Send Orders for Reprints to reprints@benthamscience.ae

The Open Cybernetics & Systemics Journal, 2015, *9,* 375-383

Open Access

Web Database Sampling Based on Dependency of Keywords

Zhang Rui[1*], Wang Feng[1] and Lin Peiguang[2]

[1]*Shandong Provincial Institute of Electronic Information Products Inspection, Jinan, 250014, P.R. China*

[2]*School of Computer Science & Technology, Shandong University of Finance & Economics, Jinan, 250014, P.R. China*

Abstract: The Information Era has witnessed a huge number of sources from websites. The abundance of useful data surrounding us has made it possible for integration systems to improve the quality of the integrated data. However, how to choose proper data sources efficiently to extract data with high coverage and low redundancy is still a hot topic in the area. Sampling the databases hiding behind the websites makes it possible to obtain the characteristics of the web databases, and further to choose appropriate sources when collecting data for integration and query optimization. In this paper we construct a sampling model to represent data characteristics of web databases based on posing keyword queries on the deep web query interface. The dependency of text attribute keywords within the data source is used to construct the dependent-relational probability matrix, which indicate the sample distribution and is used for keyword extension to fetch more sampling data and get new characteristics of the actual data. Further, we provide an efficiency method to evaluate the similarity between the sample databases and the real web databases. We evaluate the proposed method in real world dataset and the results show that our method can sample the web data sources with high similarity.

Keywords: Data sampling, dependency of keywords, feature similarity, web database.

1. INTRODUCTION

A substantial quantity of structured data resides in the Deep Web. Therefore, Deep Web data extraction, integration and indexing have attracted significant research attentions and continue to represent a frequently addressed topic in the area. However, how to choose proper data sources efficiently to collect data with high coverage and low redundancy is still a hot topic in the area. Sampling the databases hiding behind the websites makes it possible to obtain the characteristics of the web databases, and further to choose appropriate sources when collecting data for integration. We have conducted some research in source selection [1]. In studies on the selection of data sources, most methods assume that the content of the data sources is known [2, 3]. However, due to the essence of big volume and velocity of web data, it is hard to know the content beforehand. Sampling the web sources with limited random selected data is a practical way to help select sources. The prerequisite is that the sample data and the real source have high consistency in characteristics. The data source characteristics can reflect the related distribution and parameters of data rules in the Web database. These distributions or parameters include text attribute keywords and frequency distributions, numerical attribute distribution, the category attribute and its statistical characteristics and the dependency relationship between the properties.

One of the authors has discussed text, numerical and category attributes and a samples collection method elsewhere [4]. As an extent research on that, we further consider the dependencies between internal keywords in this paper, based on which we construct a dependent-relational probability matrix. From the matrix, we obtain the distribution characteristics of sample data and propose a keyword extension method to fetch more sample data, which represents new characteristics of actual data.

We conclude the main contributions of this paper as follows:

(1) We construct a sampling model to represent the characteristics of data sources by posing keyword queries on the deep web query interface and evaluate the quality of sample data.

(2) We propose a method to fetch high quality sample data for web sources. We define the dependency matrix that reflects the dependencies between keywords of text attributes, from which we obtain the distribution of sample data. Based on it we propose a method to extend keywords for fetching more sample data.

(3) We evaluate the similarity between the sample data and actual data hiding behind the website with real world data. The results show that our method can extract high quality sample data.

The remainder of the paper is organized as follows: Section 2 defines related definitions and formulizes the problem. Section 3 illustrates the sampling model and discusses the method to fetch high quality sample data. Section 4 presents experiments to compare and verify the proposed method. Section 5 discuses related researches in the literature. Section 6 concludes the paper.

1874-110X/15

2. RELATED WORK

There are a series of researches about deep web source selection in recent years [5-8]. To reduce the cost of accessing data from a database, a substantial quantity of research has been performed on the traditional database sampling method. However, less research has focused on Web database sampling. As Deep Web research has developed and the requirements for a Web data integration system have increase, the demands on Web database sampling have been gradually increasing. Because Web databases are hidden behind the FORM-based query interface, the sampling for Web databases generally adopts query-based uniform sampling methods [9]. Reference [10] proposed the sampling method HIDDEN-DB-SAMPLER. This sampling method solved the problem of how to effectively obtain a uniform set of samples using the front query interface from a given Web query interface. To obtain a sample set, this method randomly selects a combination of attribute values as a query condition, which results in a substantial number of overflow and underflow results with a lower efficiency. Reference [11] processed the TOP-k-COUNT query interface with a restriction on the number of return query results. These two sampling methods have a common hypothesis: the Web database records are independent with identical distribution and contain only the category attributes. However, most Web databases contain text and scope attributes, and their records are not uniform distribution, which means different records have different sampled probability during the sampling process. That is, each record may be sampled in a different manner (query condition) simultaneously. Reference [12] proposed an incremental method WDB-Sampler for Web database sampling. This method eliminates the influences of attribute manifestation in the query interface. The same study proposed a new graph model of a Web database and performed the incremental sampling of a Web database by graph travel. The sampling query in this method is based on the keyword query of a single attribute, which causes a significant number of repeat records to appear in the sampling results and decreases the efficiency.

