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

Raju Balakrishnan, and Subbarao Kambhampati [*]
Computer Science and Engineering, Arizona State University
Tempe AZ USA 85287
rajub@asu.edu, rao@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 database selection methods (both text and relational) assess the source quality based on the query-similarity-based relevance assessment. When applied to the deep web these methods have two deficiencies. First is that the methods are agnostic to the correctness (trustworthiness) of the sources. Secondly, the query based relevance does not consider the importance of the results. These two considerations are essential for the open collections like the deep web. Since a number of sources provide answers to any query, we conjuncture that the agreements between these answers are likely to be helpful in assessing the importance and the trustworthiness of the sources. We compute the agreement between the sources as the agreement of the answers returned. While computing the agreement, we also measure and compensate for possible *collusion* between the sources. This adjusted agreement is modeled as a graph with sources at the vertices. On this agreement graph, a quality score of a source that we call *SourceRank*, is calculated as the stationary visit probability of a random walk. We evaluate SourceRank in multiple domains, including sources in Google Base, with sizes up to 675 sources. We demonstrate that the SourceRank tracks source corruption. Further, our relevance evaluations show that SourceRank improves precision by 22-60% over the Google Base and other baseline methods. SourceRank has been implemented in a system called *Factal*.

Categories and Subject Descriptors

H.3.5 [**INFORMATION STORAGE AND RETRIEVAL**]: Online Information Services—*Web-based services*

General Terms

Algorithms, Experimentation

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—popularly referred to as the deep web—is estimated to be in tens of millions [25]. Searching the deep web has been identified as the next big challenge in information management [30]. 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 best subset of sources for answering the query.

Although source selection problem received some attention in the context of text and relational databases (c.f. [26, 9, 13, 27, 21]) existing approaches are focused on assessing the relevance of a source based on local measures of similarity between the query and the answers expected from the source. In the context of deep web, such a purely source-local approach has two important deficiencies:

1. Query based relevance assessment is insensitive to the importance of the source results. For example, the query *godfather* matches the classic movie *The Godfather* as well as the little known movie *Little Godfather*. Intuitively, most users are likely to be looking for the classic movie.

2. The source selection is agnostic to the trustworthiness of the answers. Trustworthiness is a measure of correctness of the answer (in contrast to relevance, which assesses whether a tuple is answering the query, not the correctness of the information). For example, for the query *The Godfather*, many databases in Google Base return copies of the book with unrealistically low prices to attract the user attention. When the user proceeds towards the checkout, these low priced items would turn out to be either out of stock or a different item with the same title and cover (e.g. solution manual of the text book).

A global measure of trust and importance is particularly important for uncontrolled collections like the deep web, since sources try to artificially boost their rankings. A global relevance measure should consider popularity of a result, as the popular results tend to be relevant. Moreover, it is imprudent to measure trustworthiness of sources based on local measures; since the measure of trustworthiness of a source

[*] This research is supported by ONR grant N000140910032 and two Google research awards.

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

should not depend on any information the source provides about itself. In general, the trustworthiness of a particular source has to be evaluated in terms of the endorsement of the source by other sources.

Result Agreement as Implicit Endorsement: Given that the source selection challenges are similar in a way to "page" selection challenges on the web, an initial idea is to adapt a hyper-link based methods like PageRank [11] or authorities and hubs [22] from the surface web. However, the hyper-link based endorsement is not directly applicable to the web databases since there are no explicit links across records. To overcome this problem, we create an implicit endorsement structure between the sources based on the *agreement* between the results. Two sources agree with each other if they return the same records in answer to the same query. It is easy to see that this agreement based analysis will solve the result importance and source trust problems mentioned above. Result importance is handled by the fact that the important results are likely to be returned by a larger number of sources. For example, the classic *Godfather* movie is returned by hundreds of sources while the *Little Godfather* is returned by less than ten sources on a Google Products search [1]. A global relevance assessment based on the agreement of the results would thus have ranked the classic Godfather high. Similarly, regarding trust, the corruption of results can be captured by an agreement based method, since other legitimate sources answering the same query are likely to disagree with the incorrect results (e.g. disagree with unrealistically low price of the book result). We provide a formal explanation for why agreement implies trust and relevance in Subsection 3.1 below.

Challenges in Computing Result Agreement: Agreement computation between the web databases poses multiple challenges that necessitate combination and extension of methods from relational and text databases. The primary challenge is that different web databases may represent the same entity syntactically differently, making the agreement computation hard [14]. To solve this problem, we combine record linkage models with entity matching techniques for accurate and speedy agreement computation. Further, attributes matchings are weighted by the computed attribute importance. Another challenge in computing agreement is that most web databases are *non-cooperative*—i.e. they do not allow access to full data or source statistics. Instead, access is limited to retrieving a set of top-k answers to a simple key word query. To handle this, we adapt query based sampling methods used for text databases [12].

