

SCUBA: Source Selection for Deep Web Integration

Raju Balakrishnan
rajub@asu.edu

Subbarao Kambhampati
rao@asu.edu

School of Computing and Informatics
Arizona State University
Tempe AZ 85287

ABSTRACT

One immediate challenge in exploiting the deep web sources is *source selection*—i.e. selecting the most relevant sources for answering a given query. Existing work on source selection work with the implicit assumption that all sources provide results of similar quality. This assumption does not hold in uncontrolled source collections like deep web where sources vary widely in terms of the quality of the data they provide. To account for this variability, we present a robust measure for assessing the trustworthiness of a source called *SourceRank*. SourceRank assesses trustworthiness of deep web sources through an aggregate computation on the degree of agreement between the answer sets provided by the sources. Agreement between the answer sets is modeled as an agreement graph, and trustworthiness of a data source is measured as the stationary visit probability of a random walk on this graph. Since the source selection in deep web has to deal with huge number of sources, computational complexity is an important concern. Analyzing computational complexity, we show that basic source selection is NP-Hard. As a corollary of our proof of NP-hardness we show that a constant approximation algorithm for source selection is hard. Our experiments on real data bases show that SourceRank significantly improve the traditional “coverage/overlap” based source selection.

1. INTRODUCTION

By many accounts, the directly crawlable surface web contains only a fraction of the overall information available on the web. The remaining is hidden behind a welter of web-accessible databases. By some estimates, this collection, called deep web, is estimated to contain nearly 25 million data sources [30, 37, 5]. This realization is exciting for the database community as database-style query processing techniques can be brought to bear on the structured sources of the deep web. The most promising approach that has emerged for searching and exploiting the sources on the deep web is data integration. A critical advantage of integration to surface web search is that the integration system (mediator) can leverage the semantics implied in the structure of deep web tuples. Realizing this approach however poses several fundamental challenges, the most immediate of which is that of *source selection*. Briefly, given a query, the source selection problem involves selecting the most relevant subset of sources for answering the query.

Although source selection in data integration has received some previous attention (c.f. [31, 8]), most existing approaches are focused on assessing relevance of the sources

based on local measures (involving coverage and latency of the source, and the overlaps between sources). Underlying these approaches is the implicit assumption that all sources provide results of similar quality. This assumption is overly sanguine in the context of web, where many data sources may be inconsistent, poorly curated, or intentionally designed to return data corrupted with commercial intent. The reputation or *trustworthiness* of the data sources is thus of paramount importance. The problem however is that given the sheer number and diversity of the deep web sources, it is hard to estimate their reputation through human help. The main contribution of this paper is a robust approach for estimating the trustworthiness of deep web sources that is derived through an aggregate fixpoint computation on the degree of agreement of between the answer sets returned by the sources. We will show that our trustworthiness measure can be used to significantly improve the traditional “coverage/overlap” based source selection.

Trustworthiness of the the results returned by a source is an indication of relevance of the results. In general, sources try to return truly relevant tuples by assessing the relevance of the tuples by measuring how well the tuple matches the search query using a similarity measure. The results returned by a source may not be truly relevant due to inadequate relevance measure, small data set size, or if the database tries to spam the user. Since the content owners try to boost their search engine ranking, it is unlikely for a source to provide a measure of its own relevance.

A real world example shows that existing database web search engines result quality needs improvement. We issued the query *Godfather* (the all time epic movie trilogy) in Google Video and MSN Live Video Search.¹ Google Video and MSN Video Search returned first page results mixing results from six sources and nine sources respectively. All the first page results had the substring *Godfather* in their names, and clearly are relevant according to Google Video and MSN Video Search. But the top search results by Google Video was a Pepsi commercial, and top result from Live Search was a product launch video of Xbox 360. The first instance referring to Godfather movie for Google Video was the second result, and for MSN Video was the seventh result. To bolster our argument that the Godfather Movie is the true relevant result, the answers to the same query to Google, MSN and Yahoo! surface web search engines had at least all the top four results referring to Godfather movie trilogy.

¹These searchers may be using warehousing, but ranking issues remain the same both for warehousing and integration based search.

Though both the video search engines may be trying to return the diverse search results, returning a unimportant top result is not justifiable by the argument for diversity.²

Measuring the trustworthiness web source is difficult. The measure of trustworthiness of a source should not depend on the information the source provides about itself. But fortunately, the trustworthiness of a particular source is reflected as the endorsement of the source by other sources in general. For example, in surface web the trustworthiness of a page is calculated based on the endorsement as hyper links from other pages, like in PageRank [9] and in authorities and hubs [26]. But the hyper link based endorsement is not directly applicable to the web databases since there are no links between database records.

We present a method to calculate the trustworthiness of a source based on how well the results from the source are agreed by other sources. To clarify the meaning of agreement, two sources agree each other if they return the same tuple in answer to the same query.³ Agreement increases as the fraction of the common results between the sources increases. A source is considered popular if many other sources agree with its results. Intuitively, popular query results are liked by many users and likely to be relevant. Relevant results are likely to be returned by many sources, i.e. agreed upon by many sources. We will provide a more formal argument for why agreement indicates relevance in Section 4. We expand this basic idea, and calculate the trustworthiness of a source as the stationary visit probability of a random walk on the agreement graph of sources.

We examined whether the results agreed upon by the sources indeed are relevant in the *Godfather* example above. The only results agreed upon by two sources in first page results were *Godfather* trailer and *Godfather theme song* in Google Video, and only *Godfather* trailer for MSN Video. Note that both results are truly relevant to the search. No other results, including the two commercials successfully deceived search engines to occupy the two top positions in two Video search engines were not agreed upon by any two sources. This shows that the agreement measure of trustworthiness in fact works in our example.

In addition to the trust consideration detailed above, the cost consideration in traditional sources selection is important for web databases. Sending queries to a source incurs a cost. This cost is sum of costs of resources used for query execution in individual data sources, network bandwidth, source resource utilization, time for merging multiple result lists etc [15, 6, 14, 39]. Hence a web database selection criteria should combine trustworthiness of the source with the cost consideration in current database selection methods.

Current database selection methods maximize number of distinct relevant records from minimum number of sources, to minimize cost [15, 14, 39]. Two parameter widely considered for this minimum cost access are coverage and overlap of sources. Coverage of a database is a measure of number of relevant tuples to the query in the database. Overlap between two databases is a measure of common tuples in databases. The problem of selecting sources is formulated as selecting the least number of databases maximizing cover-

age while keeping the overlap between the selected databases minimal. The coverage and overlap statistics are calculated using samples collected from the databases [31, 24, 27, 18, 23].

Since the number of web databases is estimated in the millions [30], the scalability of database selection is particularly important. Though source selection papers agree that the search space is exponential, and use greedy source selections, a formal hardness proof is not known. Before combining trust and cost based selection criteria we analyze complexity of basic source selection problem. Optimal solution is analyzed. We formally prove that database selection considering coverage and overlap is NP-Hard. A corollary of our proof is that a constant ratio approximation algorithm is hard. Following this, we use a greedy source selection strategy combining source trust and cost.

Rest of this paper is organized as following. Next section discusses the related work. Section 3 introduces notations used for the paper, defines relevance measures and the problem formally. Next section provides background information required for the paper. Section 4 introduces concept of the source reputation based on the database overlap and presents the *SourceRank* algorithm to calculate the source reputations. We combine cost and trust considerations for database selection and explore the problem of optimal source selection in Section 5. Finally in Section 6 we describe experiments and evaluations followed by the conclusions.