The studies discussed above represent the typical Web database-sampling methods published in recent years. However, this sampling method requires improvement in many respects, such as the sampling independence, the flexibility of the parameters configuration and the integrity of the results that reflect the database content [13-15] In addition, these random sampling methods cannot ensure consistency between the sampling data and the WDB data, such as the consistency of keywords in text attributes, the consistency of numeric distribution in numerical attributes and the integrity of text attributes. Moreover, these methods do not consider the dependencies among attributes in the WDB. Furthermore, most of the various methods to select data sources consider the related characteristics of the terms in data sources. Therefore, to achieve optimal data source selection in Web data integration, it be required that in the premise of the uniform distribution, the samples must reflect the distribution characteristics of the data sources' own data as much as possible and then provide the necessary parameters for the data source selection or other optimisation operations.

3. PROBLEM DEFINITION

This paper focuses on an independent data sampling method of the Web database that uses text attributes query interface. Here '*independent'* means to reflect the related characteristics of data sources to maintain unbiased sample data with regard to actual data. We first define some notations used in this paper before we formally define the problem to solve.

Definition 1: a $WDB=\{r_1, r_2, \ldots r_N\}$ denotes a Web database, r_i $(i = 1, \ldots, N)$ is a record residing in the *WDB* and the number of records in the database is N.

Definition 2: a $QI=\{ A_1, A_2, \ldots, A_m \}$ denotes a query interface that contains the query condition attributes A_i $(i = 1, \ldots , m)$, which are denoted as *AttrQ*, m is the number of attributes in the query interface.

Definition 3: a $D_i= \{v_{i,1},v_{i,2}, \ldots, v_{i,mi}\}$ denotes the range of values attached to attribute A_i, that is, attribute A_i of a *WDB* has m_i different values.

Definition 4: a $Q = \{A_1=v_1, A_2=v_2, \ldots, A_p=v_p\}$ denotes a query request on *QI* defined above. When we pose a query to a query interface, the *WDB* returns a result set $R=\{r_1, r_2, \ldots, r_n\}$, where n is the number of query results returned by the *WDB*. Suppose N' is the number of records in the *WDB* that satisfy the query condition, obviously, $n \le N'$. The attributes $\{a_1', a_2', \ldots, a_m'\}$ contained in query results R are denoted as $Attr_R$.

Definition 5: *Consistency* between sample data and actual data means that there are a substantial number of similar characteristics, including the data proportion of a returned category attribute, the data proportion of a returned text attribute, keyword distribution and the distribution of a returned numerical attribute in both data sets. In this paper, we focus on text attribute consistency. Suppose that *QI* contains only a text attribute, *QI*'s range is $C=\{k_1, k_2, \ldots, k_p\}$ and *QI*'s characteristics are as follows:

$$L = \begin{pmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,p} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ l_{p,1} & l_{p,2} & \cdots & l_{p,p} \end{pmatrix} \quad (1)$$

$l_{i,j}$ denotes the number of records that satisfy keyword k_i and k_j. Obviously, the matrix is symmetric, and the records distribution that satisfies single keywords and two-keyword combinations and the dependencies between two keywords can straightforwardly be obtained using this matrix. We name the matrix as *NDM* (*Number Dependency Matrix*).

Problem: giving a deep web, how to sample the actual data hiding behind it with high consistency with limited sample data.

To solve the problem, the sampling process can be divided into three stages as follows:

(1) Construct queries based on the query interface and known knowledge.

To acquire Web database query results from the query interface, one requirement is to choose appropriate keyword values for one or more attributes in the query interface. Without loss of generality, a WDB is treated as a data table, and a query request for a WDB is treated as a select query for the corresponding data table. Based on the definition of a query above, the *SQL-style* query is as follows:

Select $a_1^{'}, a_2^{'}, ..., a_m^{'}$ *from a WDB*

where $a_1 \oplus v_1$ *and* $a_2 \oplus v_2$ *... and* $a_p \oplus v_p$

$\oplus$ denotes the operator in *SQL*, which corresponds to the relational operators in SQL, such as "like", "=" and ">".

For arbitrary condition attribute a_i, if this attribute belongs to the category attribute, it is straightforward to acquire all possible values and construct an attribute range using the query interface directly and then assign values to "a" one by one based on the range to construct a query request. For a text attribute, we cannot enumerate all the possible values of the attribute. Therefore, the appropriate strategy must be defined to obtain possible values from a Web database. This paper focuses on a text attribute-oriented independent data sampling method for a Web database and provides an auto-expansion method for text attribute keywords.

(2) Acquire a query result and sampling a WDB

After a query request is constructed, query results can be acquired by submitting the request via the query interface. The results might be all of the query results that satisfy the query request or only the top-K results. Particularly for the latter result, if more auxiliary information regarding the Web database, such as database size and the quantity of return data, cannot be obtained (although random sampling of the database can be completed), the consistency of the characteristics between the sampling data and the Web database cannot be guaranteed. Therefore, the corresponding optimisation strategy must be studied further to enable the sampling results to represent the actual characteristics of the database to the greatest possible extent.

(3) Evaluating the sampling result

After obtaining the sample data in stage two, we need to evaluate if the sample data is consistent in characteristics with the actual data. In this paper, the primary target of the WDB sampling is to ensure consistency between the sampling data and the *WDB* characteristics. That is, a user can obtain results that exhibit a consistent data distribution by submitting a query request through the query interface to a *WDB* or the sampling data. We provide the following evaluations of the sampling results.

