Noname manuscript No.
(will be inserted by the editor)

Estimating Deep Web Data Source Size by Capture-Recapture Method

Jianguo Lu · Dingding Li

Abstract This paper addresses the problem of estimating the size of a deep web data source that is accessible by queries only. Since most deep web data sources are non-cooperative, a data source size can only be estimated by sending queries and analyzing the returning results. We propose an efficient estimator based on the capture-recapture method. First we derive an equation between the overlapping rate and the percentage of the data examined when random samples are retrieved from a uniform distribution. This equation is conceptually simple and leads to the derivation of an estimator for samples obtained by random queries.

Since random queries do not produce random documents, it is well known that the estimation by random queries has a negative bias. Based on the simple estimator for random samples, we adjust the equation so that it can handle the samples returned by random queries. We conduct both simulation studies and experiments on corpora including Gov2, Reuters and Wikipedia. The results show that our method has small bias and standard deviation.

Keywords Deep web, estimators, capture-recapture.

1 Introduction

The deep web [5] is the web that is dynamically generated from data sources such as databases or file systems. Unlike the surface web where data are available through URLs, data from deep web data sources are guarded by search interfaces.

Estimating the size of a deep web data source is an important component of data source sampling which obtains a profile or summary of a data source [10] [11] [22] [36] [38]. Data source sampling has attracted much attention in numerous contexts, such as distributed or integrated information retrieval, data source selection and categorization, and peer-to-peer information retrieval. Given the proliferation of web data sources and

Jianguo Lu
School of Computer Science, University of Windsor. Canada
jlu@cs.uwindsor.ca

Dingding Li
Department of Economics, University of Windsor, Canada
dli@uwindsor.ca

web services, quite often there are many similar data sources serving the same purpose. There is a need to select the one that is most comprehensive. While data providers do have the details such as the size of the data source, the same information may not be available to a third party.

Estimating the data source size is also an indispensable step when crawling a deep web data source for the purpose of indexing, backup, and downloading. There has been extensive research on deep web crawling and data extraction by issuing queries [28] [30] [2] [25] [32] [33], and all of them need to decide as to when most of the data have been harvested. Without knowledge on the data source size, it is difficult to decide when to stop the crawling process, and how to evaluate the performance of the data extractors.

Since most of the deep web data sources can be accessed only by queries, the estimation is based on the queries issued and the results (either complete documents or document ids) obtained. This kind of query based data size estimation has been widely studied [3] [4] [6] [7] [9] [11] [34] [36] [38] [40]. One of the basic techniques is the traditional capture-recapture method [1] [12] [14] [35] in ecology. The idea is that if one takes several samples from a population, there would be some overlapping between the samples. According to the overlapping information, various estimation methods are proposed and applied in areas including data collection size estimation.

Our first *contribution* in this paper is the derivation of a simple equation between the overlapping rate and the percentage of the data examined. The equation holds only when the queries are fired many times, which is common in query based sampling, but may not be feasible for estimation problems in ecology. Although the equation itself can not be directly used in size estimation due to the assumption of the availability of uniformly distributed random samples, the simplicity of the equation leads to the derivation of a real estimator for samples obtained by queries.

One challenge in data size estimation is the difficulty in obtaining random samples from a data source, partially because documents have unequal probabilities of being retrieved. It is well known that random queries do not return random documents, and that sampling by random queries will consistently result in negative bias [4] [9] [34] [40]. Shokouhi et al observed that there is a fixed relationship between the estimation and the actual size, and proposed the use of regression to adjust the estimation result obtained by an estimator for random samples [34]. However, this estimator is not consistent– it will overestimate when the sample size is large as we will analyze in Section 5.4.

Based on our simple estimator for random samples, our second *contribution* is to adjust our estimator so that it can handle the samples returned by random queries. When documents have unequal probabilities of being captured, the degree of heterogeneity can be used to modify the estimation. Our simulation study shows that our estimator works very well if the degree of heterogeneity is known. We also conducted experiments on various corpora and show that our method compensates for this kind of query bias [6] very well.

Many data sources, especially large search engines such as Google, rank the matching results and return only top k elements. This kind of *rank bias* [6] can be tackled by downloading and analyzing the documents [4]. Since our method does not download the documents, it can't overcome the rank bias. For ranked data sources, it works well only when there are not many overflowing queries, i.e., the queries match more than k documents.

Our method is not intended for estimating huge databases such as general purpose search engines, where ranking and overflowing queries are common and can not be ignored. Rather, it is for unranked data sources, or ranked data sources with low density

of overflowing queries. These data sources are typically of a smaller size. Compared with the methods for ranked data sources, our estimator is rather efficient–we do not need to download and analyze the documents.

Another restriction of this paper is that we focus on textual data sources that contain plain text documents only. This kind of data source usually provides a simple keywords-based query interface, instead of multiple attributes as studied in [39].

2 RELATED WORK

2.1 Basic concepts of capture-recapture method

Capture-recapture method was originally developed in ecology and used to estimate the size of an animal population [1] [31]. In the estimation process, animals are captured, marked, and released in several trapping occasions. The data collected during the process, including the number of capture occasions, the recaptured animals, and the distinct animals captured, allow one to estimate the total population. This estimation process corresponds nicely to the query based estimation of data collection sizes, where a trapping occasion corresponds to sending a query and retrieving a set of documents from a data source.

If all the documents have an equal probability of being matched by a query, and all the matched documents are returned, we have the simplest model for which many estimators have been developed. The classic estimator is the famous Petersen estimator [31] that can be applied only to two capture occasions:

$$\hat{n}_{Petersen} = n_2 U_2 / d_2, \tag{1}$$

where n_2 is the number of documents retrieved by the second query, U_2 is the number of unique documents retrieved just before the second query, and d_2 is the number of duplicate documents that are captured after the second query.

This estimator can be derived using maximum likelihood method. The problem of this estimator is that d_2 could be zero when n_2 and U_2 are not large enough. According to the birthday paradox, in general n_2, U_2 should be greater than $\sqrt{n}$, where n is the actual size of the data source, in order to have overlaps between the results of two queries. Unfortunately, many queries do not have that many matches.

One approach to solving the problem is by obtaining two large samples, each are produced by many queries instead of just one query [9].

Another approach is expanding the estimator to multiple capture occasions or queries, and taking the weighted average of the estimations. i.e.,

$$\hat{n} = \frac{\sum_{i=2}^{q} w_i n_i U_i / d_i}{\sum_{i=2}^{q} w_i} \tag{2}$$

When weight $w_i = d_i$, it is the classical Schnabel estimator [31] for multiple captures:

$$\hat{n}_{Schnabel} = \frac{\sum_{i=1}^{t} n_i U_i}{\sum_{i=1}^{t} d_i} \tag{3}$$

When weight $w_i = d_i U_i$, it is the Schumacher estimator: [35]

$$\hat{n}_{Schumacher} = \frac{\sum_{i=1}^{t} n_i U_i^2}{\sum_{i=1}^{t} d_i U_i} \tag{4}$$

Unlike two capture occasions, the MLE (Maximum Likelihood Estimator) for multiple capture model does not have a closed form solution. Without surprise, both Schnabel and Schumacher estimators are approximate estimators. However, they are widely accepted as good estimators with very small bias and variance when animals (or documents) are captured with equal probability.

In reality, individuals, be it documents or animals, seldom have equal capture probability. For animals, young animals may be easier to be captured because they are more active. For documents, large documents may be easier to be retrieved by a query because there are more words in those documents.