Combating Source Collusion: Like PageRank, databases may enhance SourceRank by colluding with each other. Differentiating genuine agreement between the sources from the collusion increases the robustness of the SourceRank. We devise a method to detect the source dependence based on answers to the *"large answer"* queries. A large answer query is a very general keyword like *"DVD"* or *"director"* with a large set of possible answers. If two sources always return the same answers to these type of queries, they are likely to be dependent (colluding). We expand on this intuition to measure and compensate for the source collusion while calculating the agreement.

Implementation and Evaluation: We implemented the SourceRank based source selection in a system called *Factal*, which may be accessed at `http://factal.eas.asu.edu/` (details of the system are given in [8]). To compare the performance, we evaluated the ability of SourceRank to select trustworthy and relevant sources over two sets of web sources—(i) sets of books and movie databases in TEL-8 repository [5] and (ii) books and movie sources from Google Base [1]. Our evaluation shows that SourceRank improves the relevance of source selection by 22-60% over the existing methods. We also show that the SourceRank combined with the default Google Base result ranking improves the top$-k$ precision of results by 23-50% over stand-alone Google Base. Trustworthiness of source selection is evaluated as the ability to remove sources with corrupted attributes. Our experiments show that the SourceRank diminishes almost linearly with the source corruption.

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 method to calculate trustworthiness of deep web sources. (iii) Domain independent computation of the agreement between the deep web databases. (iv) A method for detecting collusion between the web databases, and (v) Formal evaluations on large sets of sources.

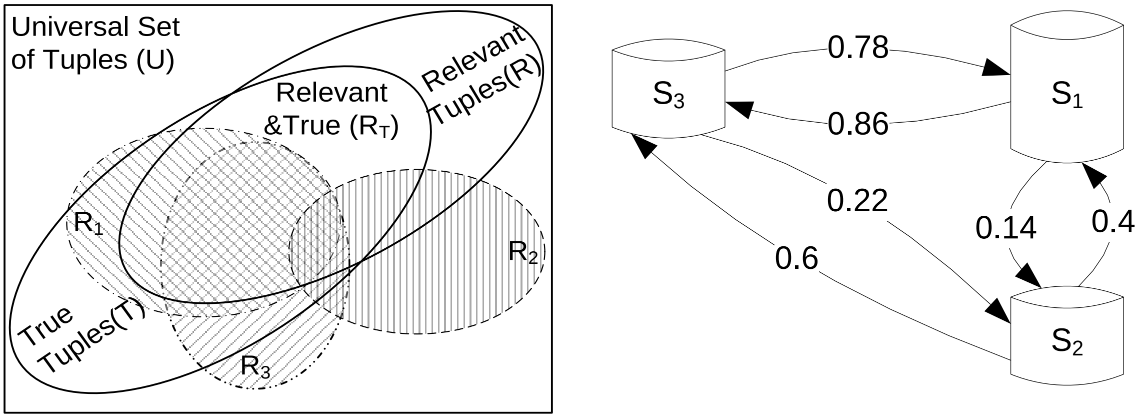
The rest of this paper is organized as follows. Next section discusses the related work. Section 3 provides a formal justification for calculating source reputation based on the agreement of sources, and presents the SourceRank calculation method. The following section explains the computation of agreement between the sources, and describes how sources are sampled to get the seed results for supporting agreement computation. Next, we explain our collusion detection method. In Section 6, SourceRank is applied to multiple domains and types of sources to demonstrate the improved relevance and trustworthiness. We also evaluate source collusion detection and the time to calculate the SourceRank.

2. RELATED WORK

The indispensability and difficulty of source selection for the deep web has been recognized previously [25]. Current relational database selection methods minimize cost by retrieving maximum number of distinct records from minimum number of sources [26]. Cost based web database selection is formulated as selecting the least number of databases maximizing number of relevant tuples (coverage). The related problem of collecting source statistics [26, 21] has also been researched. The problem of ranking database tuples for key word search is addressed [10].

Considering research in the text databases selection, Callan *et al.* [13] formulated a method called CORI for query specific selection based on relevance. Cooperative and non-cooperative text database sampling [12, 21] and selection considering coverage and overlap to minimize cost [28, 27] are addressed by a number of researchers.

Combining multiple retrieval methods for text documents has been used for improved accuracy [16]. Lee [24] observes that the different methods are likely to agree on the same relevant documents than on irrelevant documents. This observation rhymes with our argument in Section 3 in giving a basis for agreement-based relevance assessment. For the surface web, Gyöngyi *et al.* [20] proposed trust rank, an extension of page rank considering trustworthiness of hyper-linked pages. Agrawal *et al.* [6] explored ranking database search records by comparing to corresponding web search results.