(I) Sample quality evaluation (Q1). It is measured by the degree of consistency (also referred to as similarity) between the characteristics of the sample data and the actual data of the WDB. The consistency degree is calculated as follows:

$$Q_1 = \frac{\sum_{i=1}^{m} (Fea_{a_i}(S) \otimes Fea_{a_i}(WDB))}{m} \quad (2)$$

where m is the number of the condition attributes of QI, $a_i \in Attr_Q \cap Attr_R, (1 \le i \le m)$, S denotes the sampling set; $Fea_{a_i}(S)$ denotes the characteristics of sampling set S in attribute ai; $Fea_{a_i}(WDB)$ denotes the characteristics of the WDB with respect to attribute ai and $\otimes$ denotes the similarity of the characteristics of two attributes.

The characteristics similarity of a text attribute is expressed by the average value of each row vector in the NDM matrix. Obviously, $0 \le Q_1 \le 1$.

(II) Sampling efficiency evaluation (Q2). It is the number of query executions required to obtain a certain number of samples (e.g., 1,000).

4. SAMPLING DATA BASED ON KEYWORD DEPENDENCY

In this section, we discuss the proposed sample model and method to acquire high quality sample data for a giving web source.

4.1. Basic Idea

When the condition attribute is a text attribute, such as a book or job title, generally, any record r_j may be retrieved using different text keywords. In addition, unlike the category attribute, for text attributes, the user is allowed to input multiple keywords to execute a fuzzy search.

Suppose a is a text attribute, its range is $D=\{k_1, k_2, ..., k_p\}$, where p denotes the number of keywords in D. We take each value individually in range D as a query keyword to retrieve results from the *WDB*. We denote the returned query sets as $R_1, R_2, ..., R_p$, where the number of records in each result set is $n_1, n_2, ..., n_p$ and the number of total records contained in the *WDB* is N. Obviously, $\sum_{i=1}^{p} n_i \ge N$. Furthermore, we use $l_{i,j}$ as the number of records that match keyword $k_j (1 \le j \le p, i \ne j)$ in result set $R_i (1 \le i \le p)$ and construct the following matrix:

$$L = \begin{pmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,p} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ l_{p,1} & l_{p,2} & \cdots & l_{p,p} \end{pmatrix} \quad (3)$$

The above matrix illustrates the number dependency matrix of a text attribute, or *NDM* (*Number Dependency Matrix*). Based on this matrix, it is straightforward to deduce the following conclusions:

We call the *NDM* matrix that satisfies the above conditions as *complete NDM matrix* and otherwise as *local NDM matrix*.

$$l_{i,j}=l_{j,i}, n_i=l_i, i=\sum_{j=1}^{p} l_{i,j}(j\neq i)$$
$$P(k_i)=\frac{l_{i,i}}{N}, P(k_j\mid k_i)=\frac{l_{i,j}}{l_{i,i}} \quad (4)$$
$$P(k_j\mid k_i)\times P(k_i)=P(k_i\mid k_j)\times P(k_j)=P(k_i,k_j)=\frac{l_{i,j}}{N}$$

For any record r_j in a query result set, the total probability of sampling is as follows:

$$P(r_j)=\sum_{i=1}^{p} p(r_j\mid k_i)\mid p(k_i) \quad (5)$$

$P(k_i)$ denotes the probability that k_i is selected as a query keyword and can be written as $P(a=k_i)$, whose value is defined as $P(k_i)=\frac{n_i}{N}, P(r_j\mid k_i)$, which means that when k_i is the selected keyword, the probability of accurately sampling the record r_j is $\frac{1}{n_i}$.

In addition, $l_{i,j}(i\neq j)$ of the matrix denotes the frequency of the co-occurrence of keywords k_i, k_j, if these co-occurrence records among all of the returned records are treated as one identical record. Then, the number of distinct records N' should be as follows:

$$l'_{1,1}=l_{1,1}$$
$$l'_{i,i}=l_{i,i}-\sum_{j=1}^{i-1} l_{i,j}, 1<i\leq p \quad (6)$$
$$N'=\sum_{i=1}^{p} l'_{i,i}$$

Thereby, we can generate the transform *NDM* matrix for sampling:

$$L'=\begin{pmatrix} l'_{1,1} & l_{1,2} & \cdots & l_{1,p} \\ 0 & l'_{2,2} & \cdots & l_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & l'_{p,p} \end{pmatrix} \quad (7)$$

It can be observed that, ideally, to evaluate whether the samples in *S* are consistent with the original *WDB*, we only require the formula $L'\times\frac{S}{N'}$. Thereby, we obtain the number of records that must be sampled for each keyword *k* and the number of samples for other keywords responding to keyword *k*. Next, we randomly sample records from the returned records.

Based on the transform *NDM* or the process described above, it is straightforward to establish a sampling tree (Fig. **1**). The root node of the tree is a virtual node, and the second layer nodes are all of the possible values of attribute *a*, i.e., k_1, k_2, ..., k_p. The edges between the root node and any second-layer node k_i are marked as the number of records that satisfy the keyword query k_i. Similarly, using each second layer as root can create third-layer nodes. The value of the edge connected to the parent node is the number of records that satisfy the corresponding value of the third-layer nodes only on the condition of satisfying the parent nodes. The leaf nodes are the records that satisfy the keywords on the path from the root to the specific leaf.

