SourceRank: Relevance and Trust Assessment for Deep Web Sources Based on Inter-Source Agreement

Raju Balakrishnan #1, Subbarao Kambhampati #2

# *Computer Science and Engineering, Arizona State University*
Tempe AZ USA 85287
{1rajub,2rao}@asu.edu

ABSTRACT

One immediate challenge in searching the deep web databases is *source selection*—i.e. selecting the most relevant web databases for answering a given query. The existing methods of database selection (both text and relational databases) uses relevance measures based on the similarity with the queries for the quality assessment of the sources. Existing methods have two deficiencies for applying to the open collections like the deep web. First is that the methods are agnostic to the correctness (trustworthiness) of the sources. Secondly, since the existing measures are fully dependent on the query similarity, they do not consider the popularity of the results for computing the probability of relevance. Since number of sources provide their own answer sets to the same query in the deep web, the agreements between theses answer sets are likely to be helpful in assessing the relevance and trustworthiness of the sources. We start with this hypothesis and compute the agreement between the sources using entity matching methods. Agreement is modeled as a graph with sources at the vertices. On this agreement graph source quality scores—namely *SourceRank*—is calculated as the stationary visit probability of a random walk. Our evaluations on the online deep web sources show that the relevances of the sources selected by SourceRank is improved by 20-50% over the existing methods; and that SourceRank of a source reduces linearly with corruption levels. Also we demonstrate that SourceRank can be combined with Google Base ranking to improve the precision by 22-60% and to select sources better trusted by the users.

1. INTRODUCTION

By many accounts, surface web containing HTML pages is only a fraction of the overall information available on the web. The remaining is hidden behind a welter of web-accessible relational databases. By some estimates, the data contained in this collection—called deep web—is estimated to be in tens of millions [12]. 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.

Source selection for text and relational databases involving relevance, coverage, and the overlaps between sources has received some previous attention (c.f. [13, 4, 7, 14]). These existing approaches are focused on assessing relevance of a source based on local measures, as they evaluate quality of a source based on the similarity between the answers provided by the source and the query. The answers to the same query by the other sources are disregarded. For applying in the deep web, this pure query based local approach for source selection has the following two deficiencies:

1. Local relevance assessment is susceptible to easy manipulation, since the measure does not consider popularity of results. Since the local features can be easily tampered by the content owner to boost source's rank (e.g. a database may return many fabricated tuples containing search key words to boost its relevance value). To give a real world example of these problems, we issued the query *godfather movie* to Google Product Search. None of the first page results in relevance ranking contains a Godfather movie (DVD, Blu-Ray etc).[1] The first page results is populated with results like *Godfather Movie Poster*, *Godfather Movie T-Shirt* etc. These results are ranked high as they contain all the query terms in the title; whereas movie result are ranked low since they do not contain search term *Movie.*

2. The source selection is agnostic to the trustworthiness of the answers. For example, many queries in Google Product Search returns answers with unrealistically low prices. While we proceed towards the checkout, many of these low priced results turned out to be non-existent, a different product with same title (e.g. solution manual with same title by a different author) etc. Relevance is a measure of whether query is answered by the tuple; and trustworthiness is a measure of whether the answer is correct. Relevance and trustworthiness of an answer are independent quantities. In particular, the corruption in unspecified attributes in the results generates untrustworthy results; whereas difference in specified attributes generates irrelevant results. Any query based relevance measure is insensitive to trustworthiness.

WWW2009, April 20-24, 2009, Madrid, Spain.
.

[1] Google Product Search works over Google Base. Though this is a warehousing approach, the problem of uncontrolled collections of sources is common.

A global measure of trust and popularity is particularly important for uncontrolled collections like deep web, since sources try to artificially boost their rankings. A global relevance measure should consider popularity of a result, since the popular results tends to be relevant. Moreover, it is impossible to measure trustworthiness of sources based on local measures; since measure of trustworthiness of a source should not depend on any information the source provides about itself. In general, the trustworthiness of a particular source is reflected as the endorsement of the source by other sources. In surface web the trustworthiness of a page as well as popularity is calculated based on the endorsement as hyper-links from other pages, like in PageRank [5]. 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 and probability of relevance of a source based on how well the results from the source are agreed upon by other sources. Two sources agree with each other if they return the same tuple in answer to the same query. While trustworthiness and relevance are orthogonal quantities, both are reflected as the agreement of other sources to the answers provided by the source. For example, in the *Godfather* query example above, since Godfather movie is a popular result returned by large number of sources, a global relevance assessment based on the agreement of the results would have ranked the movie instances high. In the case of untrustworthy answers, the corruption can be captured by an agreement based method, since other legitimate sources answering same query are likely to disagree with the corrupted result. We provide a formal explanation for why agreement implies trust and relevance in Subsection 3.1 below.