2. RELATED WORK

According to some estimates, data contained in the deep web is as large as five hundred times of that in the surface web [5]. Searching this humongous deep web data has been identified as the next big challenge in information management [37, 10].

Existing source selection methods is focussed on two types of sources: databases and text collections. The database selection methods try to retrieve maximum number of tuples from minimum number of sources, minimizing cost [15, 14, 39]. Text collection selections also try to optimize cost, but mostly uses information retrieval metrics and methods [33, 8, 32, 35]. Related problem of collecting statistics for source selection has been researched in detail also [31, 24, 27].

The problems related to deep web integration—schema mapping, query routing, sampling web databases, parsing HTML pages returned by web databases etc have attracted considerable research. Automated schema mapping—mapping the corresponding fields in different web forms—has been solved with good accuracy [21, 40, 28, 38]. Methods for sampling and crawling the deep web sources and relational databases have been discussed in many papers [34, 18, 11]. There are many efforts trying to automated parsing of database tuples from HTML results returned by the web databases [19, 7]. Another related problem to web integration is automatic classification of web sources [17, 25].

While we address the problem of identifying popularity of deep web sources, the corresponding problem in surface web—identifying page popularity—has been solved by seminal papers on PageRank and authorities-hubs [9, 26].

Efficiency issues in information integration and greedy algorithms for source selection has been discussed by many researchers [13, 12, 36].

An alternate approach to search the deep web is to crawl and index deep web data as plain HTML pages [29, 22]. This

²These example searches are last preformed on December 23 2008 with safe search option in default moderate settings.

³Resolving record linkage is an issue in calculating agreement. Our experiments will demonstrate that approximate similarity based record linkage is good enough.

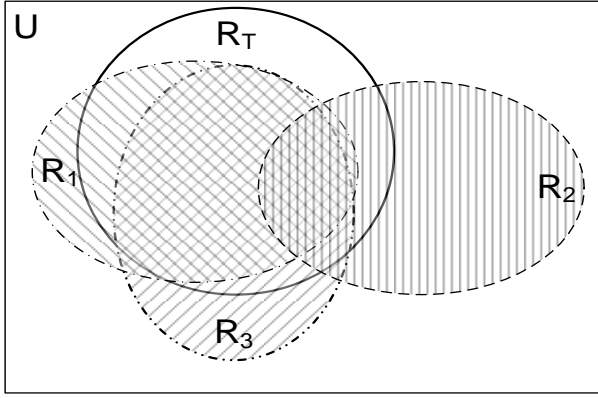


Figure 1: Intuitive explanation of agreement implies trust model. Universal set U is the whole search space, R_T is the set of true relevant results, R_1, R_2 and R_3 are the set of results returned by three independent sources. Since all result sets are independent best effort estimates of set R_T , the result sets agreeing with other result sets are likely to agree more with R_T also. Note that R_1 and R_3 agree with each other more as well as R_T , compared to R_2 . This effect is likely to be more evident as the number of trustworthy sources increases.

approach is technically less challenging, but crawling has the disadvantage of losing the semantics implied by structured database records; in addition to difficulties of crawling entire deep web and maintaining data coherency.

3. ASSUMPTIONS AND DEFINITIONS

We use the probabilistic information retrieval (PIR) model. Probability of relevance of a tuple t given a query q is denoted by $R(t|q)$. We model a database as a set of tuples, and definitions for set of tuples are applicable for databases, as well as answer sets.

Coverage ($C(T|q)$): Coverage of a set of tuples T given a query q is the sum of relevances of tuples in the set.

$$C(T|q) = \sum_{t \in T} R(t|q) \quad (1)$$

Coverage is equal to the expected size of relevant set of tuples. For a set of queries Q , coverage is defined as the mean coverage for the set of queries,

$$C(T|Q) = \frac{\sum_{q \in Q} C(T|q)}{|Q|} \quad (2)$$

Note that the binary relevance model is a special case of probabilistic relevance model with relevance being one or zero. All the definitions and methods in the paper are applicable for binary relevance model also.

Overlap $O(T_1, T_2|q)$: First order overlap between two sets of tuples is T_1 and T_2 is sum of the relevances of the common tuples,

$$O(T_1, T_2|q) = \sum_{t \in T_1 \wedge t \in T_2} R(t|q) \quad (3)$$

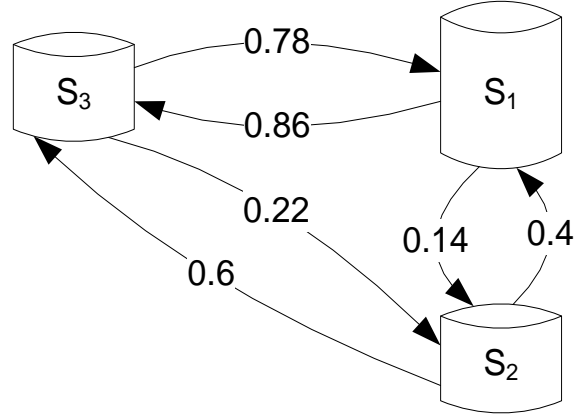


Figure 2: A sample agreement graph structure of three sources. The weight of the edge from S_i to S_j is computed by equation 12. The weights of links from every source S_i is further normalized against sum of the weights of the out links from S_i .

This is equal to the expected number of common relevant tuples.

Similarly we can derive the $(n-1)^{th}$ order overlaps, i.e. overlap between n sets T_1, T_2, \dots, T_n as sum of the relevances of tuples common to all these sets,

$$O(T_1, T_2, \dots, T_n|q) = \sum_{t \in T_1, T_2, \dots, T_n} R(t|q) \quad (4)$$

For a set of queries Q overlap of sets T_1 and T_2 is defined as the mean overlap for the query set i.e.

$$O(T_1, T_2|Q) = \frac{\sum_{q \in Q} O(T_1, T_2|q)}{|Q|} \quad (5)$$

Agreement $A(T_1, T_2)$: Agreement between two sets T_1 and T_2 is the number of common tuples in sets,

$$O(T_1, T_2) = |\{t|t \in T_1 \wedge t \in T_2\}| \quad (6)$$

Note that the only difference between the agreement and overlap is that agreement does not consider the relevance of the tuples.

Source Trust $T(S|Q)$: Source trust is the same as the trustworthiness of the source we described in the introduction. Trust of a source S is denoted by $T(S|Q)$, where Q is the set of queries for which the trustworthiness of the source is calculated. The calculation of trust worthiness is detailed in Section 4.

Utility of Selection $U(DS|q)$: Utility is a measure suitability of a set of data sources DS to execute query q . We define the overall utility of a set of data sources as a combination of source trusts, coverages and overlaps.

Let E_T be the expected number of unique relevant tuples to a query from a set of databases. To compute the exact value of expected number of tuples, we have to consider overlaps of all orders between the data sources. For a set of n databases this would require maintaining $(2^n - n - 1)$ overlap values and clearly infeasible for large number of databases.

As an approximation, we ignore all overlaps of order higher than one. E_T considering only the first order overlaps is,

$$E_T(DS|q) = \sum_{S_i \in DS} C(S_i|q) - \sum_{S_i, S_j \in DS, i < j} O(S_i, S_j|q) \quad (7)$$

We combine the expected number of tuples $E_T(DS)$ with reputation of the source as a simple linear combination. The overall utility $U(DS)$ of selection a set of databases DS is,

$$U(DS|q) = \alpha \sum_{S \in DS} T(S) + (1 - \alpha) E_T(DS|q) \quad (8)$$

where $0 \leq \alpha \leq 1$ is empirically determined.