For this kind of heterogeneous population where each individual has a unequal catachability, the estimation is notoriously difficult [1]. MLE technique can no longer be used to derive an estimator because there can be as many as n+1 parameters: n and and capture probabilities $p_1, p_2, \ldots, p_n$. Estimating this many parameters from the capture data is not possible. Although there are several empirical estimators proposed for this model, including the Jacknife estimator [31] and Chao [12] method, both can be only applied to small population with hundreds of elements, and require large sample size.

2.2 Estimators for data collections

Data source size estimation methods can be classified into two categories. One relies on the sample set(s) of documents and the lexical analysis of those documents [3][4][9], while the other analyzes the document ids only [6][16][34][38][40].

2.2.1 Methods based on document analysis

Estimation methods based on document analysis date back to a sample-resample method proposed by Si et al [36]. More recently fairly sophisticated methods such as [4] [3] [9] are proposed.

Taking Broder et al's work [9] for example, they first try to establish the sizes of two subsets of the corpus. A subset is the documents that are indexed by a query pool, which may contain millions of queries such as all the eight digit numbers. Since the query pool is rather large, it is impractical to fire all the queries to decide the number of documents that can be retrieved using the query pool. Hence, from the query pool a set of sample queries are randomly selected and sent to the data source. All the matched documents are downloaded and analyzed to obtain the weight [9] of each query. Based on the average weight of the queries from the sample, the number of the documents that can be obtained by the query pool can be estimated by multiplying the average weight by the query pool size. Once two such subsets of the documents are obtained, the Petersen estimator (Equation 1) is used to estimate the total corpus size.

Obviously this approach is rather inefficient and sometimes infeasible because

- Some deep web data sources may not support the downloading of the documents. For example, Amazon book search web service will return the basic information of the books, instead of the entire books. For this kind of data sources, it is impossible to use the returned documents to obtain the source profile.
- Even when the documents are downloadable, the estimation process is very expensive. In addition to the downloading of the documents, obtaining random documents

may incur further costs. For example, one of Bar-Yossef et al's methods [3] needs to fire 2000 queries in order to obtain one random document.
– The approach is not stable. Quite often the links provided by a deep web data source may not be active, and this may disrupt the estimation process.

2.2.2 Methods based on document ids

This paper focuses on another category of estimation methods, which rely on the returning documents ids instead of the entire documents. Document ids can be in various forms such as ISBN numbers for books in Amazon, URLs for web pages indexed in Google, or file names in our experiments. Compared with the bulk of the documents downloaded, this type of estimator only needs to know the document identifiers, which will save network traffic tremendously.

This approach originated from the Capture-Recapture method, which is extensively studied and widely used in estimating the population of wild animals [1]. The basic idea is to capture a collection of animals as randomly as possible, mark them and release them. Then capture another sample and count the duplicates with the previous captures. With this data various approaches are proposed to estimate the animal population size.

Liu et al proposed using the Capture-Recapture method to estimate a data source size [26]. More recently, Carverlee et al [11] discussed the problem in the setting of a distributed environment. The estimator they used is the traditional Peterssen estimator.

Shokouhi et al [34] proposed to use multiple capture-recapture method, or Capture with History (hereafter CH) method, to estimate a data source size. Using regression method, they developed a new estimator based on the traditional Schumacher and Eschmeyer estimator [35] as shown in Equation 4.

Bharat and Broder used a large query pool to estimate the sizes of search engines and the web [6]. In particular, they identified various biases, especially the *query bias* and *ranking bias*, during the estimation process. Gulli and Signorini [16] improved the method described in [6].

2.3 Compensate the bias

Capture-recapture based methods tend to underestimate the size of a data source due to several reasons. One is the assumption that documents (or the wild animals in the case of animal population estimation) are of the same probability of being captured. When using random queries to capture documents, the retrieved documents are actually not randomly selected.

Various methods have been proposed to compensate this kind of bias. In the traditional capture-recapture research, people proposed various estimators [1] [12] to cope with the population that have unequal catchability. In data source estimation, Bar-Yossef and Gurevich strive to obtain random samples [3][4] by downloading and analyzing the texts.

Shokouhi et al [34] corrected the bias of the CH method using regression (CH_{reg} hereafter). They conjecture that there is a fixed relationship between the initial estimation, which is obtained by capture-recapture methods, and the actual data size.

Based on this hypothesis, they use a training data set to obtain an equation between the initial estimation and the actual size using non-linear regression.

Xu et al [40] tried to compensate the bias by modeling the document capture probabilities with logistic regression.

3 AN ESTIMATOR FOR RANDOM SAMPLES

When random samples are obtained from a uniform distribution, the collection size can be estimated fairly accurately using various estimators such as Equation 4. However, it is difficult to adjust Equation 4 for samples obtained by queries due to its complexity. Hence we derived a simple equation as follows:

$$P \approx 1 - OR^{-2.1} \tag{5}$$

where P is the percentage of the samples obtained from the data source, OR (Overlapping Rate) is the ratio between the accumulative total number of data items and the unique data items in the samples.

More precisely, suppose that a data source has an actual size n. When a sequence of samples is obtained randomly from the data source, suppose that there are t accumulative data items, and u number of unique items. The overlapping rate OR is t/u and P is u/n.

3.1 Simulation study

Before deriving Equation 5, we run a simulation study in the context of a variation of urn model [21] with replacement.

Urn model: Consider an urn that contains n balls, each is labeled with a unique number from 1 to n. Randomly select k balls, where $k \ll n$, from the urn, record the ball numbers that are drawn, then put back the balls into the urn. Repeat the process for i times, in which stage $u(i)$ unique balls are selected, which constitutes $P(i)$ percent of all the balls in the urn. The overlapping rate $OR(i)$ is $t(i)/u(i)$.

The question is: What is the relationship between P and the overlapping rate OR, if there is any?

We first run an urn model simulation to reveal the relation.

Urn model simulation: In this experiment, n is the total number of data items in the data source, and k is the size of each sample. In each step, we generate k random numbers that are uniformly distributed within the range between 1 and n. Add those numbers to the result set and record the overlapping rate OR and hit rate P. Repeat the process until the cardinality of the result set is close to n.

Figure 1 draws a relationship between P and OR for step lengths ranging from 1 to 40 where n, the total of number data items, is 5,000. We tested other values of n and observed similar results.

The simulation indicates that the hit rate P is a function of the overlapping rate OR, independent of the step length k. This prompts us to derive the equation between P and OR.

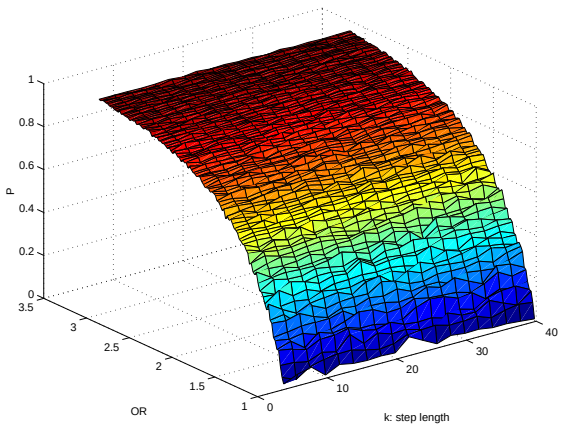

Fig. 1 Urn model simulation. n=5,000.