Different web databases represent the same object in syntactically varying manner, making it hard to calculate agreement. To solve this problem, we augmented the existing record linkage models in relational databases [8] with named entity matching methods to calculate the agreement between the web databases. Also, though the data is stored in relational databases, the keyword queries and non-cooperative nature of the sources requires sampling methods in text database selection [6]. Thus computing agreement between the deep web sources requires combination and adaptation of methods in relational databases, text database selection in information retrieval and natural language processing.

The overall contributions of the paper are: (i) An agreement based method to calculate relevance of the deep web sources based on popularity. (ii) An agreement based methods to calculate trustworthiness of deep web sources. (iii) Domain independent computation of agreement between the deep web databases and formal evaluations, and comparison against Google Product search.

We evaluated the ability of SourceRank to select trustworthy and relevant sources in two sets of web sources—(i) set of online databases in TEL-8 repository [2] (ii) Google Base [1]. The evaluation shows SourceRank improves relevance of source selection by 20-50% over the existing methods. Also SourceRank combined with Google Base result ranking improves the top$-k$ precision of results by 22-60% over stand-alone Google Base. Trustworthiness of source selection is evaluated as the ability to remove sources with corrupted unspecified attributes. The SourceRank reduces almost linearly with source corruption. Similarly, user ratings of the sources selected by the SourceRank show improvement over the baseline methods.

2. RELATED WORK

Searching the deep web has been identified as the next big challenge in information management [16].

Current relational database selection methods predominantly try to maximize the number of distinct relevant records from minimum number of sources, to minimize cost [13]. The parameter widely considered for this minimum cost access is coverage of sources. Coverage of a database is a measure of number of relevant tuples to the query in the database. Hence cost based web database selection is formulated as selecting the least number of databases maximizing sum of coverages. Related problem of collecting statistics for source selection has been researched in detail also [13].

Considering research in the text databases selection, Callan *et al.* [7] formulated method CORI for query specific selection of text databases based on relevance. Cooperative and non-cooperative text database sampling [6] and selection considering coverage and overlap to minimize cost [15, 14] are addressed by number of papers.

In his early work of combining multiple retrieval methods to improve the retrieval accuracy for text documents, Lee [11] observes that the different methods are likely to agree on same relevant documents than on irrelevant documents. This observation confirms the argument in this paper.

Framework for trust assessment of facts based on agreement of web pages has been discussed by Yin *et al.* [17]. Their work assumes a question answering scenario, where queries have single correct answers (questions like *Who is the director of The Godfather?*), which is not true about deep web search queries since they may have multiple correct answers. Also, need for a domain specific method to compute the probability with which the fact implies another fact, ignoring record linkage, and disregarding influence of relevance on agreement limits the applicability of the method on the deep web. Dong *et al.* [10] extend this model considering source dependence; but uses the same basic model as Yin *et al.*

The relevance computation based on agreement of sources—to the best of our knowledge—is a novel idea. Though the notion of agreement based computation of trust is known as described above, we present many extensions required for calculating agreement on the deep web.

3. SOURCERANK: TRUST AND RELEVANCE RANKING OF SOURCES

In this section we elaborate the argument that the relevance and trustworthiness of a database manifests as agreement of other databases. We devise a method to calculate SourceRank—a trust and relevance 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 weighted 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 SourceRank.

3.1 Agreement as Endorsement

We show in this section why agreement in fact implies endorsement. First let us argue that two independently picked relevant and trustworthy tuples are likely to agree each other with significantly higher probability than two independently picked irrelevant tuples (we agree that the assumption of sources picking tuples independently may not be fully correct, we relax this independence assumption below). Let $P_A(r_1, r_2)$ denotes the probability of agreement of two independently picked trustworthy and relevant tuples by two sources.

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

where R_T is the complete set of relevant and trustworthy tuples.

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

$$P_A(f_1, f_2) = \frac{1}{|U - R_T|} \tag{2}$$

where U is the search space (the universal set of all tuples searched). For any web database search, the search space is much larger than the set of relevant tuples, i.e. $|U| \gg |R_T|$. Applying this inequality in Equation 1 and 2 directly implies that

$$P_A(r_1, r_2) \gg P_A(f_1, f_2) \tag{3}$$

To provide an intuitive understanding of magnitude of these probabilities, let us consider an example. Assume that the user issues the query *Godfather* for the Godfather movie trilogy. Assume that three movies in trilogy *The Godfather I*, *II* and *III* are the results 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 $P_A(r_1, r_2) = \frac{1}{3}$ 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. It is easy to extend the argument above to answer sets from single answers.

A possible concern with the above argument is that the assumption of independence between databases may not be completely true for web sources. 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 (Equation 1)—compared to probability of agreement of irrelevant results (Equation 2)—will still be much higher. This implies that even for partially independent sources relevance and trust manifests as agreement. Though full dependence is intuitively hard to happen on real web databases, the only way to conclusively prove that web databases are at least partially independent is testing our model on actual web databases; which we do in experiments in Section 5.

3.2 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 1. In the graph, the vertices represent 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 described in Section 4. 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.