Problem Definition: Given a query q and a set of data sources $DS = \{S_1, S_2, \dots, S_n\}$ assign q to a set of k source $S_k \subseteq DS$ to maximize the sum of utility of of assignment $U(S_k|q)$.

4. SOURCERANK: TRUST RANKING OF SOURCES

In this section we elaborate the argument that the trustworthiness of a database manifests as agreement of other databases. We devise a method to calculate SourceRank—a trust measure for web databases—based on the agreement between the sources. Calculating SourceRank is a two step process: (i) create a source graph based on agreement between the sources (ii) assess source reputation as the static visit probability distribution of a markov random walk on the source graph. In next subsection we show that the result set agreement is an implicit endorsement. Subsequent subsections describe the process of calculating source rank.

4.1 Agreement as Endorsement

We show in this section why agreement in fact implies endorsement. First let us prove that two independently picked truly relevant tuples are likely to agree each other with much higher probability than two independently picked irrelevant tuples. let $P_A(r_1, r_2)$ denotes the probability of agreement of two independently picked relevant tuples by two sources.

$$P_A(r_1, r_2) = \frac{1}{|R_T|} \quad (9)$$

where R_T is the set of truly relevant tuples.

$P_A(f_1, f_2)$ denotes probability of agreement of two independently picked irrelevant tuples.

$$P_A(f_1, f_2) = \frac{1}{|U - R_T|} \quad (10)$$

where U is the search space (the universal set of all tuples searched). For any search engine, search space is much larger than the set of relevant tuples, i.e. $|U| \gg |R_T|$. Applying this inequality in Equation 9 and 10 directly implies that

$$P_A(r_1, r_2) \gg P_A(f_1, f_2) \quad (11)$$

To provide an understanding of magnitude of these probabilities, let us consider the *Godfather* example in Introduction. Assume that three movies in trilogy The Godfather I, II and III are the results truly relevant to the user. Let us assume that total number of movies searched by all the databases (search space U) is 10^4 . In this case $P_A(t_1, t_2) = \frac{1}{3}$ and $P_A(f_1, f_2) = \frac{1}{10^4}$ (strictly speaking $\frac{1}{10^4 - 3}$). Similarly probability for three tuples picked independently by three

different sources to agree are $\frac{1}{9}$ and $\frac{1}{10^8}$ for relevant and irrelevant results respectively.

Now we extend this argument to show that agreement between the set of sources implies trust worthiness. In Figure 1, R_T represents truly relevant set of results. R_1 , R_2 and R_3 are the results sets returned by three independent sources. U is the entire search space. The results sets of databases are best effort estimates of R_T . Assuming that a good fraction of search engines have a reasonably good relevance assessment, many databases are likely to pick tuples from R_T . Typically the results sets from individual sources would contain a fraction of truly relevant tuples from R_T , and a fraction of not truly relevant tuples from $U - R_T$. Assume that the search engines are picking these tuples independently. By the proof above in this subsection, independently picked tuples from R_T are likely to agree with much higher probability than independently picked tuples from $U - R_T$. This implies that the more number of tuples a source pick from R_T the more likely that other search engines agree with the results of the search engine. So agreement of results of a source by other sources directly implies that the source is likely to have picked many tuples from R_T . Otherwise, agreement implies trust indeed.

The possible concern with the argument above is that the assumption of independence between databases may not be fully true for web sources. For example two databases may use a vector space similarity between query and tuples for relevance assessment. Similarly, the data tuples in databases are likely to be correlated. These correlations means that result sets from different sources may be partially dependent. This dependence between the result sets manifests as increased probability of two databases to agree on true results as well as false results. This implies actual probabilities are likely to be greater than the probabilities given by Equation 9 and 10. But as long as no two sources have exactly same data and relevance measure the sources are at least partially independent. Partial independence between result sets means the probability of agreement for truly relevant results—compared to probability of agreement of irrelevant results—will be still be much high. This implies that even for partially independent real sources true relevance (trust) manifests as agreement.

Let us give a real world example of how the trust ranking would eliminate un-trusty sources. Assume that a book database decides to rank its result combining relevance to query and profit from selling the book. In this case the database is in a way lying to the user, saying these are the relevant results to the user but the ranking is in fact based on the relevance and profit. Since the other databases are using a relevance based the ranking, the agreement between the results provided by this database and other databases is likely to decrease. This decrease in agreement will decrease in its SourceRank.

4.2 Creating The Agreement Graph

To facilitate the computation of SourceRank, we represent the agreement between the source result sets as an agreement graph. Agreement graph is a directed weighted graph as shown in example Figure 2. In the graph the vertex set is the set of sources, and weighted edges represent the agreement between the sources. The edge weights are assigned equal to the normalized agreement values between the sources. For example, let R_1 and R_2 be the result

sets of the source S_1 and S_2 respectively. Agreement between R_1 and R_2 is calculated as defined in Equation 6. Let $a = A(R_1, R_2)$ be the agreement between the results sets. In agreement graph we create two edges: one from S_1 to S_2 with weight equal to $\frac{a}{|R_2|}$; and one from S_2 to S_1 with weight equal to $\frac{a}{|R_1|}$. The semantics of the weighted edge from S_1 to S_2 is: S_1 endorse a fraction of S_2 's tuples, where the fraction of tuples endorsed is equal to the weight of the edge in the agreement graph.

These agreement links described in the paragraph above are constructed based on the results to the sample queries. In addition to these agreement links, we add links of small weights between every pair of vertices, namely *smoothing links*. Like smoothing in any sample based method, these smoothing links account for the unseen samples. That is, though there are no agreement between the sampled results sets used to calculate the links, there is non-zero probability for some of the results to agree for queries not used for sampling. This probability corresponding to unseen queries are accounted by smoothing links with small weights. Adding this smoothing probabilities, the overall weight of link from S_1 to S_2 is,

$$w(S_1 \rightarrow S_2) = \beta + (1 - \beta) \times \frac{A(R_1, R_2)}{|R_2|} \quad (12)$$

where β is the smoothing factor and typically around 0.1. These smoothing links assure convergence of random walk based computation of SourceRank, since smoothing links strongly connect agreement graph. Finally we normalize the weights of out links from every vertex by dividing edge weights by sum of the out edge weights from the vertex. This normalization would make the edge weights equal to the transition probabilities for the random walk computations.

To build the agreement graph, we need to calculate the agreement of sources. To calculate the agreement, we crawl the web databases with sample queries. If the domain of the source is known, the source may be probed with queries from a single domain. If the domain of the source is not known, the source may be probed with queries from multiple domains. The queries used for sampling should be a representative sample of the set of domains for which the SourceRank needs to be calculated. The agreement between the query results are calculated based on common records in query results of sources (using Equation 6). For identifying common records in two result sets (record linkage) to calculate agreement exact similarity measures like string matching of individual attribute values of tuples, or approximate measures like bag of words similarity may be used. We detail on the method used to build the agreement graph in experimental section.

4.3 Calculating SourceRank

To calculate the SourceRank from agreement graph, let us enumerate the desiderata for the measure of reputation with respect to the agreement graph (i) highly pointed nodes should get higher rank (i) pointing by a source which is highly pointed should be more respected than pointing by a less pointed source. A natural choice of algorithm satisfying these properties is a markov random walk on the agreement graph, which has been used very successfully for PageRank [9].