3.2 Derivation

$P(i)$, the percentage of the data obtained up to i-th iteration, can be also interpreted as the probability of one particular document that is captured in all i iterations. We start with:

$$P(1) = \frac{k}{n} \tag{6}$$

$$P(2) = \frac{k}{n} + \frac{k}{n} - \frac{k}{n} * \frac{k}{n}$$

$$\dots$$

$$P(i) = P(i-1) + \frac{k}{n} - P(i-1) * \frac{k}{n} \tag{7}$$

Solving the equations (6) and (7) we obtain the following:

$$\begin{aligned} P(i) &= 1 - \left(1 - \frac{k}{n}\right)^i \\ &= 1 - \left(1 - \frac{u}{n}\frac{ik}{u}\frac{1}{i}\right)^i \\ &= 1 - \left(1 - \frac{P(i) * OR(i)}{i}\right)^i \end{aligned}$$

Moving $OR(i)$ to the left hand side of the equation we have:

$$OR(i) = \frac{i * \left(1 - (1 - P(i))^{\frac{1}{i}}\right)}{P(i)} \tag{8}$$

The question we want to ask is: whether OR is a function of P? For example, given a predetermined P=10%, what is the OR at that point? Since i is assumed to be a large

number and P is fixed, the corresponding OR can be derived as follows:

$$
\begin{aligned}
OR &= \lim_{i\to\infty} \frac{i * (1-(1-P)^{\frac{1}{i}})}{P} \\
&= \frac{1}{P} \lim_{i\to\infty} \frac{1-(1-P)^{\frac{1}{i}})}{\frac{1}{i}} \\
&= \frac{1}{P} \lim_{i\to\infty} \frac{\partial(1-(1-P)^{\frac{1}{i}})}{\partial\frac{1}{i}} \\
&= \frac{1}{P} \lim_{i\to\infty} \frac{\frac{\partial(1-(1-P)^{\frac{1}{i}})}{\partial(i)}}{\frac{\partial(\frac{1}{i})}{\partial(i)}} \\
&= \frac{1}{P} \lim_{i\to\infty} \frac{\frac{\partial(1-(1-P)^{\frac{1}{i}})}{\partial(\frac{1}{i})} \times \frac{\partial(\frac{1}{i})}{\partial(i)}}{\frac{\partial(\frac{1}{i})}{\partial(i)}} \\
&= \frac{1}{P} \lim_{i\to\infty} \frac{\partial(1-(1-P)^{\frac{1}{i}})}{\partial\left(\frac{1}{i}\right)} \\
&= \frac{1}{P} \lim_{i\to\infty} \left(-(1-P)^{\frac{1}{i}} \times ln(1-P)\right) \qquad \text{(by } \tfrac{\partial\alpha^x}{\partial x} = \alpha^x ln\alpha\text{)} \\
&= \frac{-ln(1-P)}{P} \lim_{i\to\infty} \left(-(1-P)^{\frac{1}{i}}\right) \\
&= -\frac{ln(1-P)}{P}
\end{aligned}
$$

Hence we have Equation 9:

$$OR = -\frac{ln(1-P)}{P} \tag{9}$$

Note that although both OR and P are dependent on i, k, and n, what Equation 9 shows is that when i is large, no matter what values i, k , and n take, OR solely depends on P.

Another view to understand Equation 9 is that in the sampling process there are various ways, either by increasing i or k, to reach a certain value of P. When i is large, Equation 9 states that no matter how P is obtained, OR is a function of P only.

Equation 9 can be also illustrated by drawing Equation 8 as in Figure 2, where OR is a function of both P and i. It shows that while the relation between P and OR is volatile when i is small, it tends to be stable with the increase of i. We can see that the threshold value for i is dependent on P. Although in the above derivation i in theory should be a large number, our empirical experiments showed that 100 is good enough.

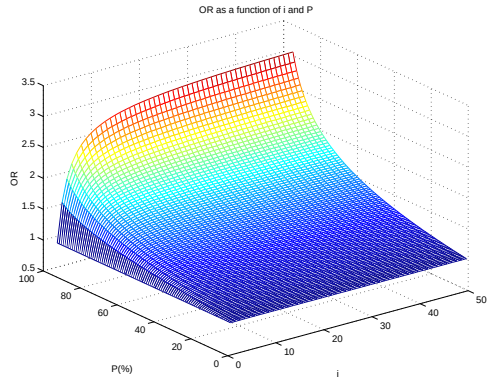

Fig. 2 Plot for Equation 8 with i as X axis, P as Y axis, and OR as Z axis. With the increase of i, the relation between P and OR becomes stable.

To estimate the value of P based on OR, we need to inverse Equation 9. Since P is smaller then 1, we can postulate the regression equation as follows:

$$P = 1 - \alpha OR^{\beta}$$
$$ln(1 - P) = ln\alpha + \beta ln(OR)$$

By running linear regression on the above equation with data generated from Equation 9 when P is smaller than 0.5, we have

$$\hat{ln(1 - P)} = 0.005448767 - 2.108591182 ln(OR),$$
$$R^2 = 0.999791835$$

i.e., $\hat{\alpha}$=1.005463639, and $\hat{\beta}$=2.108591182.

Here R-square is very close to one, and the standard errors for the intercept and slope are close to zeroes (0.000748673 and 0.00453560 respectively). Thus, we derive the following approximation for the relation between P and OR when P is smaller than 0.5:

$$P = 1 - OR^{-2.1} \tag{5}$$

One of the reasons to restrict P to be smaller than 0.5 is that in estimation applications, the samples in general do not need to be very large. Another reason is that when P is small, Equation 9 and 5 can fit very well as illustrated in Figure 3. However, when P (and OR) becomes larger, there is a small discrepancy between these two equations as shown in Figure 3 (A).

3.3 Varying k

In the derivation of Equation 5 we assume that all the captures have a fixed size k. We need to show that Equation 5 also holds for varying k size as long as $k \ll n$. This will make the equation more applicable in practice, since in many cases each capture may return varying number of elements.

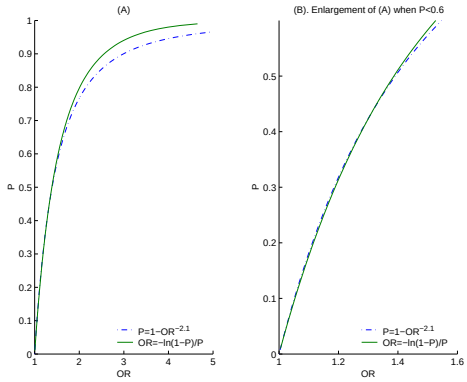

Fig. 3 Plots for Equations 5 and 9. Sub figure (B) is an enlargement of (A) when OR is small.

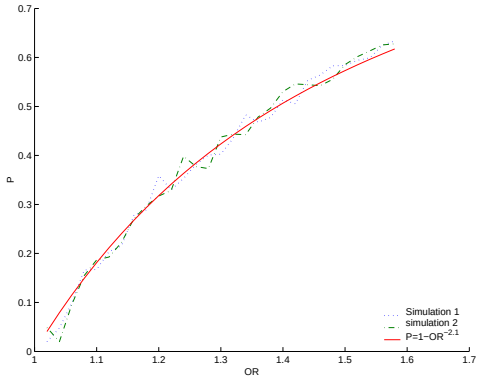

Fig. 4 Simulation with random step length and the comparison with Equation 5. Two simulations (Simulation 1 and Simulation 2) are run on the same condition, i.e., n=5000, $stepSize$ is a random number between 1 to 100.

We conducted another simulation which is the same as the previous one described in Section 3.1 except that now in each capture the number of elements is a random number, i.e., in each step we generate a random number $stepSize$ within range 1 and k, then we generate $stepSize$ number of random numbers within [1, n]. The relationship between P and OR is similar to the study in Section 3.1. In the next section more simulations on varying step lengths are conducted to confirm this result. Figure 4 shows the results of two runs of the simulation with exactly the same condition, and the comparison with Equation 5.