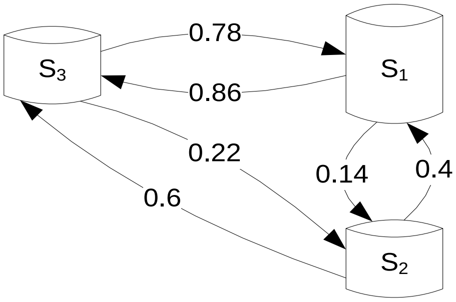


Figure 1: A sample agreement graph structure of three sources.

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 is no agreement between the sampled results sets used to calculate the links, there is a 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 probability, the overall weight $w(S_1 \rightarrow S_2)$ of the link from S_1 to S_2 is computed as,

$$A_Q(S_1, S_2) = \sum_{q \in Q} \frac{A(R_{1q}, R_{2q})}{|R_{2q}|} \tag{4}$$

$$w(S_1 \rightarrow S_2) = \beta + (1 - \beta) \times \frac{A_Q(S_1, S_2)}{|Q|} \tag{5}$$

where R_{1q} and R_2q are the answer sets of S_1 and S_2 for the query q, and Q is the set of sampling queries over which the agreement is computed. β is the smoothing factor and typically around 0.1. These smoothing links strongly connect agreement graph (strong connectivity is important for SourceRank calculation). The details of calculating agreement for query q—$A(R_{1q}, R_{2q})$—are described in Section 4 below. Finally we normalize the weights of out links from every vertex by dividing the 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.

3.3 Calculating SourceRank

To formulate a method to calculate the SourceRank on the agreement graph, we ask the question what would be the rational behavior of a deep web searcher, if he is provided with this agreement graph. Since the searcher does not have an idea about which node is relevant and trustworthy, assume that the searcher starts on a random database node. The searcher may find that the database is not interesting and choose to restart his search in another random node; these random restart is represented by the smoothing links in the agreement graph. If he finds the database useful, for further search he would choose to traverse one of the database having agreeing data by the current database. This is based on the logic that since the current database has useful information, the sources agreed by the current database is likely to have useful information also (we do not consider possible need for diversity of results in the case of multiple results search here). The searcher may traverse one of the outgoing links and search in database at the other end of the link. An outgoing link has an associated weight; and the weight is proportional to the fraction of tuples the current database is agreeing in the target. So it is logical for the searcher to choose an outgoing link randomly with a probability proportional to the weight of the link; rather than uniform random selection of a link (If he always deterministically chooses the out link with highest weight, the search is not complete and he may never reach some vertices in the graph). If he continues this process; he would be performing a weighted random walk on the agreement graph. The transition probabilities of the random walk are the same as that of the edge weights.

In the random walk described above, the probability with which the searcher would visit different databases will be the stationary visit probability of the random walk on the database nodes. 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; which is equal to the visit probability of the random searcher described above for that source.

4. AGREEMENT COMPUTATION AND SAMPLING

Since deep web sources present an interesting middle ground between free-text sources assumed in IR literature, and fully structured sources assumed in the database literature, the agreement computation over their results presents challenges that cannot be handled by the traditional methods from either discipline. In the following subsection, we will describe our approach for agreement computation in three phases– how the attribute values are compared, how the tuples are compared and how the answer sets are compared. Subsequently, we describe our database sampling method.

4.1 Computing Agreement

Computing agreement between the sources involves three levels of similarity computations, as described below.

Attribute value similarity: Cohen *et al.* [8] shown that assumption of common domains is far from the truth in web databases (common domains means names referring to the same entity is the same for all databases, or can be easily mapped to each other by normalization). For example, title of the same movie is represented as *Godfather, The: The Coppola Restoration* in one database and *The Godfather - The Coppola Restoration Giftset [Blu-ray]* in another database. Recognizing the semantic similarity between attribute values in different databases is not straightforward.

The textual similarity measures works best for scenarios like web databases where common domains are not available [8]. Since the challenge of matching attribute values is essentially a name matching task, we calculate the agreement between attribute values using SoftTF-IDF with Jaro-Winkler as the similarity measure. (please refer to Cohen *et al.* [9] for details). Comparative studies showed that this combination provide best performance for name matching [9]. For pure numerical values (like price) we calculate similarity as the ratio of the difference of values to the maximum of the two values.

Tuple similarity: The tuples are modeled as a vector of bags adapting model by Cohen [8]. The problem of matching between two tuples based on the vector of bags model is shown in Figure 2. If we know which attribute in t_1 maps to which attribute in t_2, the similarity between the tuples is simply the sum of the similarities between the matching values. Finding this mapping is the well known automated answer schema mapping problem in web databases. We do not assume availability of predefined answer schema mapping and reconstruct the schema mapping based on the attribute value similarities, as we describe below.