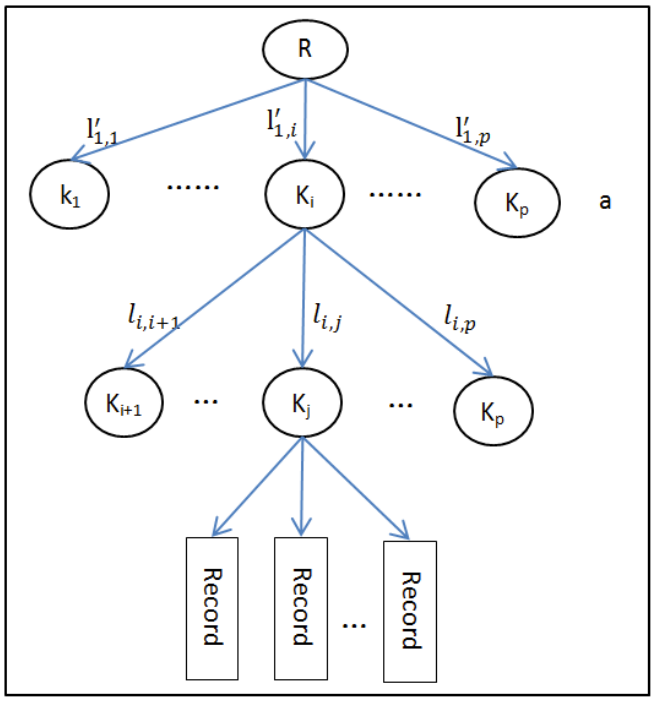


Fig. (1). Text sampling tree.

Based on the sampling tree, it is straightforward to complete the sampling process for a single text attribute. First, based on the first layer edge, count the total number of records $N'=\sum_{i=1}^{p} l'_{i,i}$. Second, calculate the samples number for each keyword. That is, modify the weight of the first layer edge $m_i=S\times\frac{l'_{i,i}}{N}$. Third, for each node of the second layer, calculate the samples number for each keyword in the third layer $m'_{i,j}=m_i\times\frac{l'_{i,j}}{l'_{i,i}}$. Finally, randomly collect $m'_{i,j}$ records from the leaf nodes.

4.2. Limitations and Improvement

The sampling process for the text attribute described above is the ideal process. However, in practical application, the sampling process has the following limitations.

(1) The WDB will return only the top-K records when the WDB contains a large quantity of data.

(2) The process is unable to obtain all of the valid keywords [10].

The two problems may violate the above conclusions and generate post-sampling bias. A solution to the first problem is to submit a multi-keyword combination based on the characteristics of the text attribute while enhancing the condition restriction to decrease the number of records that satisfy the condition and limiting the number of returned records in the top-K. However, this method increases the complexity of the query. Therefore, in this paper, the process is based on the actual return results of the WDB. For limitation (2) about the keywords problem, we propose a method to obtain more keywords to achieve consistency with the distribution characteristics of WDB keywords:

(I) Since keywords in a WDB are unknown, we begin from any single keyword for which query results were returned. This keyword is set as k_1.

(II) We construct and submit a query request using k_1 and obtain the query result set that satisfies this request from the WDB.

(III) We select a new keyword from the obtained query results for the next query. Results returned by the new keyword may contain more new keywords.

(IV) Repeat steps (II) and (III) until the termination condition is satisfied, that is, a new unknown keyword cannot be obtained. The key issue to be solved here is how to select the next query keyword from the query results.

If k_1 is assumed to be a query request and submitted, the query results R_1 are returned from the WDB, which contains n_1 records. R_1 contains attribute a, which is the same as the attribute of the query condition. We process the word segmentation for all of the values of the attribute a in R_1. Suppose that we obtain keywords $k_2, k_3, ..., k_p$ exclude k_1. Then, from n_1 records in set R_1, we obtain the number of records that contain the keywords $k_2, k_3, ..., k_p$. Next, we obtain $P_{1,i} = P(k_i \mid k_1) = \frac{n_i}{n_1}, (1 \le i \le p)$. Using this result, we can further count the number of records $n_{i,j}$ that contain keyword $k_j (2 \le j \le p)$ in the records that satisfy query keyword $k_i (2 \le i \le p)$ and obtain $P_{i,j} = P(k_j \mid k_i) = \frac{n_{i,j}}{n_i} (2 \le i \le p, i < j \le p)$. In addition, we define $P_{i,i} = 1 (2 \le i \le p)$. Based on the above steps, the following matrix is generated:

$$\begin{pmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,p} \\ / & P_{2,2} & \cdots & P_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ / & P_{p,2} & \cdots & P_{p,p} \end{pmatrix} \quad (8)$$

This probability matrix based on the dependencies between the internal keywords among the query results of single keyword is known as the *PDM (Probability Dependency Matrix)*. In this matrix, except for the elements of the first row and the first column, the elements constitute a sub-matrix, which reflects the dependencies between other keywords under the condition $a=k_1$. Based on the matrix, we select a keyword $k_i (i \ne 1)$ to increase the number of returned unknown keywords or non-retrieved records through the next query request with k_i.