3.4 A naive estimator

Based on Equation 5, a native estimator is given below:

$$\hat{n} = \frac{u}{1 - OR^{-2.1}} \tag{10}$$

where u is the number of distinct documents retrieved.

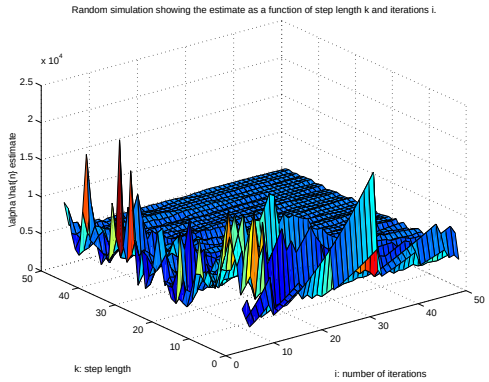

Fig. 5 Estimation result using Equation 10 with different but fixed step lengths and iterations. n=5,000. The estimate becomes stable and accurate when k and i increase.

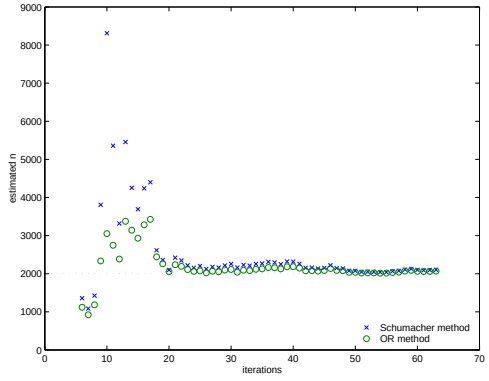

Fig. 6 Estimation result using our OR method (i.e. Equation 10) and Schumacher and Eschmeyer method. n=2,000. In each iteration k elements are selected, where k is a random number between 1 and 40.

Although the estimator works only for random samples from a uniform distribution, which is rare in practical applications, it leads to the derivation of the estimator for unequal catch probabilities as described in the next section.

Figure 5 shows the results of the estimation using Equation 10, where the elements are selected randomly with uniform distribution in the range of [1, 5000]. Here the actual size n is 5000. When both i and k are small ($i < 20, k < 20$), many of the estimates are infinite and are not shown in the chart. This can be explained by the Birthday Paradox that the sample size should be larger than $\sqrt{n}$ to produce collisions, or make overlapping rates greater than 1. With the increase of k and i, the estimate becomes stable and accurate.

3.5 Comparison with Schumacher and Eschmeyer method

We also compared our method (OR method) described in Equation 10 with the traditional Schumacher and Eschmeyer method [35] described in Equation 4. Table 1

Table 1 Details of the simulation and the stepwise estimation result by SE(Schumacher & Eschmeyer) method and OR method. n=2000, step size ranges between 1 and 40.

i	k(i)	d(i)	u(i)	$\hat{n}_{SE}$	$\hat{n}_{OR}$
1	3	0	3	∞	∞
2	6	0	9	∞	∞
3	37	0	46	∞	∞
4	23	2	67	1,359	1,119
5	8	1	74	1,086	924
6	10	0	84	1,425	1,182
7	35	0	119	3,808	2,336
8	38	0	157	8,311	3,051
9	10	1	166	5,359	2,748
10	31	3	194	3,317	2,384
11	38	0	232	5,456	3,376
12	32	3	261	4,253	3,143
13	14	2	273	3,693	2,933
14	28	1	300	4,240	3,285
15	18	1	317	4,400	3,426
16	32	10	339	2,614	2,442
17	27	6	360	2,360	2,261
18	34	9	385	2,097	2,054
19	39	3	421	2,420	2,235
20	32	7	446	2,347	2,193
21	24	7	463	2,221	2,109
22	17	5	475	2,155	2,064
23	39	8	506	2,204	2,076
24	21	7	520	2,125	2,024
25	6	0	526	2,181	2,066
26	6	2	530	2,160	2,053
27	18	3	545	2,217	2,097
28	28	6	567	2,258	2,112
29	38	13	592	2,164	2,039
30	19	3	608	2,232	2,092
...					

explains the details of one simulation where n=2000. $k(i)$ is the number of elements captured in the i-th step, $d(i)$ is the duplicate elements in the i-step, $u(i)$ is the unique elements captured up to the i-th step. Figure 6 is the plot of the data in Table 1.

Table 2 summarizes the MSE(Mean Squared Error), variance, and bias for various sample sizes ranging between 20% and 50% of n. For each sample size 50 simulations are done and the MSE et al are calculated from the 50 simulations. It can be seen that our OR method is systematically better than Schumacher and Eschmeyer method.

Table 2 Comparison of SE (Schumacher and Eschmeyer) method and OR method. The actual size is 2,000, and the step size is a random number between 1 and 40. The data is obtained by 50 runs.

Sample size (%)	$\hat{n}$		MSE		Var		Bias	
	SE	OR	SE	OR	SE	OR	SE	OR
20	2239	2000	138,909	61,335	81,582	61,334	239.43	0.87
30	2105	1959	48,645	28,579	37,612	26,975	105.04	-40.04
40	2070	1990	18,323	10,507	13,286	10,415	70.97	-9.64
50	2052	2017	11,789	7,106	9,031	6,782	52.52	17.99

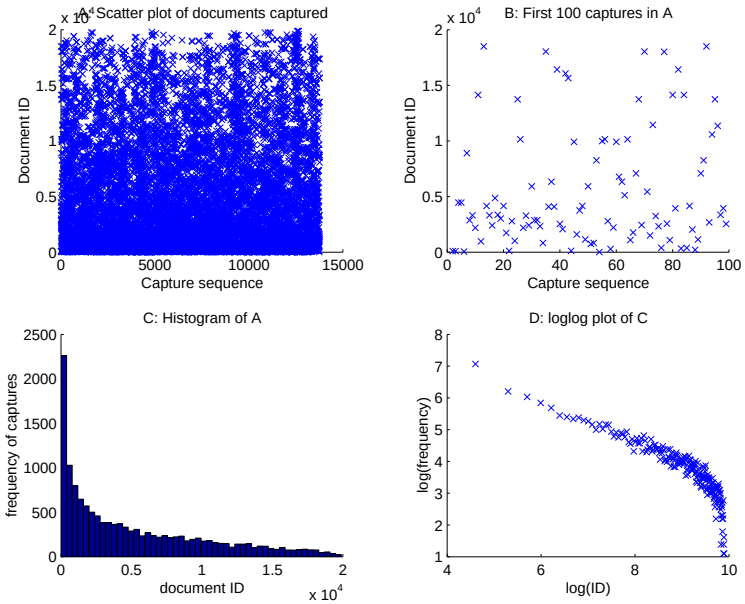

Fig. 7 Capture frequency of news groups documents by queries: (A) is the scatter plot when documents are selected by queries. In total 13,600 documents are retrieved. (B) is the first 100 captures in figure (A). (C) is the histogram of (A). (D) is the log-log plot of (C).

4 ESTIMATOR WHEN DOCUMENTS ARE SELECTED BY QUERIES

Equation 10 is derived on the assumption that documents are selected randomly from uniform distribution. Although we can select random queries, picking out a set of random documents is by no means an easy task [3][6][37]. When samples are not randomly selected with a uniform distribution, there is a tendency that the estimation is negatively biased in traditional capture-recapture method [1] and in data size estimation [34][4][6]. Almost all the methods underestimate the size consistently.

Prompted by Equation 5, we conjecture that there is also a fixed relation between P and OR in real data sources, with a modified equation:

$$P = 1 - OR^{\alpha}, \tag{11}$$

where α is a value, between 0 and -2.1, that is to be determined by the corpus under investigation. The previous section has shown that if the documents can be sampled with a uniform distribution then $\alpha = -2.1$. Before describing the method to calculate α for documents obtained by queries, we need to understand the non-uniform sampling first.