Once the pairwise value similarities are calculated as described above (this has computational complexity $O(|t_1||t_2|)$), tuple similarity computation is same as the well known maximum weighted bipartite matching problem. Hungarian algorithm gives the lowest time complexity for the maximum matching problem, and is $O(V^2 log(V) + VE)$ (V is the number attribute values to be matched, and E is the number of similarity values). Also E is $O(V^2)$ for our problem, and overall time complexity is therefore $O(V^3)$. We use the $O(V^2)$ greedy matching algorithm as a favorable balance between time complexity and performance. To match tuples, say t_1 and t_2 in Figure 2, the first attribute values of t_1 is matched against the most similar attribute values of t_2 greedily. Two attributes values are matched only if the similarity exceeds a threshold value (we used a threshold of 0.6 for our experiments). Subsequently, the second attribute value in the first tuple is matched against the most similar *unmatched* attribute value in the second tuple and so on. The edges picked by this greedy matching step are shown in solid lines in Figure 2. Agreement between the tuples is calculated as the sum of the similarities of the individual matched values. The two tuples are considered matching if they exceed a threshold similarity of 1.3 (approximately twice the value match threshold above). As this threshold increases the comparison between tuples becomes more and more exact matching.

Agreement Between Answers Sets: After computing the similarity values of tuples, the agreement between two result sets R_{1q} and R_{2q} from two sources for a query q is defined as,

$$A(R_{1q}, R_{2q}) = \arg\max_{M} \sum_{(t_1 \in R_{1q}, t_2 \in R_{2q}) \in M} S(t_1, t_2) \qquad (6)$$

where M is the optimal matched pairs of tuples between R_{1q} and R_{2q} and $S(t_1, t_2)$ is the similarity measure used. Again finding the optimal matching M between the tuples of two results sets is a weighted bipartite matching problem with $O(k^3)$ time complexity at the best, where top-k tuples from each query for agreement calculation. Hence for re-

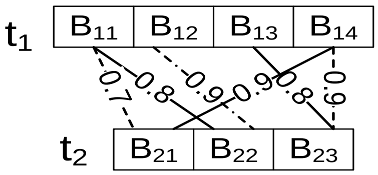


Figure 2: Calculating tuple similarity based on vector of bags model of tuples. The edges in solid line represent the matches picked by the greedy matching algorithm used.

duced time complexity we use a greedy matching similar to the approach used above for finding tuple similarity. First tuple in R_{1q} is matched greedily against the tuple with highest match in R_{2q}. Subsequently, the second tuple in R_{1q} is matched with most similar unmatched tuple in R_{2q} and so on for the entire tuple set. The agreement between two result sets is calculated as the sum of the agreements between the matched tuples. The agreement is calculated for the queries are used in the Equation 4.

We calculate agreement between the top-k (with $k = 5$) answer sets of the each query in the sampling set described in the subsection below. We stick to top-k results since most web information systems focus on providing best answers on few top positions, since users rarely go below top results. The agreements of the answers to the entire set of sampling queries is used in Equation 4 to compute the agreement between the sources. Note that even though we used top-k answers, the normalization against the answer set size in Equation 4 is required, since answer set sizes varies as some sources return less than k results to some queries.

4.2 Sampling

Web databases are typically non-cooperative, i.e. do not share the statistics about the data contained, or allows access to the entire set of data. For sampling, we assume only a form based query interface allowing key word queries; similar to the query based sampling used for the non-cooperative text databases [6]. We used the same sampling method and queries for both online databases and Google Base.

For generating sampling queries, we use the publicly available book and movie listings. We use two hundred queries each from book and movie domain for sampling. To generate queries for book domain, we randomly select 200 books from New York Times yearly number one book listing from the year 1940 to 2007. For the sampling query set of movie domain, we use 200 random movies from second edition of New York Times movie guide.

As key word queries for sampling, we use partial titles of the books/movies. We generate partial title queries by randomly deleting words in titles with a probability of 0.5 from the titles of length more than one word. Use of partial queries are based on the fact that two sources are less likely to agree each other on partial title queries, since there are more number of possible answers for a partial title query than a full title query (since partial titles are less constraining). Hence agreement on answers to partial queries is more indicative of agreement between the sources (our initial experiments validated this assumption).

We preform a query based sampling of database by sending the keyword queries in the sampling set to the title keyword search fields provided by the web databases. Such a simple key word based sampling is used, since many web databases allow only key word based queries. The sampling is automated here, but we wrote our own parsing rules to parse the result tuples from the returned HTML pages. This parsing of tuples has been solved previously [3], and can be automated. For Google Base experiments, the parsing is not required as structured tuples are returned.

5. PERFORMANCE EVALUATION

In this section we evaluate the relevance and trustworthiness of SourceRank for domain specific source selection. The top-k precision and discounted cumulative gain of SourceRank based ranking is compared with (i) Coverage based source selection used in relational databases, (ii) CORI method used in text databases, (iii) results provided by Google Product search on Google Base.

