# Efficient Selection and Integration of Hidden Web Database

Xuefeng Xian[1,2], Pengpeng Zhao[1,2], Yuanfeng Yang[1,2], Jie Xin[2] and Zhiming Cui[1,2*]
[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*—An ever increasing amount of valuable information is stored in web databases, "hidden" behind search interfaces. A new application area emerge for information retrieval and integration. There may be hundreds or thousands of web databases providing data of relevance to a particular domain on the web. So a primary challenge to internet-scale hidden web database integration is to determine in which web databases to include in the integration system with the aim of making the system contain as much high-quality data as possible and the least degree of overlap. In this paper, we present an approach to iteratively select and integrate candidate web database. The core of this approach is a benefit function that evaluates how much benefit the web database brings to a given status of an integration system by integrating it. We devise a benefit function based on the volume and quality of those new data that added to integration system by integrating the web database. We show in practice how to efficiently apply our approach to select and integrate web database. Our experiments on real hidden web databases indicate that the selected and integrated result of web databases produced by our approach yields an integration system with a significant higher utilities than a wide range of other strategies.**

***Index Terms*—hidden web, data integration, web database selection.**

## I. INTRODUCTION

An ever increasing amount of information on web is available through search interfaces, as Figure 1 shows. 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 do not have the 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 hidden web is practically inaccessible to most users and "hidden" from them. Even if users are aware of a certain part of the hidden web, they have to go through the painful process of issuing queries to all potentially relevant hidden web database and investigating the results manually. On the other hand, the hidden 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 hidden web sites and 450,000 query interfaces available at that time, and the two figures are still increasing rapidly.

In order to assist users accessing the information in the hidden web, recent efforts have focused on building hidden web data integration system. Ideally, to provide comprehensive query results in the integration system, the system should ask user to integrate and query most or even all web database in a particular domain. This approach, however, is not feasible given the scale and nature of internet-scale data integration. The main reason is that hidden web is so enormous in scope that there may be hundreds or thousands of web databases providing data of relevance to a particular domain. The user may not want to include all available web databases in the integration system being defined and also may not want to query all web database in the system to a user's query, especially if there is significant overlap in the data in the different web databases and a lot of the low quality of the web databases. Moreover, there are networking and processing costs associated with including 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. The more sources we have, the higher these costs. So a integration system cannot possibly involve in all of them, The problem of web database selection has been a primary challenge to internet-scale hidden web data integration.

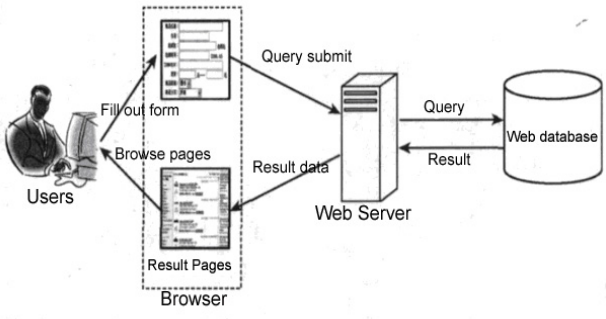


Figure 1. The process of accessing data from hidden web

In the internet-scale hidden web data integration, the problem of web database selection emerge in two-phases. First-phase, before building an integration system, the m web databases must be automatically selected to integrate from hundreds or thousands of web databases relevance to a particular domain. m is the maximum number of web databases that the user is willing to select. Second-phase, after building an integration system, given a query, a set of the most relevant web databases must be selected to do the search. In this paper, we study the problem of automating the selection of web databases to integrate in first-phase.


doi:10.4304/jcp.5.4.500-507

In this paper, our goal is to select and integrate m web databases that contain as much high-quality data as possible and the least degree of overlap between the data in the integration system. We begin by presenting an approach for iteratively selecting and integrating hidden web database. The approach selects a most benefit web database from a set of candidate web databases to integrate each time. After when each web database is integrated, we update the status of integration system and recompute the next most benefit web database to integrate. The core of this approach is a benefit function that evaluates how much benefit the web database bring to a given state of a integration system by integrating it. Thus, we devise a benefit function for web database based on the volume and quality of new data that added to the integration system by integrating the web database.