4.1 The unequal catching probability

To reveal the cause of the negative bias, we conducted an experiment to show how documents are captured when random queries are issued. In this experiment 20,000

documents in 20 Newsgroups [24] are sorted according to their file sizes, and are numbered through 1 to 20,000 in a decreasing order. Then we retrieve documents by firing single word queries that are randomly selected from the Webster dictionary. In Figure 7, (A) depicts the 13,600 documents that are selected by queries, and includes duplicates. To give a better view of the (A), (B) depicts the first 100 documents that are captured.

To understand the plot better, we draw the histogram of retrieved documents as in sub-figure (C), where the bin size is 100. It shows that the largest one percent of the documents (i.e., the largest 200 documents) are captured around 2,300 times. The second largest 200 documents are retrieved around 1,000 times. Many small documents, i.e., the documents with larger IDs, are rarely retrieved by queries. The histogram demonstrates that the capture frequency follows the power law, as its log-log plot in Figure 7 (D) shows.

Since the capture frequency follows the power law, in the following simulation studies, we will generate random numbers following Pareto distribution to simulate the capture frequency.

4.2 Measure the heterogeneity

One way to measure the degree of heterogeneity of the capture probability distribution is coefficient of variation (CV) [12]. Assume that the documents in the data source have different but fixed probabilities of being captured, i.e., $p = \{p_1, p_2, \ldots, p_n\}$, $\sum_{j=1}^{n}(p_j) = 1$. Let $p_1, p_2, \ldots,$ and p_n have mean $\bar{p} = \sum p_i/n = 1/n$. The coefficient of variation of the probabilities p is defined as

$$\gamma = \frac{\sqrt{\sum_{i=1}^{n}(p_i - \bar{p})^2/n}}{\bar{p}}$$

Figure 8 shows how documents are captured when CV varies, where x-axis is the capture sequence, and y-axis is the document ID. If a marker appears at the position (10, 1000), it means that the document labeled 1000 is retrieved in the 10-th capture occasion. Here we suppose that in each capture we will retrieve only one document. Figure 8 (A) has a small CV, hence almost all the documents are retrieved with similar probability. With the increase of CV, certain documents, most probably large ones, are retrieved more frequently, hence resulting in higher overlapping between the query results. We can see that when CV=2.6, among the 50% of the small documents with document IDs greater than 10,000, only a few of them are retrieved in 200 captures. If we continue to use the estimator in Equation 10, a higher CV will induce a larger negative bias. Hence we need to adjust α in Equation 11 according to the value of γ.

4.3 Estimate α using CV

To reveal the relationship between α and γ, we conducted a simulation study that produces a set of pairs (γ, α). We first generate random numbers that follow Pareto distributions with different exponents and cutoff values. Each distribution will have a different value for γ.

For each distribution, i.e., each γ, we randomly select some elements from the population, and keep record of the values for P and OR. After certain amount of

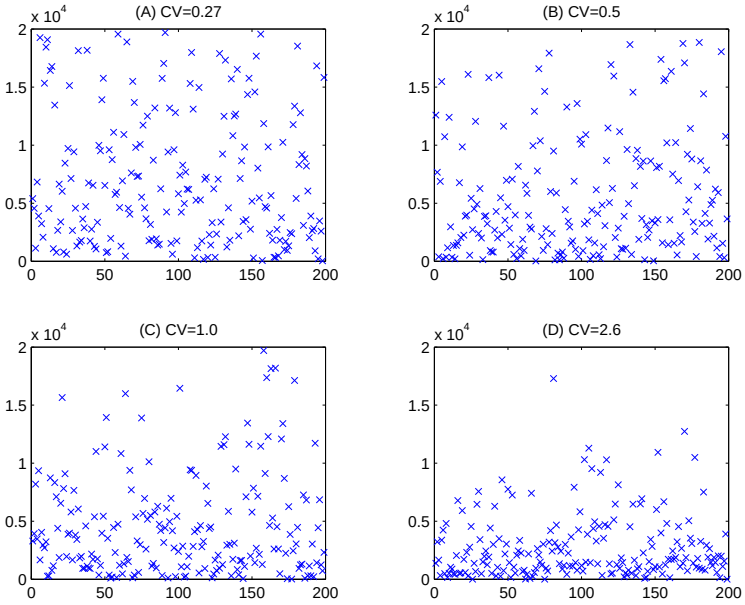

Fig. 8 Scatter plots for various CVs. 200 random numbers within the range of 1 and 20,000 are generated in Pareto distribution.

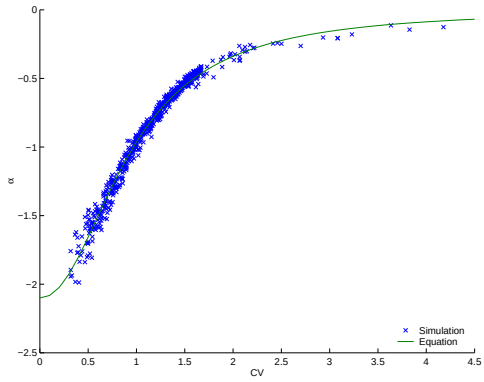

Fig. 9 Relationship between CV (γ) and α.

elements are selected from the population, we obtained $\alpha = ln(1-P)/ln(OR)$ according to Equation 11.

Figure 9 shows the relationship between α and γ. From the figure it is obvious that α is dependent on γ.

Next we need to find a formula to describe the relation. Note that α ranges between 0 and -2.1. When $\gamma = 0$, it is a uniform distribution, hence $\alpha = -2.1$. When γ increases, α will become closer to zero. Given those observations, we postulate that the following relation between α and γ holds, where a and b are to be determined.

$$\alpha = \frac{-2.1}{1 + a\gamma^b}$$

To find the values for a and b, we run linear regression again on

$$ln(-2.1/\alpha - 1) = lna + bln\gamma.$$

with the values for α and γ that are obtained in the simulation. The regression produces

$$ln(-2.1/\alpha - 1) = 0.1644 + 2.1408ln\gamma, R^2 = 0.9743$$

i.e., $\hat{a}$= 1.1786, $\hat{b}$=2.1408.

Thus we derived the equation

$$\alpha = \frac{-2.1}{1 + 1.1786\gamma^{2.1408}}$$

Hence,

$$P = 1 - OR^{-2.1/(1+1.1786\gamma^{2.1408})} \tag{12}$$

and

$$\hat{n} = u\left(1 - OR^{-2.1/(1+1.1786\gamma^{2.1408})}\right)^{-1} \tag{13}$$

4.4 Simulation study

To evaluate the behavior of the estimator described in Equation 13, we carried out a simulation study that includes 30 combinations of the following

- Six different CVs, i.e., 6.04, 3.83, 2.86, 2.32, 1.24, and 0.66.
- Five different sample sizes in terms of the fraction of data source size n, i.e., 0.05, 0.25, 0.45, 0.65, and 0.85.

For each combination, we run 100 trials, and record in Table 3 the mean estimation of the data size $\hat{n}$ and its standard deviation. In each trial, we randomly generate numbers in the Pareto distribution within the range 1 to n, where $n = 100,000$. Other values for n are also tested and similar results are obtained . We adjust the exponent of the Pareto distribution and cutoff value so that various CVs are produced. For each CV and sample size we run 100 trials and obtained the estimation $\hat{n}$. Table 3 gives the simulation result.