The transition probabilities of the random walk is same as that of the edge weights. Recall from the last subsec-

tion that these edge weights are normalized. The graph is strongly connected and irreducible, hence the random walk will converge to unique stationary visit probabilities for every node. This stationary visit probability of a source would give the SourceRank of that source.

4.4 Discussion on SourceRank

We discuss the properties of SourceRank. We focus on suitability and use of SourceRank as a web database ranking method, and interesting properties requiring future research.

Spam Susceptibility: Even if we have much more number of independent spam sources than the trust worthy sources in search space, SourceRank will be able to eliminate spam sources. This is a direct implication of equation 10, i.e. probability of agreement of irrelevant independent tuples is very low. To spam SourceRank, the two conditions need to be satisfied (i) There should be large number of spam sources, number comparable to trust worthy sources in the search space. (ii) Spam sources should agree each other. If the degree of agreement between the spam sources is very high, the same fact may be used for spam detection. Also, if many sources are agreeing highly with each other, all of them, except one would be eliminated due to overlap consideration described in section 5. In short, spam susceptibility of SourceRank is low.

Time to Reflect Changes: SourceRank immediately reflects the source trust changes (disregarding delay due to sampling frequency). On the other hand, link based methods like PageRank do not reflect the changes in content quality. At best, PageRank follows changes in content quality with a time lag; as the content quality change is reflected as changes link structure of the surface web.

Domain Specificity: There may be a SourceRank computation for each domain separately, with each agreement graph spanning only sources in one domain. Otherwise single SourceRank computation may be performed on an agreement graph spanning the entire set of domains. Domain agnostic page rank combined with query specific coverage measure will give the right set of sources for the given query.

Query Specificity: SourceRank may be computed offline independent of query, and combine with query specific coverage measure of databases (like PageRank used for surface web). But it is possible to retrieve a set of relevant sources first based on query specific coverage measure. Then the SourceRank may be computed within the selected sources as described by Kleinberg in his authorities and hubs paper [26], and used for a second level ranking. We leave this for future research

Scalability: Once the agreement graph is created, computation required for SourceRank is the same as that for the PageRank—which is computationally feasible even for large graphs [?]. The computational complexity for creating source rank depends on the approximate similarity measure used for the agreement calculations, but is tractable.⁴

Specificity of Endorsement: For the databases, the specific tuples agreeing between the result sets are identifiable. The agreeing tuples between two databases are likely to be more reputed than non-agreeing tuples, by the the same argument as for the result sets. Also the number of

⁴The optimal agreement (as well as overlap) between two set of results using an arbitrary similarity measure may be formulated as a weighted bipartite matching problem, and can be solved in polynomial time.

agreeing tuples is a measure of weight of the endorsement. Note that this specificity of endorsement is an advantage over the hyper link based endorsement. This specificity may have interesting application of finding reputation of individual tuples.

5. OPTIMAL SOURCE SELECTION

We address the problem of selecting sources to maximize utility given in Equation 8 in this section. Since the deep web source selection has to deal with large number of sources computational complexity must be low. We analyze the complexity of optimal source selection, and prove that the optimal selection is NP-Hard. Also shown that a constant ratio approximation algorithm for optimal selection is hard.

The utility of source selection is given in Equation 8. We start by the subproblem of selecting a set of sources to maximize the coverage and overlap, which corresponds to the expected number of relevant tuples $E_T(DS)$ in Equation 7. We show that optimizing E_T is NP-Hard.

THEOREM 1. *Selecting sources to optimize expected number of relevant tuples E_T in Equation 7 is NP-Hard.*

Proof Sketch: The independent set problem can be reduced to selecting optimal set of sources considering coverage and overlap. See Appendix A for complete proof. \square

Recall that we formulated E_T in 7 is by ignoring higher order overlaps of sources. Considering only first order overlaps corresponds to the special case of source selection in which all the higher order overlaps are zero. So optimizing utility considering higher order overlaps is at least as hard as optimizing E_T in Equation 7. This directly implies that optimal source selection considering higher order overlaps are NP-Hard also. Similarly, the NP-Hardness of optimizing overall utility U in Equation 8 follows from the fact that optimizing E_T is a special case of optimizing overall utility U for which all the source reputations (i.e. $T(S)$) are equal.

To the best of our knowledge no other formal proof for NP-Hardness available already. Vassalos and Papakonstantinou [36] proved that the query optimization for integration systems is NP-Hard. They were considering the case in which a composite query is splitted into multiple sub-queries, to fetch the parts of answers corresponding to parts of the original composite query. They modeled query optimization as an AND/OR scheduling problem, where AND nodes corresponds to same composite query splitted into multiple sub-queries. Ours is a simpler scenario in which one query is not split into component queries. In this simple but common scenario, proof by Vassalos and Papakonstantinou fails, since there are no AND nodes. Our proof is strictly more generic, showing that simple source selection is NP-Hard even for an atomic query. This directly implies the proof by Vassalos and Papakonstantinou. In addition, the corollary given next answers the open problem left by Vassalos and Papakonstantinou, i.e. whether a constant ratio approximation is possible for query optimization in integration systems.

Beyond just NP-Hardness, the following result is a corollary of our proof,

COROLLARY 1. *The constant approximation algorithm for source selection considering overlap and coverage is hard.*

Algorithm 1 greedy-select($Q, S = S_1, S_2, \dots, S_n$)

```

1:  $Ch = \Phi$ 
2: while  $|Ch| < k$  AND  $S \neq \Phi$  do
3:   for all  $s \in S$  do
4:      $util(s) = C(s|Q) - \max_{S_i \in Ch} O(s, S_i|Q)$  (difference
       of coverage of  $s$  and maximum value of overlap with
       selected sources)
5:      $util(s) = (1 - \alpha)util(s) + \alpha T(s)$  (combine with rep-
       utation of the source)
6:   end for
7:    $s_{next} = s \in S$  with maximum value of  $util(s)$ , ties are
       broken arbitrarily
8:    $Ch = Ch \cup \{s_{next}\}$ 
9:    $S = S \setminus s_{next}$ 
10: end while

```

Proof: The proof of NP-Hardness theorem above shows that the independent set problem is a special case of source selection. This means a constant ratio approximation algorithm for optimal source selection would be a constant ratio approximation algorithm for the independent set problems also. Since constant ratio approximation algorithm for independent set algorithm is known to be hard by Garey and Johnson [16] and Håstad [20] the corollary follows. To define hard, in his seminal paper Håstad proved that independent set can not be solved within $n^{1-\epsilon}$ for $\epsilon > 0$ unless all problems in NP is solvable in probabilistic polynomial time, which is widely believed to be not possible.⁵ \square

6. EXPERIMENTAL SETUP

6.1 Greedy Selection

Since a tractable optimal strategy is infeasible, we set up our experiments using a greedy strategy. Greedy selection of sources has been discussed in number of research efforts [13, 31]. We formulate a simple greedy selection including coverage and overlap given in Algorithm 1. The algorithm starts by selecting the source with maximum value for combined coverage and trust. In Step 7 of the algorithm, the maximum value of first order overlaps for each unselected source s with any of the selected sources is calculated. The difference of this maximum overlap and coverage of s gives the $util(s)$. For example, if j sources are already selected, the overlap of s with all these selected j sources are calculated and maximum of these overlaps is subtracted from coverage of s to calculate $util(s)$. In Step 5 $util(s)$ is combined with the trust value of the source using the linear combination in Equation 8. In Step 8, the source s with maximum value for $util(s)$ is added to the selected set.