We describe a detailed experimental evaluation on real hidden web databases which shows that the selected and integrated result of web databases produced by our approach yields a integration system with the more high-quality data and lower degree of overlap between the data than a variety of other strategies.

The remaining of the paper is organized as follows. In section 2, first we discuss the related work. Section 3 discusses our benefit function for evaluating the benefit of web databases. Section 4 describes the algorithms of web database selection and integration. Section 5 presents a detailed evaluation of our web database selecting and integrating strategy. We conclude in section 6.

## II. Related Work

There have been considerable researchs on the problem of web database selection in both the two-phases. We survey the most related work in this section.

Recent mainly efforts have been focused on the second-phase that automatically selects the most relevant databases to a user's query[3,4,5,6,7,8,9]. For example, Cori[3] 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[4]. Redde[5] 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[6] 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 database.

In traditional small-scale data integration tasks, domain expertise determines in which web databases that should be included in the integration system. So there is a little work on the first-phase. In [10], the problem of source selection is modeled as an optimization problem and solved by using the data envelopment analysis technique. The solution is computationally expensive so that it does not apply to internet-scale data integration. In [11], data source is selected by the user depending on several subjective and objective criteria. Because it depend on some subjective preferences of the user, it does not apply to automatic web database selection. Moreover, these strategies are to select top-m web databases in a time for building data integration system, the overlap between the data in the top-m web databases is not to be considered, a high degree of overlap is bad because we unnecessarily get the same data from many sources.

Our approach has two major differences compared to [10] and [11].

1.Our approach select web databases based on the volume and quality of those new data that added to the integration system by integrating the web database, and it does not rely on subjective criteria, so it can select web database for hidden web data integration automatically, not requiring any human intervention.

2.Our approach select and integrate web databases in a iterative manner, where web databases are integrated incrementally. The significant overlap is avoided by using benefit function and iterative integration manner in the integration system.

## III. Benefit Evaluation

Suppose we are given an integration system $D$ and a set of candidate web databases $S = \{s_1, s_2, ..., s_n\}$ in a particular domain. Benefit function evaluates the benefit of web database $s_i$ bringing to the status of integration system $D$ by integrating $s_i$. In this paper, we referred to as the benefit of web database $s_i$ bringing to the status of integration system $D$ by integrating $s_i$ to the benefit of web database $s_i$. In the following, we describe how to evaluate the benefit of web database.

In this paper, the benefit of web database $s_i$ can be defined as follows.

Definition 1 (The benefit of web database): Given a candidate hidden web database $s_i$ and the status of integration system $D$, the benefit of $s_i$ is expressed by the product of the volume and quality of those new data that added to the integration system by integrating $s_i$, denoted by $Benefit(s_i, D)$.

The $Benefit(s_i, D)$ can be expressed by the following equation.

$$Benefit(s_i, D) = Amount(s_i, D) * Quality(s_i, D) \quad (1)$$

Where $Amount(s_i, D)$ is the volume of those new data that added to the integration system by integrating $s_i$, $Quality(s_i, D)$ is the quality of those new data that added to the integration system by integrating $s_i$

In next two subsections, we show how we measure $Amount(s_i, D)$ and $Quality(s_i, D)$ respectively.

### A. $Amount(s_i, D)$

$Amount(s_i, D)$ is expressed by the volume of new data that added to the integration system by integrating $s_i$, Simply

speaking, $Amount(s_i, D)$ is the amount of data that contains in $s_i$, but not in $D$.