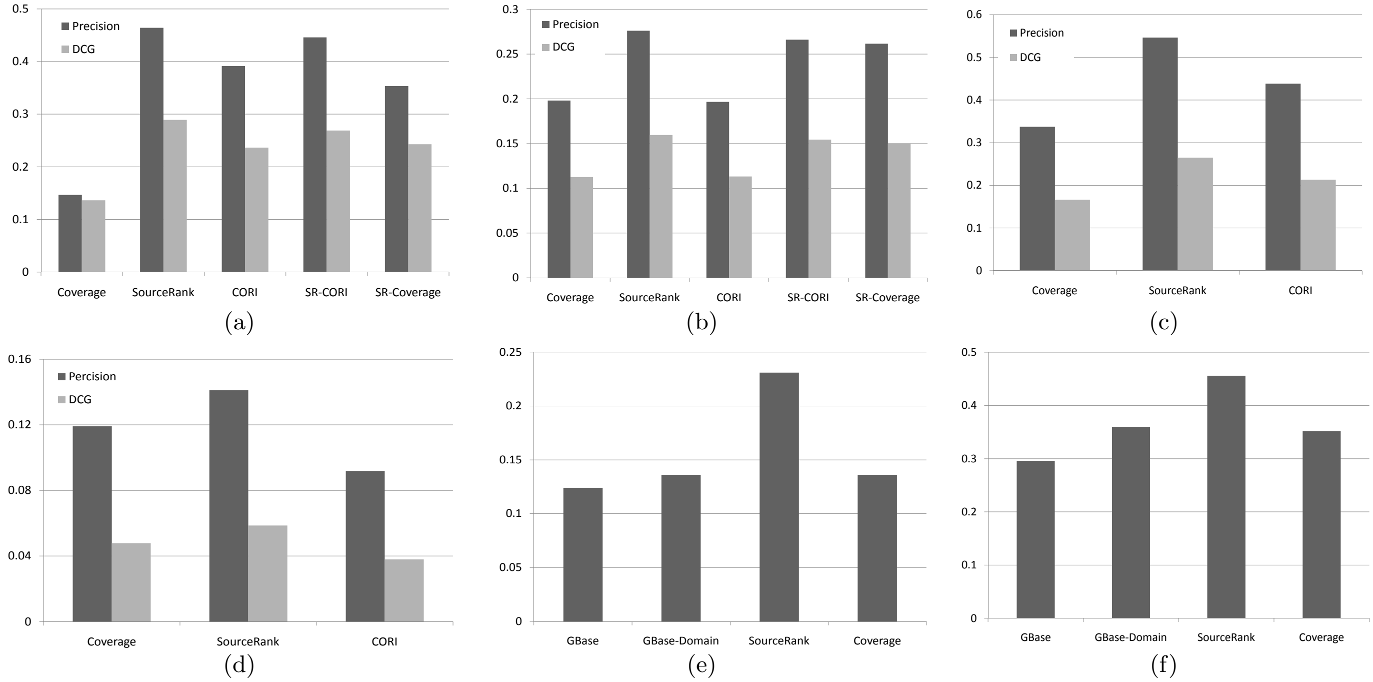
5.1 Experimental Setup

Databases: We performed the evaluations in two vertical domains—book sellers and movies (movies means DVD, Blu-Ray etc). We used two sets of data bases— (i) a set of online data sources accessed by their own web forms; (ii) data from hundreds of sources collected in *Google Base*.

Databases listed in TEL-8 database list in the UIUC deep web interface repository [2]—after removing non-working and databases using post (since post is not idempotent)—are used for online evaluations. We used nineteen movie database and twenty two book database in TEL-8 repository. Also we added five video sharing databases to movie domain and five library sources to book domain. These out of domain sources are added to make the domain less pure and hence to make the ranking difficult. If all sources are mostly of same quality, selecting one or another does not make a difference. Since these out of domain sources share common titles with in-domain sources, distinguishing between them is difficult for a source selection method.

Google Base is a collection of data from large number of web databases, with an API based access to data returning ranked results [1]. Each source in Google Base has a source id. For our experiments, we need sources specific to movies/books domain. For selecting domain sources, we probed the Google Base with a set of ten book/movie names as queries. From the first two pages of results (first 400 results) to each query, we collected source ids of all the sources; and considered them as a source belonging to that particular domain. We used a set of 675 book sources and 209 movie sources thus obtained from Google Base for our evaluations. These lists contain some out-of-domain sources also since some non-domain results may contain matching results to a query. Sampling from these sources are performed through Google Base API's as described in Section 4.2. Considering the difference between the two sets of databases used for evaluation, for Google Base—since the same ranking is applied the entire data—the only source of independence is difference in data. On the other hand, for the online databases both data and ranking may be independent.