In this matrix, if $P_{i,j} < P_{i',j}$, then compared to the records that satisfy keyword k_i, there are more returned records that satisfy keyword k_j among the records that satisfy keyword $k_{i'}$, which indicates that the records that satisfy keywords k_i contains more records that satisfy other keywords. The value of $|P_{i,j} - P_{i',j}|$ indicates the number of returned by k_i records different from k_j. In addition, for each row in the PDM, the more keywords that satisfy the condition $P_{i,j} < P_{i',j}$, the more keywords contain unknown data. Using this idea, this paper proposes a selection method based on the relative quantity of unknown information.

First, the formula for the quantity of unknown information contained by keyword $k_i (2 \le i \le p)$ is calculated:

$$\begin{aligned} & f_{i,i'|j} = \begin{cases} 1, P_{i,j} < P_{i',j} \\ 0, P_{i,j} \ge P_{i',j} \end{cases} \\ & IF_{i,i'|j} = |P_{i,j} - P_{i'j}| \\ & I(k_i, k_{i'}) = -\sum_{j \ne i, 1 < j \le p} f_{i,i'|j} IF_{i,i'|j} \log IF_{i,i'|j} \\ & I(k_i) = \max_{i'} (I(k_i, k_{i'})) \end{aligned} \quad (9)$$

Where $2 \le i' \le p, i' \ne i, 2 \le j \le p, \log 0 = 0$ in the above formula. We initialize the keyword list by adding k_1 as the first one. Based on above *PDM* matrix, the relative quantity of unknown information contained by keywords $k_2, ..., k_p$ is calculated and a keyword is selected that is not included in the keywords list and contains the largest quantity of unknown information. The selected keyword is added to the keywords list.

In addition, we define the termination condition. The condition is that the difference between the average and the maximum quantity of unknown information is in a given range ε (as in the following formula) or a keyword cannot be found in a new location.

$$|\max(I(k_i)) - mean(I(k_i))| \le \varepsilon, 2 \le i \le p \quad (10)$$

4.3. Data Sampling Evaluation

Based on the evaluation method of the text attribute sampling quality, we calculate the similarity between any two keywords. For text attribute a, suppose range of a is $D=\{k_1, k_2, ..., k_p\}$, where p denotes the number of keywords in D and q is the number of keywords obtained by sampling queries. Obviously, there is $q \le p$. Assume that the range of text attribute a is $D' = \{k'_1, k'_2, ..., k'_q\}$ and $n'_1, n'_2, ..., n'_q$ are the numbers of returned records by sampling queries based on each keyword.

We can adjust the keyword sequence in range D of text attribute a, to produce the same sequence with the keywords in D'. Thus, the *NDM* of samples L' can be generated as follows:

$$L' = \begin{pmatrix} l'_{1,1} & l'_{1,2} & \cdots & l'_{1,q} \\ l'_{2,1} & l'_{2,2} & \cdots & l'_{2,q} \\ \vdots & \vdots & \ddots & \vdots \\ l'_{q,1} & l'_{q,2} & \cdots & l'_{q,q} \end{pmatrix} \quad (11)$$

With the expansion of samples range D' to the entire domain D, L' transfers to the $P \times P$ matrix as follows:

$$L'' = \begin{pmatrix} l'_{1,1} & l'_{1,2} & \cdots & l'_{1,q} & \\ l'_{2,1} & l'_{2,2} & \cdots & l'_{2,q} & O_1 \\ \vdots & \vdots & \ddots & \vdots & \\ l'_{q,1} & l'_{q,2} & \cdots & l'_{q,q} & \\ & O_2 & & & O_2 \end{pmatrix} \quad (12)$$

In the matrix, O_1 is the zero matrix $q \times (p-q)$, O_2 is the zero matrix $(p-q) \times q$ and O_3 is the zero matrix $(p-q) \times (p-q)$.

For any keyword k_i, calculate the keyword's similarity using the following formula:

$$sim(k_i) = \frac{\sum_{j=1}^{p} l_{i,j} \times l'_{i,j}}{\sqrt{\sum_{j=1}^{p} l^2_{i,j}} \times \sqrt{\sum_{j=1}^{p} l'^2_{i,j}}} = \frac{\sum_{j=1}^{q} l_{i,j} \times l'_{i,j}}{\sqrt{\sum_{j=1}^{q} l^2_{i,j}} \times \sqrt{\sum_{j=1}^{q} l'^2_{i,j}}} \quad (13)$$

Where

$$n_i \le \sum_{j=1}^{q} l_{i,j} \le qn_i, n'_i \le \sum_{j=1}^{q} i_{i,j} \le qn'_i, i_{i,j} \le l_{i,j} \quad (14)$$

And

$$\frac{(\sum_{i=1}^{n} a_i)^2}{n} \le \sum_{i=1}^{n} a_i^2 \le (\sum_{i=1}^{n} a_i)^2 \quad (15)$$

Thus,

$$\frac{n'_i}{qn_i} \le sim(k_i) \le 1 \quad (16)$$

Therefore, the sampling quality of text attribute a is expressed as follows:

$$\frac{1}{pq} \times \frac{n}{N} \le Q_1 = \frac{\sum_{j=1}^{p} sim(k_i)}{p} = \frac{\sum_{j=1}^{q} sim(k_i)}{p} \le \frac{q}{p} \quad (17)$$

5. EXPERIMENT

We evaluate the proposed sampling method with real-world dataset. In the following parts of this section, we set up the dataset and evaluate the method by analysing keyword collection algorithm and sampling quality.