The $Amount(s_i, D)$ can be expressed by the following equation.

$$Amount(s_i, D) = | D \cup s_i | - | D | \quad (2)$$

Where $| D |$ is the amount of data of unions of web databases in $D$, duplicate of data in $D$ is not counted, $| D \cup s_i |$ is the amount of data after $D$ integrate $s_i$. Broadly speaking, $Amount(s_i, D)$ can be measured by analysing all the data at $D$ 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 $Amount(s_i, D)$. Our experimental evaluation shows that despite our approximations, our approach is effectively to select and integrate web databases.

**Approximating** $Amount(s_i, D)$**:** As the above discussed, we cannot possibly analyse all the data in $D$ and $s_i$. So we estimate approximate $Amount(s_i, D)$ by analysing partial data that are obtained by randomly sampling small amount of data from $D$ and $s_i$ with query-based sampling.

**Queries and Workloads:** Queries are the primary mechanism for retrieving information from web database. Given an 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 are retrieved by random queries, query-based results indicated the objective content of the web database.

To estimate approximate $Amount(s_i, D)$, we analyse 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 $Amount(s_i, D)$. The approximate $Amount(s_i, D)$ can be expressed by the following equation.

$$Amount(s_i, D) = \frac{| Q(D) \cup Q(s_i) | - | Q(D) |}{| Q(s_i) |} * size(s_i) \quad (3)$$

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 $s_i$:

$$Q(s_i) = \bigcup_{i=1}^{|Q|} (q_i(s_i)) \quad (4)$$

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 $Q(D)$.

Web databases, as we know, are heterogeneous. 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 easily be detected in centralized sample database.

**Estimate size of database:** Based on equation 3, to compute approximate $Amount(s_i, D)$, 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[11]. Thus, we need a way to estimate the amount of database in web database with a few accessing the data. Ling et al [12] propose an based on the word frequency 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 $s_i$.

### B. $Quality(s_i, D)$

As the above-discussed, the greater of $Amount(s_i, D)$, $s_i$ have the higher priority to integrated, but if there are a large number of the low quality of those new data that added to $D$ by integrating $s_i$. They would reduce the overall quality of the data integration system. The quality of those new data must be considered and they may be just as important to users. So we must exactly estimates the quality of those new data.

In this paper, we measure quality on multiple dimensions that depend on the characteristics of those new data, results of assessment which can be represented in different forms. Numbers are used in quality vectors, representing dimension values for a certain quality criteria (e.g., a completeness of 0.7). Such numbers can be aggregated to single scores, allowing a comparison of the quality of different sources or even a quality ranking. In contrast, assessment categories provide only a few values (e.g., accurate versus not accurate). They are easy to use and to interpret by the user, but are difficult to aggregate in a single score. So we use a number in the range [0,1] representing a measure of the quality of those new data.

To evaluate quality on each of multiple dimension, we define a quality evaluation model which is a quaternion in order to assess the quality of those new data:

$$Qfunction = \{ND, F, W, Qscore\} \quad (5)$$

Where $ND = \{nds_1, nds_2, ..., nds_n\}$, $nds_i$ is a set of those new data that added to $D$ by integrating each $s_i$ in $S$; $F = \{f_1, f_2, ..., f_m\}$ is a set of quality dimensions; The weights, $W = \{w_1, w_2, ..., w_m\}$, are all between 0 and 1, and they sum to 1, weights reflect the relative importance to the different quality dimensions. The weights are set by the user based on their interest to the different quality dimensions; For each quality dimension , $f$. $f(nds_i)$ returns a number in the range [0,1] representing a measure of quality for $nds_i$ on dimension $f$. The higher the value of $f(nds_i)$, the better the quality of $nds_i$ on dimension $f$. $Qscore$ is a quality scores set of all the set of new data in $ND$, $Qscore = \{Qscore_{nds_1}, Qscore_{nds_2}, ..., Qscore_{nds_n}\}$. The quality scores of $nds_i$ is defined as:

$$Qscore_{nds_i} = \sum_{j=1}^{|F|} w_j * f_j(nds_i) \quad (6)$$

Where $Qscore_{s_i}$ returns a number in the range [0,1] representing a measure of aggregate quality for $s_i$ on all quality dimensions. The higher the value of $Qscore_{nds_i}$, the better the quality of $nds_i$.