Test Query Set: Test query sets for both book and movie domains are selected from different lists than the sampling query set, so that test and sampling sets are disjoint. The movie and books titles in several categories are obtained from a movie sharing site and a favorite books list. We



(a) (b) (c)

(d) (e) (f)

Figure 3: (a-b) Comparison of precision of top-4 **sources selected by Coverage, SourceRank, CORI,** $(0.1 \times SourceRank + 0.9 \times CORI)$**,** $(0.5 \times Coverage + 0.5 \times SourceRank)$ **for (a) movies. (b) books. (c-d) Comparison of source precision and DCG of top-8 source selection for (d) movies (e) books. (e-f) Comparison of top-**5 **precision of results returned by SourceRank, Google Base and Coverage for (e) movies (f) books.**

generated queries by randomly removing words from the movie/book titles with probability of 0.5—in the same way as described for the sampling queries above. We used partial titles as the test queries, since intuitively typical web user queries are partial descriptions of objects.

5.2 Measurements

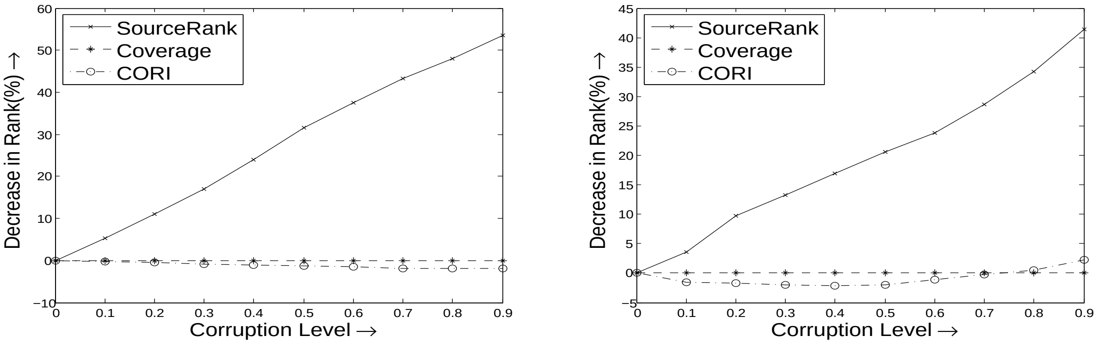
Coverage: Coverage is computed as the mean relevance of the top-5 results to the sampling queries described in Section 4.2 above. For assessing relevance of the results, we used the SoftTF-IDF with Jaro-Winkler similarity between the query and the results (please recall that the same similarity measure is used for the agreement computation).

CORI: For CORI, we used the same parameters as found to be optimal by Callan *et al.* [7]. To collect source statistics for CORI, we used terms with highest document frequency from the sample crawl data describe in Section 4.2 as crawling queries. The highest document frequency terms in related text databases used as queries to crawl is observed to be performing well by Callan *et al.* [6]. We used two hundred queries and used top-10 results for each query to create resource descriptions for CORI. We used comparison with CORI, since later developments like ReDDE [15] depend on database size estimation by sampling, and it is not demonstrated that this size estimation would work on web sources returning top-k ranked results.

5.3 Relevance Evaluation

Assessing Relevance: To assess the relevance, we used randomly chosen queries from test queries described above in Subsection 5.1, and issued the queries to the top-k sources selected by different methods. The results returned are classified as relevant and non-relevant manually. The first author performed the classification of the tuples, since thousands of tuples have to be classified as relevant and non-relevant for each experiment. The classification is simple and almost rule based. For example, assume that the query is *Wild West*, and the original movie name from which the partial query is generated is *Wild Wild West* as described in test query description in Subsection 5.1. If the result tuple refers to the movie *Wild Wild West* (i.e. DVD, Blue Ray or CD etc.) result is classified as relevant, otherwise irrelevant. Similarly for books, if the results is the queried book to sell it is classified as relevant and otherwise classified as irrelevant. As an insurance against classification being biased, we randomly mixed tuples from all methods in a single file; so that the author does not know which method each result came from while he does the classification.

Online Sources: We used twenty five queries for top-4 online source selection and fifteen queries for top-8 source selection for both the domains. We compared mean top-5 precision of top-4 sources, and DCG of top-4 rank list of sources (we avoided normalization in NDCG since rank lists are of equal length). Five methods, namely Coverage, SourceRank, CORI, and two linear combinations of SourceRank with CORI and Coverage—$(0.1 \times SourceRank + 0.9 \times CORI)$ and $(0.5 \times Coverage + 0.5 \times SourceRank)$—are compared. For all the three methods, scores are normalized against the highest score before combining (for example, all the SourceRank scores are normalized against highest SourceRank score). The higher weight for CORI in CORI-SourceRank combinations is to compensate for the higher dispersion of SourceR-



(a) (b)

Figure 4: Decrease in ranks of the sources with increasing source corruption levels for (a) movies and (b) books domain. The SourceRank reduces almost linearly with corruption, while CORI and Coverage are insensitive to the corruption.

ank compared to CORI scores.