5.1. Experiment Dataset and Process Introduction

(1) Experiment Dataset

The test dataset is a user information database that was downloaded from an open micro-blog platform. This data table contains 244959 records. After the irrelevant attributes are omitted, the following attributes remain: {id, name, province, description}, where description is a text attribute. The query interface is defined as {description} and the attributes results set as {name, province, description}.

(2) Experiment Process

The experiment process was divided into two phases. In phase one, we test the keyword extension algorithm in text attribute with unrestricted database access (i.e., no restriction on access time and the number of return results), test keywords expansion and relations with related parameters. In phase two, we test the sampling algorithm and evaluate the quality of sampling data and efficiency of the algorithm.

In phase one, we extracted three sub-libraries from the user information database, which contained 5000, 10000, 50000 records, respectively. To increase the test speed during the test process, in each iteration, new keywords were chosen from the selected 50, 100, 200 and 400 keywords in the query results. Otherwise, we defined the three values 0.3, 0.5 and 1 as threshold ε in the termination condition of the keyword extraction.

In phase two, based on the three sub-libraries described above, a 100000-record sub-library was added. We set a different sampling number for each library (Table **1**) and evaluated the quality and efficiency.

Table 1. Number of Sampling Data for Different Sub-libraries

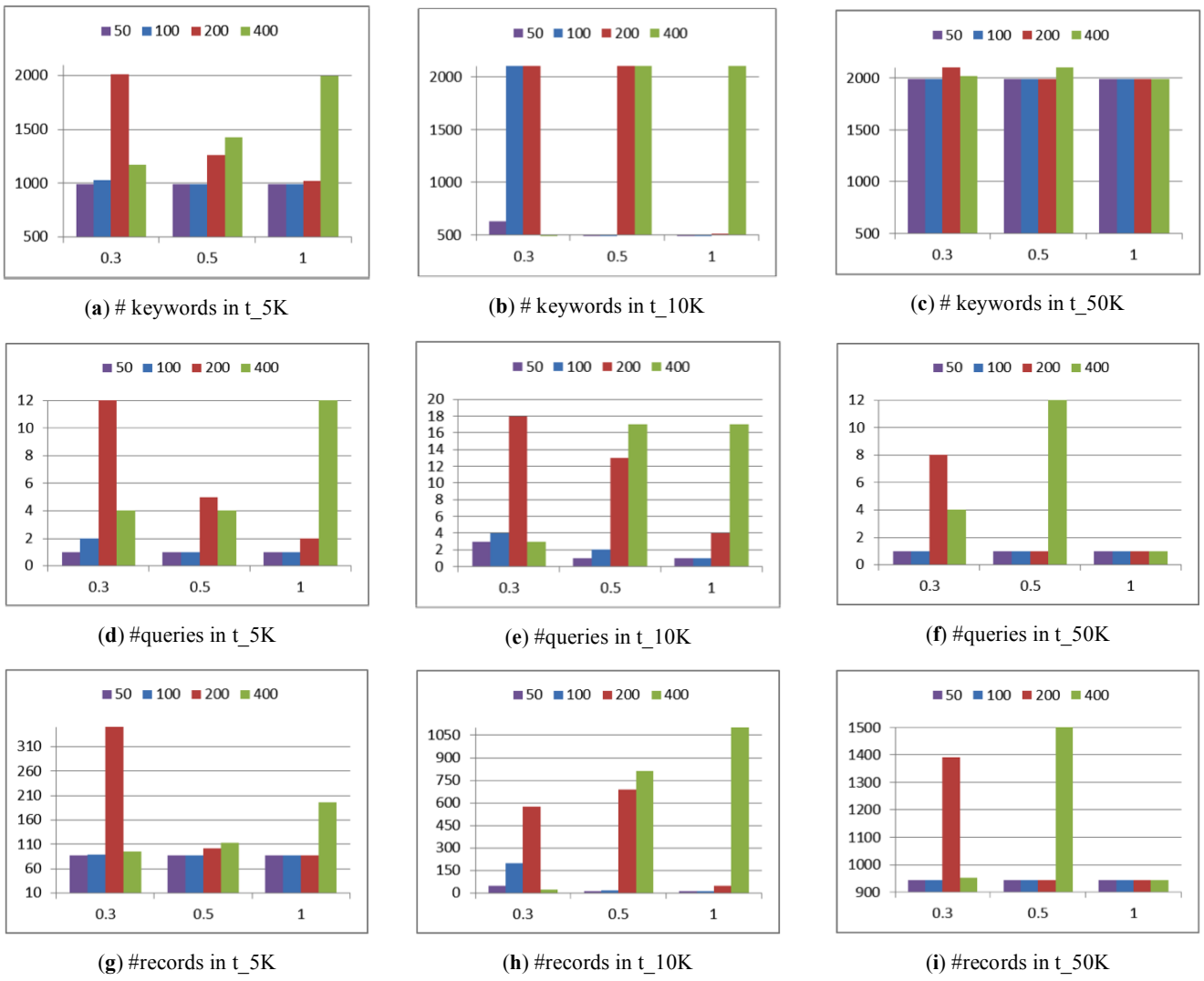
ID	Sub Dataset Name	Records Number	Sampling Number
1	t_5k	5,000	300, 500, 1000
2	t_10k	10,000	500, 1000, 2000
3	t_50k	50,000	1000, 3000, 5000
4	t_100k	100,000	2000, 5000, 10000