This greedy algorithm underestimates the overlap of the source, since only the maximum pairwise overlap is considered. The algorithm penalizes sources predominantly overlapping a single sources, but number of overlapping sources is small. Still, the overlap calculated by Algorithm 10 is a lower bound on total overlap. The improvements on the greedy selection is likely to be possible, and we leave this for future research.

6.2 Databases and Query Sets

⁵This belief is almost as strong as belief $P \neq NP$

Databases: We performed evaluations on the TEL-8 database list in the UIUC deep web interface repository [4]. Among the domains available, we used two of the most popular deep web database domains—books and movies. As we pointed out in the *GodFather* example in the introduction, these domains are popular enough to have many non-relevant answers disguised as relevant ones. Among the many web database interfaces in the TEL-8 repository, we removed web databases no longer working, database forms using HTTP post method,⁶ and sources having popups etc making crawling difficult. According to these criteria, we used all the twenty six book data bases in TEL-8, and twenty one movie databases. Since we found the number of movie databases in TEL-8 satisfying our criteria is small, our twenty one movie databases include nine databases not in TEL-8 repository. Note that number of databases are comparable to number of databases searched by the current web database search engines (like Google Videos, and Microsoft Live Video Search).

Sampling Query Set: We sampled the set of databases with key word queries, and parsed the result tuples. The result sets returned for these queries are used to calculate the agreement, coverage and overlap of these databases. As sampling query set, we used two list of complete movie and book names, fifty each from books and movies.

For the book domain, the sampling query set is chosen from two book lists: 1999 outstanding book list for college bound by American Library Association (ALA) [1], New York times best seller list on November 22 2008. The ALA list spans five categories: fiction, non-fiction, biography, drama, and poetry. New York times list contains three categories: fiction, non-fiction, advice. From the combination of these two lists, fifty books are chosen at random.

For Movies, sampling set used was the list of movies from a movie sharing site. This list contains un-categorized list of three hundred and twenty three movies spanning over fifty years. For sampling set, fifty movies are chosen at random from the list.

These lists are chosen to get a list spanning across the specific domain, and unbiased to any particular database. Also sampling list of queries should contain a mix of popular and uncommon objects. Since the popular items would be contained in every database, sampling will collect some statistics about every database in the list. Since uncommon queries would be answered only by a few databases, the coverage and overlap statistics would be able to discriminate between the databases.

Test Query Set: Test query sets for both book and movie domains are selected from different lists than the sampling query set. For books test set of queries, we used New York Times yearly number one book listing from year 1940 to 2007 [2]. This list contains two categories of fiction and non-fiction. From these two categories of fiction and non-fiction, hundred books each are chosen at random to form a list of two hundred books. For test query set of movie domain, we used list of movies from second edition of New York Times guide to best movies [3]. From this list, test set of two hundred movies are chosen at random.

6.3 Measurements

⁶Unlike HTTP get, post is not idempotent and submission of forms using post method may involve database updates; it is a courteous not to crawl web forms using post

Relevance Measure: A relevance measure is used to calculate the the coverage in Equation 1 and overlap in Equation 3. We used a relevance based on containment of the query object name in the result title. The containment measure is: if the query string contained as a substring of the answer title (ignoring upper/lower case), the results are considered to be of relevance value one. Otherwise the relevance value of the result is zero. Containment measure is a modification of exact string comparison binary relevance measure. The string comparison measure is modified based on the observation that many result titles, which are super-strings of the actual object name queried, are relevant. For example, for the query “The GodFather” answers like “The GodFather Trailer” and “The GodFather (DVD)” etc are very commonly returned. The containment similarity very well captures similarity of these results with query. Note that our sampling queries are full names of objects, and containment similarity essentially means the title contains the full name of object as a substring.

Agreement: For measuring the agreement between two result sets, we need to find out which titles in the first result set agree with which results in the second result set. This means we need a similarity measure for calculating agreement. We used the same containment similarity measures as that used for measuring the relevance. For relevance we were measuring the similarity between the query string and the individual result titles. For calculating agreement between the result sets, we need to measure the similarity between the pair of results in two result sets returned by the databases. To calculate agreement between two databases, we used top-10 results from two databases for the same query. The pairwise agreement is calculated for individual results. For example, let “The GodFather” is a result from the first database and “The GodFather(DVD)” is the result from the second database. Since “The GodFather” is a substring of “The GodFather(DVD)” (or if the “The GodFather(DVD)” is a substring of “The GodFather”), they are considered to be agreeing each other. Also we avoided matching one result in a set with multiple results in the other result set; i.e if “The GodFather” is matched with “The GodFather(DVD)” in the second result set, the same result—“The GodFather”—will not be matched again with another result in second result set. Similarly second result set—“The GodFather(DVD)” —will be matched maximum of only one result set in first result set.

For calculating agreement, we used a slightly different crawling method than sampling. Instead of using the full object names to crawl as for coverage and overlap, we used a partial descriptions of objects. These partial descriptions of objects are generated by deleting words in full object names. The words are deleted with a probability of 0.5 at random. If the entire object name is deleted in this process, original object name is replaced. These set of partial object names are used for crawling and agreement is calculated as described in the paragraph above. This variation in crawling is based on the intuition that the agreement of answers is less likely when queries are partial descriptions of object names, than queries are full object names. This implies that agreement is more indicative of relevance of results for partial names.⁷ Partial queries—fifty each for books and movie

⁷Agreement based on answers to full descriptions as queries gave similar results as we have shown below; but agreement based on answers to partial descriptions showed performance

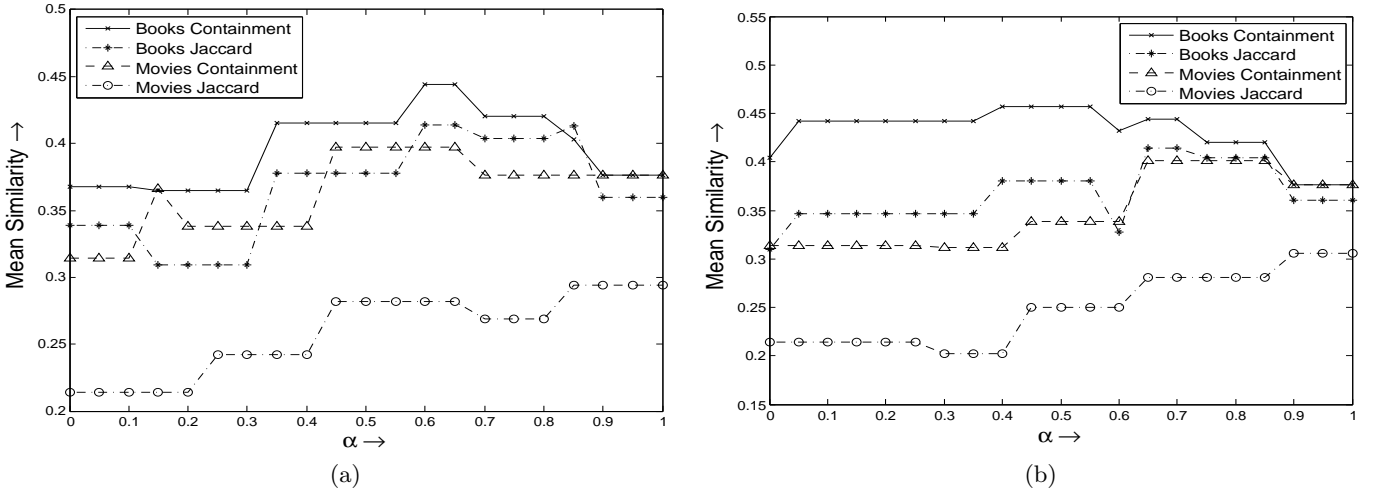


Figure 3: Performance of combined source selection for books and movie domains considering SourceRank against (i) coverage only source selection in 3(a) (ii) coverage and overlap based source selection in 3(b). In the figures, $\alpha = 0$ correspond to coverage only source selection in 3(a) and coverage and overlap based selection for 3(b). For both figures, $\alpha = 1$ corresponds to the SourceRank only selection. As α increases from left to right, weightage to SourceRank increases. y -axis indicates the mean similarity of top-10 results. For 3(a) combined source selection considering SourceRank and coverage significantly outperforms coverage only source selection. Similarly for 3(b) the combined source selection shows much better results than coverage-overlap source selection. Both containment, and jaccard coefficients with bag of words similarity shows similar trends for both movies and books domain.

domains—are formed from the same set of fifty crawling queries described above.