In the literature, data quality is considered as a multidimensional concept [13],i.e., It is defined on the basis of a set of "dimensions". For example, in [14] data quality dimensions are organized according to data quality categories(such as intrinsic, contextual, accessibility, and representational). Many proposals concerning the set of dimensions characterizing data quality have been made, a survey of which is given in [15]. One of the reasons for such a wide variety is that it is very difficult to define a set of data quality dimensions suitable for every kind of context. In the present paper, we focus only on the self-characteristics of those new data, aiming at capturing the most important and practical dimensions for automatic assessing quality of those new data. However, automatic quality assessment is a difficult task due to some reasons, such as many data quality dimensions are subjective and therefore they can not be automatically assessed (e.g., trust or understandability). In this paper, the choice of core dimensions is guided by those that are objective and take advantage of automatic assessing.

*1) Quality dimension* In the following, we give a definition for these core dimensions.

Completeness: It is defined as the degree to which the elements of an aggregated element are present in the aggregated element instance. A work [16] distinguishes three kinds of completeness, namely: schema completeness, column completeness and population completeness. The measures of such completeness types can give very useful information for a "general" assessment of the data completeness of those new data.

Consistency: It is can also be viewed from a number of perspectives, one being consistency of the same (redundant) data values across tables. Codd's Referential Integrity constraint is an instantiation of this type of consistency. Consistency implies that two or more values do not conflict with each other. Information in the hidden web is likely to be inconsistent as it is provided by multiple information providers, which might use different procedures to capture information, have different levels of knowledge and different views of the world.

Redundancy: it is the measure of the degree of overlap between the data in those new data.

*2) Quality assessment on those dimensions* As the above-discussed, the sheer new data that added to integration system by integrating the web database can not be obtained. In this section we use a part new data instead of all new data to approximately assess the quality scores of new data. The part new data $nds_i$ that added to integration system by integrating $s_i$ is obtained by the following equation.

$$nds_i = Q(D) \cup Q(s_i) - Q(D) \quad (7)$$

In the following, we show how to assess the quality of those new data on these core dimensions.

(1) Completeness

Because we can not obtain sheer schemas and the all data of hidden web database, so we only consider column completeness for those new data, one can define column completeness as a function of the missing values in a column of a table. This measurement corresponds to Codd's column integrity which assesses missing values. it can be measured by taking the ratio of the number of incomplete items to the total number of items and subtracting from one.

The quality score of $s_i$ in the completeness dimension:

$$f_1(nds_i) = 1 - \sum_{j=1}^{N} \frac{c(r_j)}{Total\,\mathrm{Re}\,cord} \quad (8)$$

Where $f_1(nds_1) \in [0,1]$ is the quality score of $s_i$ in the completeness dimension. $c(r_j)$ is the number of incomplete records in $r_j$ field in those new data, $Total\,\mathrm{Re}\,cord$ stands for the amount of those new data and $N$ represents the number of field that contains incomplete records.

(2) Consistency

As the above-discussed dimensions, a metric measuring consistency is the ratio of violations of a specific consistency type to the total number of consistency checks subtracted from one.

The quality score of $s_i$ in the consistency dimension:

$$f_2(nds_i) = 1 - \sum_{j=1}^{N} \frac{Inconsistency(r_j)}{Total\,\mathrm{Re}\,cord} \quad (9)$$

Where $Inconsistency(r_j)$ is the number of violations records in $r_j$ field; $Total\ \mathrm{Re}cord$ stands for the size of dataset and $N$ represents the number of field that contains incomplete records.

(3) Redundancy analysis

Redundancy analysis mainly is the quantization of duplicate records in the those new data.