5.2. Experiment Results: Evaluation and Analysis

(1) Keywords collection algorithm analysis

The experiment results are shown in Fig. (**2**). For sub-libraries 1, 2, 3, which are defined in Table **1**, Figs. (**2a-c**) illustrate the number of keywords obtained for each sub-library respectively. Whereas Figs. (**2d-f**) illustrate the number of queries for each sub-library respectively, and Figs. (**2g-i**) illustrate the number of records related to the sampling.

It can be observed from Figs. (**2a-c**) that the sub-library t_50k had a maximum average return of keywords and was minimally affected by other parameters. Considering the number of filter keywords, 200 and 400 keywords



(**a**) # keywords in t_5K (**b**) # keywords in t_10K (**c**) # keywords in t_50K

(**d**) #queries in t_5K (**e**) #queries in t_10K (**f**) #queries in t_50K

(**g**) #records in t_5K (**h**) #records in t_10K (**i**) #records in t_50K

Fig. (2). Keywords extraction results for different sub-libraries.

displayed a more stable structure. Additionally, Figs. (**2d-f**) indicated that the sub-library t_50k had a minimal number of query requests, and Figs. (**2g-i**) indicated that the sub library t_50K had a maximum number of records returned in the keyword filter, particularly where $\varepsilon = 1$, which was obviously better than the other two sub-libraries and relatively more stable.

Based on the above results, more records in a database can improve stability. To consider filter efficiency, 200 keywords can be used as a filter set.

(2) Sampling quality analysis

The sampling efficiency without a restriction on the return number is shown in (Fig. **3**). In this figure, the number before the horizontal ordinate expresses the sampling number, whereas the numbers in brackets are the IDs of the sub-libraries shown in Table **1**. The figure indicates that the accuracy of the classified attributes remains basically unchanged for the reason provided in the above analysis. The accuracy of the text attributes tends to decrease with the increase in the database size because the loss rate of keywords increases gradually with the database size increase. However, the overall quality of the sampling generally remained approximately 90%.

Fig. (**4**) reflects the quality and efficiency of sampling with a query restriction. That is, (a) describes the similarity of text attributes when the number of query returns is limited to 200, 500 and 1000 records. Fig. (**4a**) uses the same horizontal ordinate as Fig. (**3**), and the vertical ordinate expresses the percentage. And Fig. (**4b**) illustrates the sampling efficiency of all of the previous sampling using the unit "times/thousands of records", which indicates the average query times for the sampling of 1000 records.

Fig. (**4a**) shows that for each line the similarity of text attributes overall tends to decline because where the query restriction is the same, the number of returned keywords decreases with the increase in database size. Based on the horizontal contrast, the similarity of text attributes increases gradually with the increase in the return number. As shown in Fig. (**4b**), the sampling efficiency is directly relative to the quality of the returned data and the original data.

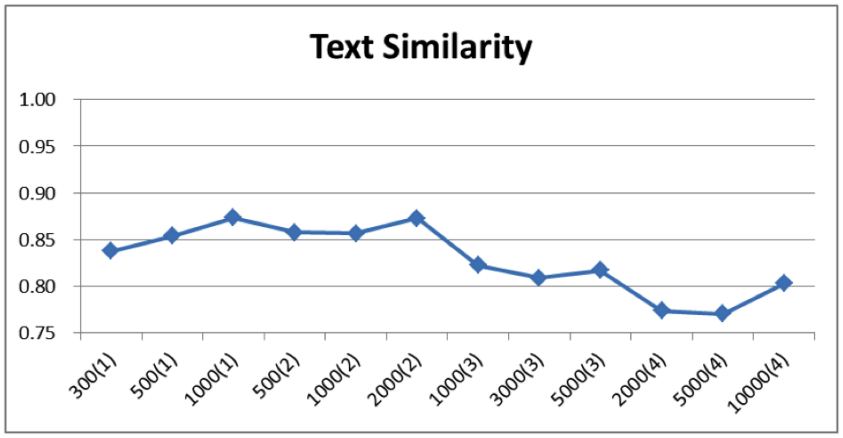
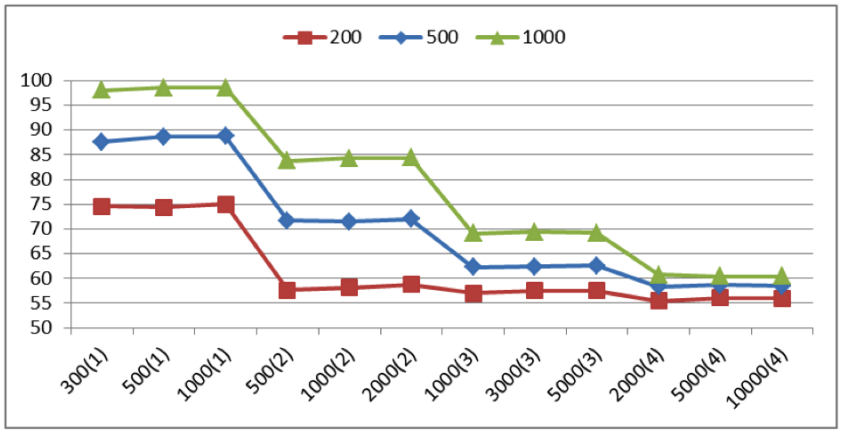
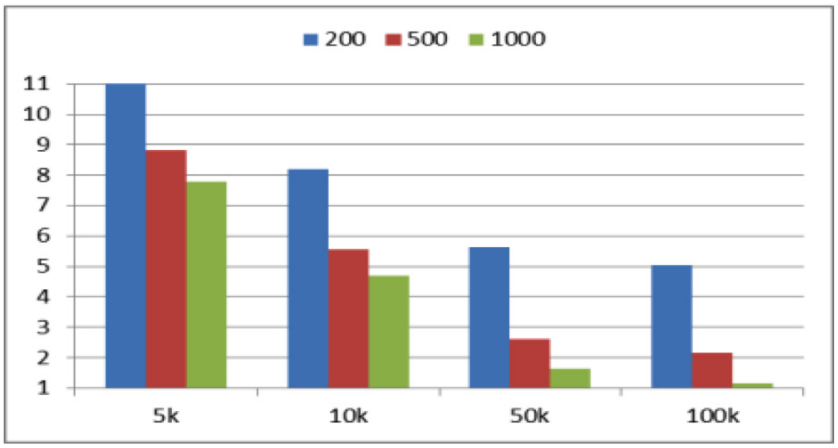