Coverage Measure: Coverage of a database is measures as in Equation 1 and Equation 2, using containment similarity based relevance measures described above. The results from the sampling set with full objects names are used for calculation.

Overlap Measure: Overlap is measure as the measurement of agreement with two differences. First difference is that the matched items are multiplied by the minimum of relevances of the matched items as in Equation 3. To measure the relevance, we used the the containment relevance as for the coverage. Second differences is that the crawl results using full object names are used as for the coverage calculation.

6.4 Sampling

Getting a true random sample of tuples from a web database hidden behind the form interface is not easy [11]. We used simple sampling by sending the queries in the sampling set to the key word search fields provided by the web databases. If there are multiple search attribute fields provided by the web forms, we used only title fields. This simple sampling can be performed on any web database providing a key word query interface. We entered the set of sampling queries one by one in the key word search field, submitted the web form, and parsed the results.

The sampling is automated here, but the manual effort was required for writing the regular expressions to parse the results returned in HTML pages. Parsing these results from the returned HTML pages has been addressed by number of research efforts [19, 7]. We tried using publicly available

software to automate this parsing, but they were not satisfactory. Since all the keywords are sent to a single search field, schema mapping is very simple. After submitting the sampling query, we retrieved only first page of results from the result set. For our experiments, we parsed only the titles (names) of the results. This is sufficient for our experimental validation, since titles almost uniquely identify the objects for the domains considered.

6.5 Performance Measures

The test for effectiveness of SourceRank is the increase in relevance of the results. To assess the relevance of the results, we leverage on the fact that typical deep web queries target the specific real world objects like movies or books. Queries are in general a partial description of the targeted real world object. Relevant results are a set of one or more real world objects. For example, the user may want “The GodFather I” and he may give a query “GodFather”. The “The GodFather I” is certainly a relevant result, and many be other result like “The GodFather II” and “The GodFather III” may be relevant also. We formulate our evaluation based on the observation that the original object (e.g. “The GodFather I”)—whose partial description is executed as query (e.g. “GodFather”)—is one of the relevant results.

Based on this observation, we set up our experiments as following. From the names of books and movies test set described above, words are deleted at random with a deletion probability 0.5 to form the queries. These queries with randomly deleted words are issued to the selected set of web databases and collected the top-k results. We compare these results with the original object name—object name with no words deleted—to assess the relevance of the results. For comparison we tried both containment and bag of words

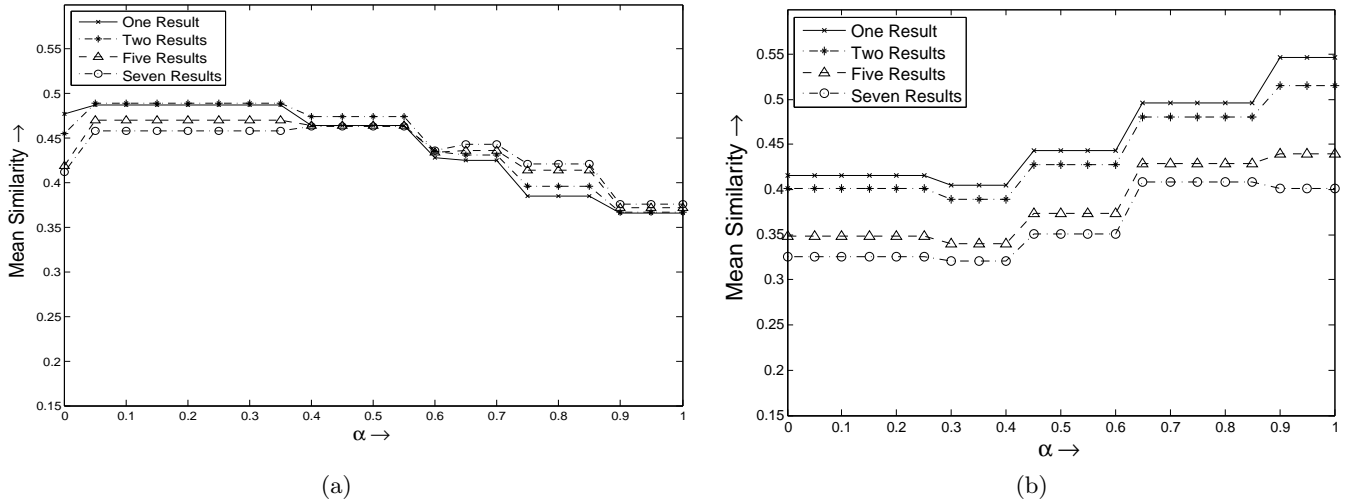


Figure 4: Effect of varying number of results retrieved from individual sources considered on similarity of combined results for (i) Books domain in 4(a) (ii) Movies domain in 4(b). Number of sources selected is four. For both the domains, the selection combining SourceRank and overlap-relevance invariably outperforms the coverage-overlap only sources selection. Also note that as k increases, mean similarity of the top- k results goes down. This trend is intuitive since the result set is ranked by relevance individual sources. As the number of results fetched increases mean relevance and hence the mean relevance comes down.

with jaccard coefficient similarity.

As an example, assume that our original object name is “The GodFather I”. We delete words at random from the name. Assume that “The” and “I” are deleted and we get back the string “GodFather”. We issue this string as query and assess the relevance of the result by comparing the source results with “The GodFather I”. The idea behind this evaluation is that a better result set will contain more occurrences of original object name. If a source is successful in guessing the object in users mind from the partial query, the source would be able to guess the full object (with no words deleted) from the query issued. So the selection choosing more sources returning original object will be better.

We decided to use simple relevance and comparison measures—like containment and bag or words similarity as described above. Please note that we are using these relevance measures just to compare performances of SourceRank and existing source selection methods. These simple measures allows to compare the effectiveness of source rank without being concerned about the parameters of relevance assessment or comparison measures. Using another comparison or relevance measure, the absolute numbers might vary, but ratio of performances of two methods is likely to remain the same.

7. RESULTS AND DISCUSSIONS

Using the performance measure described in section above, we assessed the effectiveness of SourceRank against the coverage-overlap source selection. Also we measured the effect of varying two important source selection parameters: (i) Number of sources selected (ii) Number of results fetched from each source. For these assessments, we send the query test set to each selected source, retrieved the results, and calculated the performance as described in performance measurement subsection above.

For the experiments, the SourceRank is calculated by it-

erative method on the agreement graph. The SourceRank of every source is normalized against the highest value of the SourceRank. Similarly, the coverage-overlap score is also normalized against the highest value of coverage overlap score at every iteration. The sources are selected according to the greedy Algorithm 1 described above.