The quality score of $s_i$ in the redundancy dimension:

$$f_3(nds_i) = 1 - \frac{\mathrm{Re}dundancy(nds_i)}{Total\ \mathrm{Re}cord} \quad (10)$$

Where $Redundancy(D_i)$ is the number of redundancy records and $Total\ \mathrm{Re}cord$ is the size of dataset.

## IV. Selection and Integration of Web Database

Our approach is selecting and integrating web databases in an iterative manner, where web databases are integrated incrementally. The benefit function is estimating the benefit of the web database to the status of integration system. In this section we describe how we make use of benefit function for selection and integration of hidden web database. First, Using equation 1 to calculate the approximate benefit value for each candidate web database. Then selection algorithm selects a most benefit web database $s_i$ to integrate from $S$ each time. This approach takes advantage of the fact that some web databases provide more benefit 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 efforts on them is unnecessary.

A selection algorithm has been defined to make the choice using benefit function automatically. The algorithm is made up of three elements:

1.A set $S = \{s_1, s_2, ..., s_n\}$ of candidate web databases and the status of integration system $D$.

2.Selection algorithm obtain the most benefit web database from $S$ by benefit function. This benefit function uses web database $s_i$ and the status of integration system $D$ as parameters.

3.A most benefit web database $\{u\}$ selected $\{u \in S\}$.

**Algorithm to select most benefit web database:**

**Selection Algorithm(**$D$: The status of integration system; $S$ : Set of candidates web databases**)**

$u = \phi$; $uBenefit = 0$;

$u = s_1$; //$s_1$ is first web database in $S$

$uBenefit = Benefit(s_1, D)$;

**foreach** $s_i \in S$ **do** //$s_i$ in $S$ which index is i;

**if** $Benefit(s_i, D) > uBenefit$ **then**

$u = s_i$;

$uBenefit = Benefit(s_i, D)$;

**end if**

**end foreach**

**return** $u$;

The selection algorithm select a most benefit web database from $S$ each time, $Benefit(s_1, D)$ is obtained by equation 1. Integration algorithm integrate the most benefit web database that is selected by selection algorithm each time. The integration algorithm as follow:

**Algorithm to hidden web database 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**

$s$ =Selection Algorithm($D, S$);

$D$=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, $Benefit(s_1, D)$ 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 return the most benefit web database for user integration. Finally, if the number of integrated web database equal 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 3 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 creates $Q(s_i)$, $|Q(s_i)|$ and $size(s_i)$ for each $s_i$ in $S$, and the system stores them in lists. In equation 2, $Q(D)$ 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

integrate k web databases, denoted $D_k$ . $Q(D_k)$ can be expressed by the following recursive formula.

$$Q(D_k) = Q(D_{k-1}) \cup Q(s_k) \quad (11)$$

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.

$$Q(D_i) = \bigcup_{j=1}^{|k|} (Q(s_j)) \quad (12)$$

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

Through the equation 12, we are able to effectively obtain $Q(D)$ and $|Q(D)|$ avoiding the cost of executing query workload $Q$ over $D$.

## V. 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 a query generator to randomly generate a set of queries. Each generated query refers to a single element and is representative of the set of queries that refer that element. For simplicity, the generator only produces keyword queries. In the experiment, we produce a 500 random queries of query workload.

In order to validate the effectiveness of our approach, we compare a variety of candidate web database selection and integration strategies, each strategy selects m(m=20) web databases to integration.

Benefit-based: m web databases are selected and integrated with our approach.

Quality-based: Users select web databases to integrate based on their quality[11]. In this paper, the quality of web database is measured only depending on objective criteria in [11]. This strategy selects top-m high quality web databases to integration.

Real-rank: To obtain the actual selection and integration result, this strategy uses the manual analysis for all web databases to determine which m web databases can be selected. The integration system has the least of overlap and the amount of data produced by this strategy. Note that this strategy is not a realistic approach, as it relies on knowing the actual content of all web databases. It is a upper-bound on any myopic strategy.