Fig. (3). Sampling quality without query constraints.



(**a**) Sampling quality with limited records



(**b**) Sampling quality with limited query time

Fig. (4). Sampling quality with query constraints.

6. CONCLUSION

For the rich structured data of the Web database, an independent sampling method that can reflect the internal data characteristics of a Web database has great significance for the operations of data source selection and query optimisation in data integration. Based on the text attributes in the Web database query interface, this paper proposed a new sampling method for independent data samples that uses the dependencies between internal keywords of the text field. To meet the requirements of practical applications, this method can preserve consistency between the characteristics of data samples and the Web database in the process of random sampling for the Web database.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] P.G. Lin, L. Zhao, Y. Zhang, and P.G. Nie, "Data sources selection based on WDB's characters and user's query, " *Journal of Computer Research and Development*, vol. 47, no. Z, pp. 36-42, 2010.

[2] T. Cheng, X. Yan, and K. Chang, "EntityRank: Searching Entities Directly and Holistically," In: *Proceedings of the 33rd International Conference on Very Large Data Bases*, ACM Press New York, 2009, pp. 387-398.

[3] F.J. Jiang, and X.F. Meng, "Survey of query processing in deep web data integration," *Journal of Frontiers of Computer Science and Technology*, vol. 3, no. 2, pp. 113-129, 2009.

[4] P.G. Lin, "Independent data sampling method for deep web based on the characteristics of WDB," *Journal of Computer Research and Development*, vol. 49, no. Z, 2012.

[5] X.L. Dong, B. Saha, and D. Srivastava, "Less is more-selecting sources wisely for integration," *PVLDB*, vol. 6, no. 2, pp. 37-48, 2012.

[6] Y. Kammerer, and P. Gerjets, "The role of search result position and source trustworthiness in the selection of web search results when using a list or a grid interface," *International Journal of Human-Computer Interaction (IJHCI)*, vol. 30, no. 3, pp. 177-191, 2014.

[7] T. Rekatsinas, X.L. Dong, and D. Srivastava, "Characterizing and selecting fresh data sources," *SIGMOD*, 2014, pp. 919-930.

[8] K.S. Kim, and S.C.J. Sin, "Selecting quality sources: Bridging the gap between the perception and use of information sources," *Journal of Information Science (JIS)*, vol. 37, no. 2, pp. 178-188, 2011.

[9] J. Callan, and M. Connell, "Query-based sampling of text databases," *ACM Transactions on Information Systems (TIOS)*, vol. 19, no. 2, pp. 97-130, 2001.

[10] A. Dasgupta, G. Das, and H. Mannila, "A random walk approach to sampling hidden databases," *SIGMOD*, ACM Press, New York, pp. 629-640, 2007.

[11] A. Dasgupta, N. Zhang, and G. Das, "Leveraging COUNT information in sampling hidden databases," *ICDE*, 2009, pp. 329-340.

[12] W. Liu, X.F. Meng, and Y.Y. Ling, "A graph-based approach for web database sampling," *Journal of Software*, vol. 19, no. 2, pp. 179-193, 2008.

[13] Z. Nie, and K. Subbarao, "A frequency-based approach for mining coverage statistics in data integration," *ICDE*, 2004, pp. 387-398.

[14] Z. Zhang, B. He, and K. Chang, "Light-weight domain-based form assistant: Query Web databases on the fly," *VLDB*, ACM Press, New York, 2005, pp. 97-108.

[15] F. Jiang, L. Jia, and X. Meng, "Query translation on the fly in Deep Web integration," *Wuhan University Journal of Nature Sciences*, vol. 12, no. 5, pp. 819-824, 2007.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014