As the first experiment, we analyzed the effect of varying weightage α used in linear combination in Equation 8. The results are illustrated in Figure 3(a). For Figure 3(a) we considered only coverage (overlaps are ignored), to avoid interactions of overlap measurement and agreement based page rank. We selected top four sources. Top-10 results from these sources are retrieved and mean similarity of these results are compared against the original object name; as described in the performance evaluation subsection above. In the Figure 3(a) the y-axis shows mean similarity of the result set returned with the full object description (e.g. similarity with “The GodFather I”). Higher the similarity, better the relevance of the results. For comparison, we used both the containment similarity and bag of words with jaccard coefficient. In the Figure 3(a) the value $\alpha = 0$ at left corresponds to the coverage only source selection. As the α increases from left to right, the weightage for SourceRank increases. At $\alpha = 1$ the weightage for the coverage is zero, and sources are selected considering only SourceRank.

Note that in the Figure 3(a) above the combined selection considering coverage and SourceRank outperforms coverage only source selection for both movies and books; and for both bag of words and containment similarity. In some cases the increase in similarity is as much as 43% (for movie jaccard). Minimum increase in accuracy in percentages is for containment similarity measure for books—12%. Generally similarity peaks in the range of α between 0.5 to 0.8 (except for jaccard movies, for which similarity at $\alpha = 1$ is slightly higher). Also note that both similarity measures for the same domain (movies and books) shows similar peaks and valleys (i.e similar shapes of curves), showing agree-

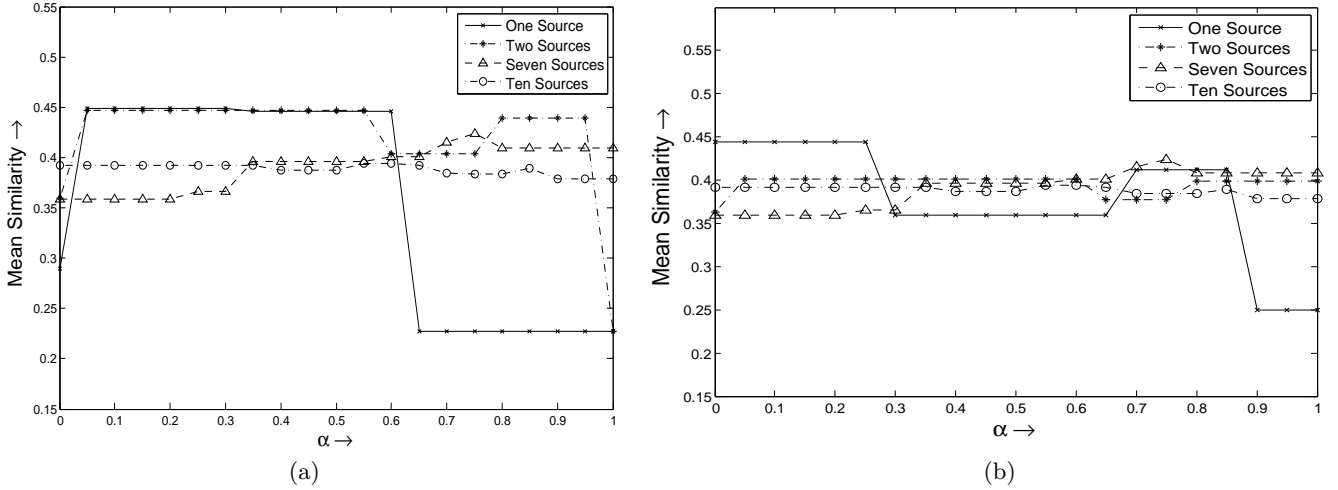


Figure 5: Effect of varying number of sources selected for (i) Books domain in 5(a) (ii) Movies domain in 5(b). We considered top-10 results from every source. For both the domains, the selection combining SourceRank and coverage-overlap outperforms the coverage-overlap only sources selection; irrespective of number of sources selected (single source selection in movies is an anomaly, with coverage-overlap source selection as good as combined selection). Also note that as number of sources increases, peak mean similarity of results goes down in general as number of selected sources increases. This is intuitive since the higher ranked sources are likely to return better results, and average quality of results returned by the sources decreases as more sources are selected.

ment with each other. This shows the robustness of the performance evaluation to specific similarity measure used.

In the second experiment we repeated the same evaluation, with overlap enabled. i.e. the sources are selected according to the greedy Algorithm 1, considering SourceRank, coverage and overlap. The results are illustrated in Figure 3(b). The results are confirming with the source selection considering only coverage and SourceRank in Figure 3(a). The combination of SourceRank with coverage and overlap considerably outperforms coverage-overlap only source selection in both domains; and for both the similarity measures.

In our next experiment we changed the number of results considered from selected sources for evaluating similarity. The number of sources selected is kept constant at four as for the previous experiments. We considered both coverage and overlap, and used containment similarity for this and all the remaining experiments. We calculated the similarity value of top- k result from each selected source, where k varies over values, one, two, five and seven. The results are illustrated in Figure 4(a) and 4(b). For both books and movies, the selection combining SourceRank and overlap-relevance invariably outperforms the coverage-overlap source selection in every case. Also note that as k increases, highest mean similarity of the top- k results goes down. This is intuitive, since the result set is ranked by relevance by individual sources. The decrease in mean similarity with increase in number of results considered from each sources is a direct indication that our similarity measure is a reasonable measure of relevance. Also, the comparative decrease in

Next, we measured the effect of varying number of sources selected on the performance. We kept the number of results considered from each source constant at ten, and varied number of sources selected from one to ten. The results for

both book and movie domain are illustrated in Figure 5(a) and 5(b). The results confirms that combination SourceRank with coverage-overlap is likely to give better results.

As the number of sources selected increases the peak average similarity of the results goes down. This is intuitive since the top sources are likely to return better results. Note that this should not be confused with general notion of getting better results as we fetch data from more sources. As we increase the sources in a real information integration system, we need to fetch only lesser number of results from each source to fetch top- k results in combined result set. The increase in quality of the combined result set is a direct implication of the fact that we are fetching only high quality results from each source. But in this experiment—since we are picking top-10 results from each source irrespective of number of sources chosen—our average relevance goes down as number of sources selected increases. This is a direct implication of decrease in quality of results returned by the lower ranking sources.

Two other trends need to be noticed also. First, as the number of sources selected increases (the curve corresponding to 10 sources) the average similarity curve almost becomes flat; not varying with α . This flattening is due to the fact that both the methods already picked good sources from the list of 22 and 27 in the list, and remaining sources are not much better from each other. So irrespective of methods, the relevance of the result remains the same. Another observation is that the set of experiments choosing only top source (i.e. top-1 source) shows sharp jumps and anomalous trends. This is expected since the result of the experiments are statistically more significant as the number of sources selected is more. The curves corresponding to selecting smaller number of sources is likely to show anomalous peaks and jumps, resulting in curves which are less smooth.

In short these set of experiments clearly illustrates, the effectiveness and robustness of SourceRank based source selection. For all these experiments source selection combining SourceRank and coverage-overlap clearly outperforms the coverage-overlap based source selection, without exceptions. Along with this increase in quality of results, the properties of SourceRank discussed in Section 4.4 makes SourceRank an excellent candidate for source selection in the deep web.