Random: Finally, the naive strategy is to treat each candidate web database as equally important. Thus, the web database is selected randomly. This strategy provides a baseline to which the above strategies can be compared.

### B. Experimental Results

To study the basic efficacy of our approach, the first experiments we present investigate the performance of different selecting strategies on the datasets. we use Kendall's method to evaluate the effectiveness of method proposed in this paper. $p1$ denotes the order sequence of the web database selection created by Real-Rank. $p2_i$ shows web database order sequences of the benefit-based selection strategy, the quality-based strategy, and the random selection strategy, In $p1$ the two web databases have very the strict sequence, the same to $p2_i$. If the sequence $p1$ and the corresponding $p2_i$ have the same order couple, we can say it coordination, or say it incompatible. $p$ is the record number of coordination, and $k$ records the number of uncoordinated couple. The Kendall's distance between $p1$ and $p2_i$ is: $(pk)/(p+k)$. The results are shown in Table 1, we can see that the benefit-based selection strategy has made a good performance relatively. Kendall's value has reached approximate 0.82.

TABLE I.
TABLE PERFORMANCE OF VARIOUS WEB DATABASE SELECTION STRATEGIES

| Web database selection strategy | Kendall's |
|---|---|
| Benefit-based | 0.82 |
| Quality-based: | 0.58 |
| Random | 0.35 |

The experiments above show that the benefit-based strategy is effective. In order to validate our benefit-based strategy in an even wider range of scenarios, we compare the amount of data and degree of overlap data in integration systems, which are obtained by integrating selected web databases by different strategies.

The result of the percentage of the amount of data in integration system and 80 candidate web databases is shown in Figure 2. Here the amount of data does not count duplicate of data. The results in this graphic can be interpreted as follows. Since m web databases can be selected in candidate datasets, the goal is to obtain the most data in integration system. Thus, the higher the column, the better the selecting.

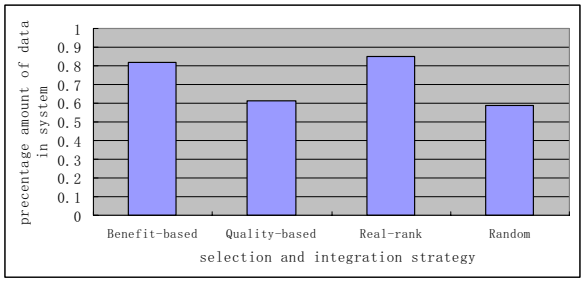


Figure 2. .Percentage the amount of data

First observe the column for the real-rank strategy. This approach selects the most beneficial candidate web databases to integrate and thus the column is very high. As can be seen, our strategy performs comparatively well: it tracks the real-rank strategy much more closely than any of the other strategies. In contrast, the height of the columns for the other strategies are much shallower; it contain much less data. The amount of data in integration system is least for the random strategy since it treats each candidate data sources as equally important.

Figure 3 shows the degree of overlap data in the integration systems that are produced by different strategies. Benefit-based strategy performed better than quality-based and random strategies. The degree of overlap by benefit-based strategy is lower than that of them-it is only 72%, it tracks the real-rank strategy. The most serious the degree of overlap is caused by quality-based strategy.

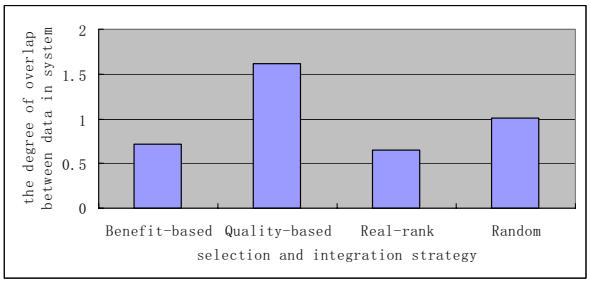


Figure 3. The degree of overlap data in the integration systems

## VI. CONCLUSION

This paper proposed an approach to the problem of selection and integration hidden web database. The approach is to select and integrate web databases in an iterative manner, where web databases are integrated incrementally. The core of this approach is a benefit function that estimated how much benefit the web database bring to a given status of an integration system by integrating it. Finally, We evaluate our method over real-world hidden web databases. Preliminary results are promising, they shows that despite our approximations, our approach is effectively to select and integrate web databases. The integration system with more high-quality data and the lower degree of overlap are produced by our approach. and our approach is low-cost, it can be used for internet-scale hidden web data integration tasks.