The results of the top-4 source selection experiments in movie and books domain are shown in Figure 3(a) and 3(b). For both the domains the SourceRank clearly outperforms the Coverage and CORI. For movie domain, SourceRank increases precision over Coverage by 216.0% (i.e. $\frac{0.464-0.146}{0.146} \times 100$) and over CORI by 22.1%; and DCG of SourceRank is higher by 111.7% and and 22.1% over coverage and CORI respectively. For books domain, SourceRank improves by precision over both CORI and Coverage by approximately 40%; and approximately 41% by DCG. The combinations does not improve performance over SourceRank. This may not be surprising, considering the fact that the sources selected return the results based on relevance. Hence the results from SourceRank only source selection implicitly account for relevance also.

To confirm the results, we compared the three basic methods—SourceRank, Coverage, and CORI—for top-8 source selection. The results are shown in Figure 3(c) and 3(d). For movie domain, SourceRank increases precision by 62.0% and 24.7% over Coverage and CORI respectively; and DCG by 59.4% and 24.3%. For books domain, SourceRank increases precision by 18.0% and 53.7% over Coverage and CORI respectively; and DCG by 22.5% and 54.3%.

Google Base: In these experiments we tested if the precision of Google Base search results can be improved by combining SourceRank with Google Base relevance based tuple ranking. Google Base tuple ranking is applied on top of source selection by SourceRank and compared with stand-alone Google Base Ranking. This combination of source selection with Google Base tuple ranking is required for performance comparison, since source ranking cannot be directly compared with the tuple ranking of Google Base. For books domain, we calculated SourceRank for 675 books domain sources selected as described in Subsection 5.1. Out of these 675 sources, we selected top-67 (10%) sources based on SourceRank. The Google Base is made to query only on this top-67 Sources, and precision of top-5 tuples compared with that of Google Base Ranking without this source selection step. Similarly for movie domain, top-21 sources are selected. The results are evaluated using twenty five testing queries for both the domains. DCG is not computed for these experiments since all the results are ranked by Google Base ranking, hence ranking comparison is not required.

	Coverage	SourceRank
Movies	4.05	4.24
Books	3.98	4.46

Table 1: Mean user ratings of the top-10% sources selected by SourceRank and Coverage. For both the domains SourceRank selects better sources.

In Figure 3(e) and 3(f), the *GBase* is the stand alone Google Base relevance ranking. *GBase-Domain* is the Google Base ranking searching only in the domain sources selected using our query probing (e.g. in Figure 3(f), 675 book sources were selected as described in database description in Subsection 5.1 above). SourceRank and Coverage are Google Base tuple rank applied to the tuples from top-10% sources selected by the SourceRank and Coverage based source selections respectively. Note that for books domain, GBase-Domain and Coverage are performing almost equally, and SourceRank precision exceeds both by 69.8% for movie domain and 26% for book domain.

5.4 Trustworthiness Evaluation

Online Sources: We evaluate the ability of SourceRank to eliminate untrustworthy sources based on the fact that corruption in unspecified attribute manifests as untrustworthy results, where mismatch in specified attributes manifests as irrelevant results—as pointed out in the introduction. Since title is the specified attribute for our queries, we corrupted attributes except the title values of the source crawls for randomly selected sources. Corruption is performed by replacing attribute values with random strings. SourceRank, Coverage and CORI ranks are recomputed using these corrupted crawls and sampling/test queries, and reduction in ranks of the corrupted sources are calculated. The experiment is repeated fifty times for each corruption level, reselecting sources to corrupt randomly for each repetition. The percentage of reduction for a method is computed as the mean reduction in these fifty runs. Since CORI ranking is query specific, decrease in CORI rank is calculated as the average decrease in rank over ten test queries.

The results of the experiments for movies and books do-

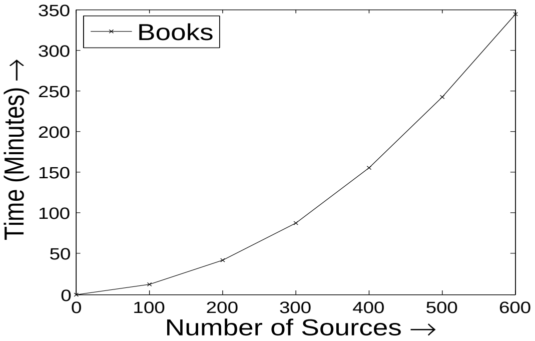


Figure 5: Time to compute agreement against number of sources.

main are shown in Figure 4. The Coverage and CORI are agnostic to the corruption, and do not lower rank the corrupted sources. On the other hand, the SourceRank of corrupted sources reduces almost linear to the corruption level of the source. Any relevance only measure would not be able to capture corruption in unspecified attributes, as we mentioned in the introduction. This corruption-sensitivity of SourceRank would be helpful to solve the trust problems we discussed in the introduction—like solution manual with same title and very low non-existent prices etc.

Google Base: In Google Base the user ratings of sources are used as the measure of source quality. These sources ratings are based on a large number of user reviews collected by Google Base from a number of review portals. The user ratings are not entirely dependent on the source relevance and trust (quantities assessed by SourceRank), and many factors—like shipping time etc—may affect the rating. But intuitively, a source providing irrelevant or untrustworthy results is not likely to be popular with users. So the source trust and relevance is likely to be correlated with the user rating.