From the table we can see that both the bias and standard deviation are rather small for all of those combinations, although in general smaller CVs demonstrate a smaller bias. In particular, small sample size such as 5% of n can also produce a rather accurate estimation, although the standard deviation is larger than that of the bigger sample sizes.

4.5 Estimate CV

One estimator for CV based on capture history is given by Chao et al [12]:

$$\hat{\gamma}^2 = \hat{n_1}\frac{\sum_{i=1}^{n} i(i-1)f_i}{t(t-1)} - 1, \tag{14}$$

where $\hat{n}_1$ is the initial estimation for the data source size, f_i is the number of documents that are captured exactly i times, and t is the total occurrences of captured documents. Given that α is a value ranging between 0 and -2.1, we set the initial estimation for n_1 as:

$$\hat{n}_1 = u/(1 - OR^{-1.1}). \tag{15}$$

Table 3 Simulation study for Equation 13 with various CV and sample size. Mean $\hat{n}$ and standard deviation (SD) are based on 100 runs. n=100,000.

		Sample size(/n)				
CV^2		0.05	0.25	0.45	0.65	0.85
6.04	mean $\hat{n}(\times 10^5)$	1.0481	0.9935	1.0086	1.0367	1.0613
	$SD(\times 10^4)$	1.1102	0.9952	1.0073	1.0340	1.0582
3.83	mean $\hat{n}(\times 10^5)$	1.0669	1.0220	1.0198	1.0327	1.0509
	$SD(\times 10^4)$	1.1355	1.0243	1.0180	1.0300	1.0471
2.86	mean $\hat{n}(\times 10^5)$	1.0747	1.0284	1.0234	1.0300	1.0423
	$SD(\times 10^4)$	1.1958	1.0317	1.0227	1.0272	1.0389
2.32	mean $\hat{n}(\times 10^5)$	1.0589	1.0262	1.0231	1.0275	1.0370
	$SD(\times 10^4)$	1.1809	1.0267	1.0218	1.0252	1.0337
1.24	mean $\hat{n}(\times 10^5)$	1.0447	1.0205	1.0188	1.0203	1.0243
	$SD(\times 10^4)$	1.2247	1.0248	1.0177	1.0169	1.0206
0.66	mean $\hat{n}(\times 10^5)$	1.0116	1.0068	1.0080	1.0140	1.0176
	$SD(\times 10^4)$	1.2388	1.0159	1.0068	1.0113	1.0143

5 EXPERIMENTS

This section evaluates our estimator in Equation 13 using standard corpora. Experiments are implemented using Lucene [18] on local data. We did not estimate the size of the real deep web data sources because the total number of documents in a real web data source is usually unknown or inaccurate, hence it is impossible to evaluate the estimation methods. An online demo of our estimation method is also available at http://cs.uwindsor.ca/∼jlu/estimate.

5.1 Data

We run our experiments on a variety of data collected from various domains. They are of different sizes ranging between 30 thousands to 1.4 millions. The corpora are Reuters, Gov2, Wikipedia, and Newsgroups, which are summarized in Table 4. These are standard test data used by many researchers in information retrieval. The file sizes of the corpora roughly follow the power law distribution. The vast majority of the files have very small sizes. On the other hand, there are also very large files, albeit the number of such files is small. Figure 10 shows the log-log plots of the occurrences against the file sizes of the documents in the corpora.

- *Wikipedia* is the corpus provided by wikipedia.org which contains 1.4 millions of English documents.
- *Gov*2 is a TREC test data collected from .gov domain during 2004, which contains 25 million documents. We used two subsets of the data for efficiency consideration.
- *Newsgroups* corpus includes posts in various newsgroups. We used two subsets of the corpus, named NG_1 and NG_2, respectively.
- *Reuters* is a TREC data set that contains 806,790 news stories in English (http://trec.nist.gov/data/reuters/reuters.html).

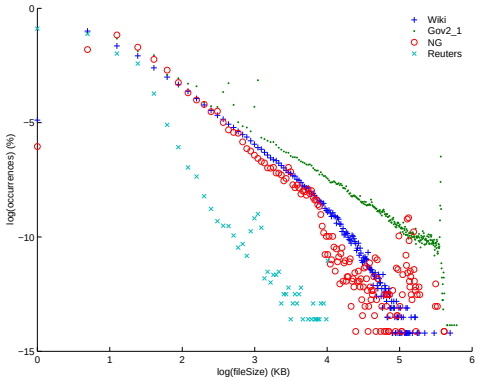

Fig. 10 File size distributions of the four corpora.

Table 4 Summary of test corpora. 'Number of words' is the number of unique words that appear in both the documents and the Webster dictionary.

Corpus name	docs	Size in MB	Avg size(KB)	mean #words	SD of #words
Wiki	1,475,022	1950	1.35	284	285
Gov2_1	1,077,019	5420	5.15	396	409
Gov2_2	1,075,270	5241	4.99	389	407
NG_1	1,372,911	1023	0.76	294	223
NG_2	30,000	22	0.73	290	180
Reuters	806,791	666	0.85	125	82

5.2 Evaluation metrics

The method is evaluated in terms of relative bias, i.e., how far away the estimation is from the actual value, and relative standard deviation, i.e., how close those estimations are with each other.

Let n denote the actual size of the collection, $\hat{n}$ the estimation. Suppose that there are m number of estimations. The expectation of $\hat{n}$, denoted as $E(\hat{n})$, represents the mean of m estimations, i.e., $E(\hat{n}) = (\hat{n_1} + \hat{n_2} + ... + \hat{n_m})/m$. Relative bias (RB) $E(\hat{n}) - n)/n$ is to measure how close the estimations are to the actual size of the data source. Relative standard deviation (RSD) characterizes the variations of estimations, i.e., how close the estimations are to the center of the estimations. It is defined as

$$RSD = \frac{1}{E(\hat{n})}\sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{n_i} - E(\hat{n}))^2}$$

5.3 Experimental setup

In the experiments, each query is a single word randomly selected from the Webster dictionary, which consists of 60,000 entries. For each query we will count all the matched documents. When a popular word, e.g., a stop word, is selected, it may match with most of the documents and make the overlapping rate extremely low, especially when

the number of queries are small. Hence, we remove the top two percent of the most popular words from the query pool.

Our first experiment recorded in Table 5 studies 30 combinations of six data collections and five sample sizes in terms of query numbers that include 50, 100, 200, 400, and 800.

For each combination 100 trials are run, each with randomly selected queries from the Webster dictionary. For each set of 100 estimates generated, bias and standard deviation are computed. To have a clear comparison among corpora of different sizes, we record the bias and standard deviation relative to the data collection size.

Table 5 shows that smaller sample sizes (when the number of queries is 50) cause significant overestimation. This is in accordance with our simulation study as illustrated in Figures 5 and 6. When sample size is too small, some queries may not have overlapping matches at all, hence resulting in overestimation. Also, the variance is large in both simulation studies and real data experiments. However, when the number of queries is equal or greater than 100, the estimation results are in general rather accurate. In particular, when the sample size increases, the standard deviation becomes smaller as expected.

In the file size distributions of the corpora experimented, some corpora have heavier tails than others, as shown in Figure 10. For example, Gov2 has a flatter slope and heavier tail than Reuters, indicating that Gov2 has more large files than Reuters. A greater portion of large files will lead to higher degree of heterogeneity for capture probability, because those large files will be sampled repeatedly while the small files, albeit they are the majority of the corpus, are seldom sampled. Thus flatter slope will cause lower values of $|\hat{\alpha}|$, which is confirmed by our experiment. In Table 5, Gov2 consistently has a smaller value of $|\hat{\alpha}|$ than that of Reuters.