## ACKNOWLEDGMENT

This research was partially supported by The Natural Science Foundation of China under grant No.60673092; The 2008 Jiangsu Key Project of science support and self-innovation under grant No. BE2008044; The Opening Project of Jiangsu Province Software Engineering R&D Center for Modern Information Technology Application in Enterprise under grant No.SX200904 and No.SX200907; The Scientific Research Foundation for the Young Teachers, Suzhou Vocational University under grant No. SZDQ09L08; The Suzhou Project of High-skilled Personnel Training R&D under grant No. GJN091206.

## 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] 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.

[4] D'Souza, J. Zobel, and J. Thom, "Is CORI Effective for Collection Selection an Exploration of parameters, queries, and data," *Proceedings of Australian Document Computing Symposium(ADCS2004),* pp.41-46, December 2004.

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

[6] Luo Si,J.P.C., "Unified Utility Maximization Framework for Resource Selection," *Proceedings of ACM CIKM Conference(CIKM2004),* pp.32-41, November 2004.

[7] Shokouhi,M., "Central-Rank-Based Collection Selection in Uncooperative Distributed information Retrieval." *Proceedings of the 29rd European Conference on Infromation Retrieval(ECIR2007),* pp.160-172, April 2007.

[8] Zhenyu Liu, Cl.,Junghoo Cho, Weslesy W. Chu, "A Probabilistic Approach to Metasearching with Adaptive Probing," *Proceedings of the international Conference on Data Engineering(ICDE2004),* pp.547-559, March 2004.

[9] P.lpeirotis, L.G., "When one Sample is not Enough: Improving Text Database Selection Using Shrinkage," *Proceedings of the ACM SIGMOD International Conference On Management of Data(ICMD2004),* pp.767-778, June 2004.

[10] F. Naumann, J. C. Freytag, and M. Spiliopoulou. "Quality-driven Source Selection Using Data Envelopment Analysis," *Proceedings of the 3rd Conference on Information Quality(ICIQ1998),* pp.137-152, October 1998.

[11] Ashraf Aboulnaga and Kareem El Gebaly. "μBE: User Guided Source Selection and Schema Mediation for Internet Scale Data Integration," *Proceedings of the IEEE International Conference on Data Engineering(ICDE2007) ,*pp.186-195*,* April 2007.

[12] Ling Yan-Yan, Meng Xiao-Feng, Liu Wei. "An Attributes Correlation Based Approach for Estimating Size of Web Databases." *Journal of Software,* vol. 19, no.2, pp.224-236,2008.

[13] Redman T.C. "Data Quality for the Information Age," *Artech House,*1996.

[14] Wang, R.Y. Strong, D.M. "Beyond accuracy: What Data Quality Means to Data Consumers." *Journal of Management Information Systems,* vol. 12, no. 4, pp.5-33,1996.

[15] S Knight, J Burn, "Developing a Framework for Assessing Information Quality on the World Wide Web," *Informing Science Journal,* vol. 8, pp.159-172, 2005.

[16] Pipino L., Lee Y., Wang R. "Data Quality Assessment," *Communications of the ACM,* vol. 45, no. 4, pp.211-218,2002.

**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.

**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.

**Xin Jie** received his Master degree in computer science from University of London, London, England, in 2006. Currently, she is working on doctoral degree at Soochow University. Her research interests include information integration, data minning.

**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.