The ratings vary from one to five (five is the best). The sources not rated by Google Base i.e. sources having zero or too few reviews are not considered for this evaluation. We did not include the Google Base in comparison since it is not a source ranking method; and that Google Base ranking may be explicitly considering the source user rating. The mean ratings of the top-10% sources selected by SourceRank and Coverage are shown in Table 1. Though the absolute difference between the mean ratings seems low, this difference is significant as dispersion of sources rating is low (e.g. mean absolute deviation of twenty randomly picked movie sources was 0.48).

6. TIMING EXPERIMENTS

For the timing experiments, since we already know that random walk computations are feasible in web scale [5] we are not including timing experiments of random walk, and focus on agreement graph computation.

The agreement computation is $O(n^2k^2)$ where n is the number of sources and top-k result set from each source is used for calculating the agreement graph (k is a constant factor in practice). We performed all experiments on 3.16 GHz, 3.25 GB RAM Intel Desktop PC with Windows XP Operating System.

Figure 5 shows the variation of agreement graph computation for the 600 book sources from Google Base we used. As expected from time complexity formulae above, the time increases in second order polynomial time. Since the time complexity is quadratic, large scale computation of SourceRank should be feasible. Also note that the agreement graph computation is easy to parallelize. The different processing nodes can be assigned to compute a subset of agreement values between the sources; and these agreement values can be computed in isolation—without any inter-process communication to pass intermediate results between the nodes.

7. CONCLUSION AND FUTURE WORK

A compelling holy grail for the information retrieval research is to integrate and search the structured deep web sources. An immediate problem posed by this quest is source selection, i.e. selecting relevant and trustworthy sources to answer a query. Past approaches to this problem depend on purely query based measures to assess the relevance of a source (such as coverage or CORI). The relevance assessment based solely on query similarity is easy to be tampered by the content owner, as the measure is insensitive to the popularity and trustworthiness of the results. 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 relevance sensitive to source popularity and 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. The SourceRank improves relevance sources selected compared to existing methods and effectively removes corrupted sources. Also, we demonstrate that combining SourceRank can be combined with Google Product search ranking significantly improves the quality of the results.

An immediate extension is computing and compensating for dependency between the deep web sources would improve the agreement graph computation as well as prevent the collusion between the sources. We are working on extending the current work on finding dependence between the sources. Also we are exploring calculation of source dependence based on answers to queries with low specificity (queries with large number of possible answers). Another extension for SourceRank is combining with a query specific relevance measure, especially to apply for domain-independent search. Also similar to other popularity based methods like page rank, SourceRank may have tendencies like suppressing useful but unique answers and reducing diversity of results; and may require combination with other measures to compensate these effects. Some other possible applications areas of SourceRank may be XML search and RDF source selection, since these data formats are structured to facilitate the computation of agreement.

8. REFERENCES

[1] Goolge products. http://www.google.com/products.

[2] UIUC TEL-8 web interface repository. http://metaquerier.cs.uiuc.edu/repository/datasets/tel-8/index.html.

[3] A. Arasu and H. Garcia-Molina. Extracting structured data from Web pages. In *Proceedings of SIGMOD*, pages 337–348, 2003.

[4] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. Improving collection selection with overlap awareness in P2P search engines. *SIGIR*, pages 67–74, 2005.

[5] 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.

[6] J. Callan and M. Connell. Query-based sampling of text databases. *ACM TOIS*, 19(2):97–130, 2001.

[7] J. Callan, Z. Lu, and W. Croft. Searching distributed collections with inference networks. In *Proceedings of ACM SIGIR*, pages 21–28. ACM, NY, USA, 1995.

[8] W. Cohen. Integration of heterogeneous databases without common domains using queries based on textual similarity. *ACM SIGMOD Record*, 27(2):201–212, 1998.

[9] W. Cohen, P. Ravikumar, and S. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IIWeb Workshop*, 2003.

[10] X. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. In *PVLDB*, 2009.

[11] J. Lee. Analyses of multiple evidence combination. In *ACM SIGIR Forum*, volume 31, page 276. ACM, 1997.

[12] 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.

[13] Z. Nie and S. Kambhampati. A Frequency-based Approach for Mining Coverage Statistics in Data Integration. *Proceedings of ICDE*, page 387, 2004.

[14] M. Shokouhi and J. Zobel. Federated text retrieval from uncooperative overlapped collections. In *Proceedings of the ACM SIGIR*. ACM, 2007.

[15] L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In *Proceedings of ACM SIGIR*, pages 298–305, 2003.

[16] A. Wright. Searching the deep web. *Commmunications of ACM*, 2008.

[17] X. Yin, J. Han, and P. Yu. Truth discovery with multiple conflicting information providers on the web. *IEEE TKDE*, 20(6):796–808, 2008.