5.3.1 Impact of sample size

Our second experiment reported in Table 6 focuses on Reuters corpus with more sample sizes. Again, each data item is obtained from 100 trials, each with a different set of words randomly selected from the Webster. In addition to bias and standard deviation, we also record the estimated values for γ and α. We can see that more queries will produce better results, and converge to the actual n.

5.3.2 Impact of vocabulary

Previous experiments use the words selected from the Webster dictionary. Our next experiment tests the impact of vocabulary. We build three other vocabularies that consist of the words from very different domains:

- newsgroups: about 20k newsgroups postings;
- wsdl: about 2k web service description files;
- ohsumed: about 50k medical documents.

Table 7 shows that four different vocabularies have little influence on the estimation result. In addition to the data reported here, we also experimented with other vocabularies from different areas, and observed similar behaviors.

Table 5 Bias and standard deviation of the estimation over 100 runs. In each run queries are randomly selected from the Webster dictionary. Sample size is the fraction of n.

		Number of queries				
Corpus		50	100	200	400	800
Wiki	RB	-0.09	-0.15	-0.11	-0.02	0.11
	RSD	0.28	0.14	0.12	0.08	0.05
	Sample Size	0.07	0.11	0.18	0.31	0.47
	mean $\hat{\alpha}$	-1.08	-1.05	-0.99	-0.91	-0.83
Gov2_1	RB	0.09	-0.18	-0.22	-0.13	0.05
	RSD	0.99	0.48	0.28	0.21	0.18
	Sample Size	0.07	0.12	0.19	0.31	0.49
	mean $\hat{\alpha}$	-1.09	-1.06	-1.00	-0.90	-0.81
Gov2_2	RB	0.19	-0.12	-0.17	-0.09	0.03
	RSD	1.33	0.50	0.29	0.24	0.17
	Sample Size	0.09	0.13	0.21	0.33	0.49
	mean $\hat{\alpha}$	-1.10	-1.06	-0.99	-0.89	-0.82
NG_1	RB	0.30	0.08	0.13	0.21	0.28
	RSD	0.64	0.28	0.16	0.10	0.05
	Sample Size	0.09	0.12	0.23	0.39	0.57
	mean $\hat{\alpha}$	-1.09	-1.06	-1.01	-0.95	-0.90
NG_2	RB	0.45	0.14	0.06	0.12	0.21
	RSD	0.78	0.30	0.13	0.10	0.06
	Sample Size	0.06	0.11	0.18	0.30	0.49
	mean $\hat{\alpha}$	-1.1	-1.09	-1.05	-1.00	-0.95
Reuters	RB	0.74	0.03	0.00	-0.04	-0.06
	RSD	3.08	0.27	0.18	0.12	0.06
	Sample Size	0.03	0.04	0.07	0.13	0.23
	mean $\hat{\alpha}$	-1.11	-1.11	-1.10	-1.09	-1.08

Table 6 Impact of sample size on Reuters Corpus. Data are obtained from 100 trials.

queries	mean sample size	mean $\hat{n}$	RB	SD	mean $\hat{\gamma}$	mean $\hat{\alpha}$
50	22,335	1,476,418	0.830	1,757,016	0.81	-1.11
100	37,884	902,351	0.118	301,139	0.81	-1.11
200	67,503	816,023	0.011	170,302	0.82	-1.10
400	119,537	755,362	-0.064	82,819	0.84	-1.09
800	237,623	754,529	-0.065	55,199	0.85	-1.08
1600	478,303	778,159	-0.035	40,524	0.89	-1.06
3200	930,729	796,311	-0.013	25,271	0.93	-1.04

Table 7 Impact of vocabulary on Reuters corpus. 200 query terms are randomly selected from different domains. Data are obtained from 100 trials.

vocabulary	$E(\hat{n})$	RB	SD
webster	803,578	-0.004	156,320
newsgroups	860,436	0.066	46,134
wsdl	931,883	0.155	94,160
ohsumed	870,843	0.079	18,286

5.4 Comparison with other methods

Compared with Broder et al.'s method [9], which is computationally much more expensive, most of our results are better than the best reported 0.22 relative bias. Broder et al.'s method also suffers from large variance.

Table 8 Comparison of OR and CH_{reg} methods. Each data item in the table is obtained from 100 runs.

Corpus		CH_{reg}	OR
Wiki	RB	0.91	-0.15
	RSD	0.03	0.14
Gov2_1	RB	-0.41	-0.18
	RSD	0.01	0.48
Gov2_2	RB	-0.41	-0.12
	RSD	0.01	0.50
NG_1	RB	1.35	0.08
	RSD	0.04	0.28
NG_2	RB	0.08	0.14
	RSD	0.01	0.30
Reuters	RB	0.56	0.03
	RSD	0.03	0.27

For the logistic method described briefly in [40], they tested on data collections with smaller sizes, which in general will result in a smaller bias than larger data collections. Even with those small data sets, the mean absolute error ratio (relative bias) reported in the paper is 0.15, which is higher than our method in most cases.

Sokouhi et al.'s CH_{reg} method is the closest to ours in that their method is also based on the multiple capture-recapture method. In addition, they also use document ids only, hence saving the effort of document downloading and parsing. Table 8 shows the comparison of the two methods. For OR method, 100 queries are issued and all the matched documents are accounted for. More queries will produce more accurate estimations. For CH_{reg} method, the parameters are set as described in the paper [34], i.e., 15000 queries are issued and only the top k documents, where $k = 10$, are selected. We find that both the value of k and the ranking method affect the estimation result greatly. We used the ranking method that produces the best estimation, i.e., the default ranking by Lucence search engine. Larger k will result in larger positive bias.

From Table 8 it can be seen that our OR method has much smaller bias in most of the cases. CH_{reg} works well only when the data source is small, i.e., corpus NG_2 which contains 30,000 documents. The standard deviation of CH_{reg} method is smaller because in each run it uses a large portion of the words (5,000) out of a vocabulary of size 60,000. In the contrary OR method uses only 100 queries.

We also tried Chao's classical estimation method in ecology [12]. As noted in the literature and confirmed by experiments on our data, the method works well only when the sample size is large and CV square is small. For example, when CV is around one, the sample size needs to be $4 \times n$. Obviously this is too costly for data source size estimation.

6 CONCLUSIONS

This paper proposes a novel method to estimate the size of deep web data sources by sending random queries and analyzing returned document ids.

Our first estimator in Equation 10 is meant for uniformly distributed samples. It is analytically derived and empirically verified by simulation. The simulation study shows that it has smaller bias and variation than the classical Schumacher et al.'s estimator[35][14].

Our second estimator in Equation 13 is derived for samples obtained by queries. It is verified by extensive simulation studies with various degrees of heterogeneity of catching probability and different sample sizes. In applications on various data collections, the estimator also demonstrates a small bias and variance compared with other estimation methods for data collection size.

In addition to the accuracy of the estimation, one salient feature of our method is its efficiency. We only need to issue random queries, instead of carefully selected ones. In most of the cases, only 100 queries are needed, although more queries will result in higher accuracy.

One limitation of our method is the assumption that all the matched document IDs can be retrieved. Very large data sources, such as general-purpose search engines, usually impose a limit on the maximal number of matched results that can be returned for practical consideration. When a query matches more documents than the limit k, i.e., when the query overflows, the matched documents are ranked and only the top-k are returned. This strategy will cause the rank bias. While our method solves the query bias quite well, the rank bias is not tackled in this paper. Nonetheless, our method has a wide area of applications such as 1) data sources that do not impose a return limit; 2) data sources that are not very large, hence most queries will match less than k number of documents; and 3) data sources that do not rank the returning results.

