIGMOD Conference*: 847-860, 2008.

[19] 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.

[20] George Casella and Roger Berger. *Statistical Inference, Second Edition*, Duxbury, 2002.

Xue-Feng Xian received his Master degree in computer science from Hohai University, Nanjing, China, in 2006. Currently, he is working on doctoral degree at Soochow University. His research interests include information retrieval and integration, data minning.

He is currently a Lecturer in the department of Computer Engineering at Suzhou Vocational University.

Mr.Xian is currently a member of the China Computer Federation.

Zhi-Ming Cui received his Bachelor degree in computer science from National University of Defense Techonlogy, Changsha, China, in 1983. His research interests include intelligent information processing, computer network applications and database applications.

He is currently a Professor and Doctoral Advisor in the College of Computer Science and Technology at Soochow University. He is also the director of JiangSu Province Support Software Engineering R&D Center for Modern Information Technology Application in Enterprise. He has served as Vice-Director of a number of departments, including the Suzhou Computer Federation, Suzhou Association for Science and Technology and Jiangsu Key Laboratory of Computer Information Processing Technology.

Prof.Cui is currently a advanced member of the China Computer Federation.

Peng-Peng Zhao received his Ph.D degree in computer science from Soochow University, Suzhou, China, in 2008. His research interests include deep web, data minning and machine learning.

He is currently a Lecturer in the College of Computer Science and technology at Soochow University.

Dr.Zhao is currently a member of the China Computer Federation. He is also a member of the IEEE Computer Science Society. He is also a member of the ACM.

Yuan-Feng Yang received his Master degree in computer science from Soochow University, Suzhou, China, in 2006. Currently, she is working on doctoral degree at Soochow University.Her research interests include data management, data minning.

He is currently a Lecturer in the department of Computer Engineering at Suzhou Vocational University.

Guang-Ming Zhang received his Master degree in computer science from Fudan University, Shanghai, China, in 2006. Currently, she is working on doctoral degree at Soochow University. Her research interests include image analysis and processing, data minning.