8. CONCLUSIONS

A compelling holy grail for the database research is to exploit the structured deep web sources through database-style query processing. An immediate problem posed by this quest is source selection, i.e., selecting relevant sources to answer a query. Past approaches to this problem implicitly assume that all sources are of equal “quality” or trustworthiness and focus on local properties of the sources (such as latency, coverage and overlap). The sheer number and uncontrolled nature of the sources in the deep web leads to significant variability among the sources, and necessitates a more robust measure of source reputation or trustworthiness. To this end, we proposed SourceRank, a *global* measure derived solely from the degree of agreement between the results returned by individual sources. SourceRank plays a role akin to PageRank but for data sources. Unlike PageRank however, it is derived from implicit endorsement (measured in terms of agreement) rather than from explicit hyperlinks. Just as PageRank can be used along with local similarity measures to improve search, our empirical results show that SourceRank can be gainfully combined with local measures such as coverage and overlap to improve source selection.

9. REFERENCES

- [1] American library association outstanding books for college bound. <http://www.ala.org/ala/mgrps/divs/yalsa/booklistsawards/outstandingbooks/obcb99.cfm>.
- [2] New york times best sellers number ones. <http://www.hawes.com/number1s.htm>.
- [3] New york times guide to best 1000 movies. <http://www.nytimes.com/ref/movies/1000best.html>.
- [4] Uiuc tel-8 web interface repository. <http://metaquerier.cs.uiuc.edu/repository/datasets/tel-8/index.html>.
- [5] White paper on deep web. <http://grids.ucs.indiana.edu/courses/xinformatics/searchindik/deepwebwhitepaper.pdf>.
- [6] S. Agrawal, S. Chaudhuri, and V. Narasayya. Automated Selection of Materialized Views and Indexes in SQL Databases. *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 496–505, 2000.
- [7] A. Arasu and H. Garcia-Molina. Extracting structured data from Web pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 337–348. ACM Press New York, NY, USA, 2003.
- [8] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. Improving collection selection with overlap awareness in P2P search engines. *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 67–74, 2005.
- [9] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.
- [10] T. Cheng and K. Chang. Entity Search Engine: Towards Agile Best-Effort Information Integration over the Web. *Proc. of CIDR2007*, pages 108–113, 2007.
- [11] A. Dasgupta, G. Das, and H. Mannila. A random walk approach to sampling hidden databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 629–640. ACM Press New York, NY, USA, 2007.
- [12] A. Doan and A. Halevy. Efficiently Ordering Query Plans for Data Integration. In *Proceedings of ICDE*, pages 393–402. IEEE Computer Society Press; 1998, 2002.
- [13] D. Florescu, D. Koller, A. Levy, and A. Pfeffer. Using Probabilistic Information in Data Integration. In *Proceedings of VLDB*, pages 216–225. IEEE, 1997.
- [14] J. French, A. Powell, J. Callan, C. Viles, T. Emmitt, K. Prey, and Y. Mou. Comparing the performance of database selection algorithms. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 238–245, 1999.
- [15] N. FUHR. A Decision-Theoretic Approach to Database Selection in Networked IR. *ACM Transactions on Information Systems*, 17(3):229–249, 1999.
- [16] M. R. Garey and D. R. Johnson. The complexity of near-optimal graph coloring. In *Journal of ACM*, 1976.
- [17] L. Gravano, P. Ipeirotis, and M. Sahami. QProber: A system for automatic classification of hidden-Web databases. *ACM Transactions on Information Systems (TOIS)*, 21(1):1–41, 2003.
- [18] P. Haas and C. König. A bi-level Bernoulli scheme for database sampling. *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 275–286, 2004.
- [19] J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo. Extracting Semistructured Information from the Web. In *Proceedings of the Workshop on Management of Semistructured Data*, pages 18–25. Tucson, Arizona: ACM, 1997.
- [20] J. Håstad. Clique is hard to approximate within n . In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 627–636, 1996.
- [21] B. He, Z. Zhang, and K. Chang. MetaQuerier: querying structured web sources on-the-fly. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 927–929, 2005.
- [22] P. Ipeirotis, E. Agichtein, P. Jain, and L. Gravano. To search or to crawl?: towards a query optimizer for text-centric tasks. *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 265–276, 2006.
- [23] P. Ipeirotis and L. Gravano. Distributed search over the hidden web: hierarchical database sampling and selection. *Proceedings of the 28th international*

- conference on Very Large Data Bases-Volume 28, pages 394–405, 2002.
- [24] P. Ipeirotis and L. Gravano. When one sample is not enough: improving text database selection using shrinkage. *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 767–778, 2004.
- [25] P. Ipeirotis, L. Gravano, and M. Sahami. Probe, count, and classify: categorizing hidden web databases. *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 67–78, 2001.
- [26] J. KLEINBERG. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632, 1999.
- [27] Z. Liu, C. Luo, J. Cho, and W. Chu. A Probabilistic Approach to Metasearching with Adaptive Probing. *Proceedings of the 20th International Conference on Data Engineering (ICDE04)*, 1063(6382/04):20–00.
- [28] J. Madhavan, P. Bernstein, A. Doan, and A. Halevy. Corpus-based schema matching. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 57–68, 2005.
- [29] J. Madhavan, A. Halevy, S. Cohen, X. Dong, S. Jeffery, D. Ko, and C. Yu. Structured Data Meets the Web: A Few Observations. *Data Engineering*, 31(4), 2006.
- [30] J. Madhavan, S. Jeffery, S. Cohen, X. Dong, D. Ko, C. Yu, and A. Halevy. Web-scale Data Integration: You can only afford to Pay As You Go. *Proc. CIDR*, 7, 2007.
- [31] Z. Nie and S. Kambhampati. A Frequency-based Approach for Mining Coverage Statistics in Data Integration. *Proceedings of the 20th International Conference on Data Engineering*, page 387, 2004.
- [32] H. Nottelmann and N. Fuhr. Evaluating different methods of estimating retrieval quality for resource selection. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 290–297, 2003.
- [33] A. Powell and J. French. Comparing the Performance of Collection Selection Algorithms. *ACM Transactions on Information Systems*, 21(4):412–456, 2003.
- [34] S. Raghavan and H. Garcia-Molina. Crawling the Hidden Web. *VLDB*, pages 129–138, 2001.
- [35] L. Si and J. Callan. Unified utility maximization framework for resource selection. *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 32–41, 2004.
- [36] V. Vassalos and Y. Papakonstantinou. Using knowledge of redundancy for query optimization in mediators. In *Proceedings of the AAAI Workshop on AI and Information Integration*, pages 29–35, 1998.
- [37] A. Wright. Searching the deep web. 2008.
- [38] W. Wu, C. Yu, A. Doan, and W. Meng. An interactive clustering-based approach to integrating source query interfaces on the deep Web. *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 95–106, 2004.
- [39] B. Yu, G. Li, K. Sollins, and A. Tung. Effective keyword-based selection of relational databases. *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 139–150, 2007.
- [40] Z. Zhang, B. He, and K. Chang. Understanding Web query interfaces: best-effort parsing with hidden syntax. *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 107–118, 2004.

APPENDIX

A. PROOF OF THEOREM I

PROOF. Independent set problem can be reduced to the problem of selecting optimal set of sources considering coverage and overlap. Consider an unweighed graph G of n vertices $\{s_1, s_2, \dots, s_n\}$ represented as an adjacency matrix. This conversion is clearly polynomial time. Now, consider the values in the adjacency matrix as overlap values between the sources to be selected. Let the sources have same coverage. Clearly in this set of n sources from $\{s_1, s_2, \dots, s_n\}$, the optimal set of size k will have k pairwise non-overlapping sources as the top k sources. But the set of k independent sources corresponds to an independent set of size k in graph G . \square