Our study shows that when there are not many overflowing queries, our method still works well by ignoring the overflowing queries. When the density of overflowing queries is high, filtering out those queries will induce another bias, because rare words usually introduce higher overlapping rate. Hence our method is only applicable to ranked data sources with low overflowing density. From this perspective, our method is suitable for relatively small text databases, depending on the result limit. To estimate the size of large search engines such as Google, throwing away overflowing queries is no longer an option since most of the random queries will match more pages than the 1000 limit Google sets.

On the other hand, with the proliferation of web services and programmable web interfaces, which typically contain data sources of relatively small size, there is a growing demand for probing those data sources to learn their profiles, such as their size. We also expect that the method can be applied in other areas where ranking bias does not exist, such as the estimation of the number of values of an attribute in a database relation [17].

It is not easy to obtain the detailed performance comparison with the methods based on document analysis [9][3][4][6]. All those works target very large data sources such as the size of the major search engines and the entire web, while we provide a convenient estimator for relatively small data sources. Compared with this group of work, the main advantage of our method is its efficiency-it does not need to download and analyze the documents; does not need to carefully select the uncorrelated queries; and does not need to send out many queries.

Future work on document id based estimators may focus on correcting the rank bias so that our method can be applied to very large data sources. In addition, we need to investigate biases caused by other sources, such as duplicate documents and documents that do not contain English words.

7 Acknowledgments

The research is supported by NSERC (Natural Sciences and Engineering Research Council Canada) and SSHRC (Social Sciences and Humanities Research Council Canada). We would like to Thank Jie Liang for providing the query interface for the corpora.

References

1. SC Amstrup, TL McDonald, BFJ Manly, Handbook of Capture-Recapture Analysis, Princeton University Press, 2005.
2. L.Barbosa and J. Freire, Siphoning hidden-web data through keyword-based interfaces, SBBD, 2004.
3. Bar-Yossef, Z. and Gurevich, M. 2006. Random sampling from a search engine's index. WWW , 2006. 367-376.
4. Bar-Yossef, Z. and Gurevich, M. Efficient search engine measurements. WWW , 2007. 401-410.
5. Michael K. Bergman, The Deep Web: Surfacing Hidden Value, The Journal of Electronic Publishing 7 (1). 2001.
6. Bharat, K. and Broder, A. A technique for measuring the relative size and overlap of public Web search engines. WWW 1998, 379-388.
7. Igor A. Bolshakov, Sofia N. Galicia-Haro: Can We Correctly Estimate the Total Number of Pages in Google for a Specific Language? CICLing 2003: 415-419.
8. Paul Bourret: How to Estimate the Sizes of Domains. Inf. Process. Lett. 19(5): 237-243 (1984)
9. Broder, A., Fontura, M., Josifovski, V., Kumar, R., Motwani, R., Nabar, S., Panigrahy, R., Tomkins, A., and Xu, Y. 2006. Estimating corpus size via queries. CIKM '06. 594-603.
10. Callan,J. and M. Connell, Query-Based Sampling of Text Databases. ACM Transactions on Information Systems, 2001. 97-130.
11. James Caverlee, Ling Liu, and David Buttler, Probe, Cluster, and Discover: Focused Extraction of QA-Pagelets from the Deep Web, ICDE 2004. 103-114.
12. Chao, A. and Lee, S-M. (1992). Estimating the number of classes via sample coverage. Journal of American Statistical Association, 87, 210-217.
13. V. Crescenzi, G. Mecca, P. Merialdo, RoadRunner: towards automatic data extraction from large web sites, VLDB J. 2001. 109-118.
14. Darroch, J. N., The Multiple-recapture Census: I . Estimation of a Closed Population, Biometrika, Vol. 45, No. 3/4 (Dec., 1958), pp. 343-359.
15. A. Dobra and S. Fienberg. How large is the World Wide Web? Web Dynamics, 23-44, 2004.
16. Antonio Gulli, A. Signorini: The indexable web is more than 11.5 billion pages. WWW 2005. 902-903.
17. Peter J. Haas, Jeffrey F. Naughton, S. Seshadri, Lynne Stokes: Sampling-Based Estimation of the Number of Distinct Values of an Attribute. VLDB 1995: 311-322.
18. Hatcher, E. and O. Gospodnetic, Lucene in Action, Manning Publications, 2004.
19. H. S. Heaps. Information Retrieval - Computational and Theoretical Aspects. Academic Press, 1978.
20. Hersh, W., Buckley, C., Leone, T. J., and Hickam, D, OHSUMED: an interactive retrieval evaluation and new large test collection for research, SIGIR 1994. 192-201.
21. Lars Holst, A Unified Approach to Limit Theorems for Urn Models, Journal of Applied Probability, 16(1). 1979. 154-162.
22. Ipeirotis, P. G., Gravano, L., and Sahami, M. 2001. Probe, count, and classify: categorizing hidden web databases. SIGMOD '01.
23. C.A. Knoblock, K. Lerman, S. Minton, I. Muslea, Accurately and reliably extracting data from the web: a machine learning approach, IEEE Data Eng. Bull. 23 (4) (2000). 33-41.
24. Ken Lang, Newsweeder: Learning to filter netnews, Twelfth International Conference on Machine Learning, 1995, 331-339.
25. Stephen W. Liddle, David W. Embley, Del T. Scott, Sai Ho Yau, Extracting Data behind Web Forms, Advanced Conceptual Modeling Techniques, 2002. 402-413.

26. Liu, K., Yu, C., and Meng, W. Discovering the representative of a search engine. CIKM '02. 652-654.
27. Jianguo Lu, Efficient estimation of the size of text deep web data source. CIKM 2008: 1485-1486.
28. Jianguo Lu, Yan Wang, Jie Liang, Jessica Chen, Jiming Liu: An Approach to Deep Web Crawling by Sampling. Web Intelligence 2008: 718-724.
29. Nelson, M. L., Smith, J. A., and del Campo, I. G. Efficient, automatic web resource harvesting. WIDM '06, 43-50.
30. Alexandros Ntoulas, Petros Zerfos, and Junghoo Cho, Downloading Textual Hidden Web Content through Keyword Queries. JCDL, 2005. 100-109.
31. Pollock, K. H., J. D. Nichols, C. Brownie, and J. E. Hines. 1990. Statistical inference for captureCrecapture experiments. Wildlife Monographs 107.
32. S. Raghavan, H. Garcia-Molina. Crawling the Hidden Web, VLDB 2001.
33. Denis Shestakov, Sourav S. Bhowmick and Ee-Peng Lim, DEQUE: querying the deep web, Journal of Data Knowl. Eng., 52(3), 2005. 273-311.
34. M Shokouhi, J Zobel, F Scholer, SMM Tahaghoghi, Capturing collection size for distributed non-cooperative retrieval, SIGIR '06. 316-323.
35. Schumacher, F. X. Eschmeyer, R. W. (1943). The estimation of fish populations in lakes or ponds. J. Tenn. Acad. Sci.18, 228-249.
36. Si, L. and Callan, J. 2003. Relevant document distribution estimation method for resource selection. SIGIR '03.
37. Paul Thomas, David Hawking, Evaluating Sampling Methods for Uncooperative Collections, SIGIR, 2007.
38. S. Wu, F. Gibb, and F. Crestani. Experiments with document archive size detection. 25th European Conference on IR Research, 2003. 294–304.
39. Ping Wu, Ji-Rong Wen, Huan Liu, Wei-Ying Ma, Query Selection Techniques for Efficient Crawling of Structured Web Sources, ICDE, 2006. 47-56.
40. Xu, J., Wu, S., and Li, X. 2007. Estimating collection size with logistic regression. SIGIR'07, 789-790.