(a) (b)

Figure 1: (a) Model for explaining why the agreement implies trust and relevance. Universal set U is the search space, R_T is the intersection of trustworthy tuple set T and relevant tuple set R (R_T is unknown). R_1, R_2 and R_3 are the result sets of three sources. (b) A sample agreement graph structure of three sources. The weight of the edge from S_i to S_j is computed by Equation 5.

A probabilistic framework for trust assessment based on agreement of web pages for question answering has been presented by Yin *et al.* [31]. Their framework however does not consider the influence of relevance on agreement, multiple correct answers to a query, record linkage and non-cooperative sources; thus limiting its usability for the deep web. Dong *et al.* [18, 17] extended this model considering source dependence using the same basic model as Yin *et al.* As we shall see, the collusion detection in the deep web needs to address different constraints like multiple true values, non-cooperative sources, and ranked answer sets.

3. SOURCERANK: TRUST AND RELEVANCE RANKING OF SOURCES

In this section we formalize the argument that the relevance and trustworthiness of a source manifests as the agreement of its results with those from other sources. We also explain the 2-step SourceRank calculation process: (i) creating a source graph based on agreement between the sources (ii) assessing the source reputation based on this source graph.

3.1 Agreement as Endorsement

In this section we show that the result set agreement is an implicit form of endorsement. In Figure 1(a) let R_T be the set of relevant and trustworthy tuples for a query, and U be the search space (the universal set of tuples searched). Let r_1 and r_2 be two tuples independently picked by two sources from R_T (i.e. they are relevant and trustworthy), and $P_A(r_1, r_2)$ be the probability of agreement of the tuples (for now think of "agreement" of tuples in terms of high degree of similarity; we shall look at the specific way agreement between tuples is measured in Section 4).

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

Similarly let f_1 and f_2 be two irrelevant (or untrustworthy) tuples picked by two sources and $P_A(f_1, f_2)$ be the agreement probability of these two tuples. Since f_1 and f_2 are from $U - R_T$

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

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 in Equation 1 and 2 implies

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

For example, assume that the user issues the query *Godfather* for the Godfather movie trilogy. Three movies in the trilogy— *The Godfather I*, *II* and *III*—are thus the results relevant to the user. Let us assume that the 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 the probability of three sources agreeing are $\frac{1}{9}$ and $\frac{1}{10^8}$ for relevant and irrelevant results respectively.

Let us now extend this argument for answer sets from two sources. In Figure 1(a) R_1, R_2 and R_3 are the result sets returned by three independent sources. The result sets are best effort estimates of R_T (assuming a good number of genuine sources). Typically the results sets from individual sources would contain a fraction of relevant and trustworthy tuples from R_T, and a fraction of irrelevant tuples from $U - R_T$. By the argument in the preceding paragraph, tuples from R_T are likely to agree with much higher probability than tuples from $U - R_T$. This implies that the more relevant tuples a source returns, the more likely that other sources agree with its results.

Though the explanation above assumes independent sources, it holds for partially dependent sources as well. However, the ratio of two probabilities (i.e. the ratio of probability in Equation 1 to Equation 2) will be smaller than that for the independent sources. For added robustness of the SourceRank against source dependence, in Section 5 we assess and compensate for the collusion between the sources.

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(b). In this graph, the vertices represent the sources, and weighted edges represent the agreement between the sources. The edge weights correspond 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. Let $a = A(R_1, R_2)$ be the agreement between the results sets (calculated as described in Section 4). In the 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 link from S_1 to S_2 is that S_1 endorses S_2, where the fraction of tuples endorsed in S_2 is equal to the weight. Since the endorsement weights are equal to the fraction of tuples, rather than the absolute number, they are asymmetric.

As we shall see in Section 4, the agreement weights are estimated based on the results to a set of sample queries. To account for the "sampling bias" in addition to the agreement links described above, we also add "*smoothing links*" with small weights between every pair of vertices. Adding this smoothing probability, the overall weight $w(S_1 \rightarrow S_2)$ of the link from S_1 to S_2 is:

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

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

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. We set β at 0.1 for our experiments. Empirical studies like Gleich *et al.* [19] may help more accurate estimation. These smoothing links strongly connect agreement graph (we shall see that strong connectivity is important for the convergence of SourceRank calculation). 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 allows us to interpret the edge weights as the transition probabilities for the random walk computations.

3.3 Calculating SourceRank

Let us start by considering certain desiderata that a reasonable measure of reputation defined with respect to the agreement graph must satisfy:

1. Nodes with high in-degree should get higher rank—since high in-degree sources are endorsed by a large number of sources, they are likely to be more trustworthy and relevant.

2. Endorsement from a source with a high in-degree should be more respected than endorsed from a source having smaller in-degree. Since a highly endorsed source is likely to be more relevant and trustworthy, the source endorsed by a highly endorsed source is also likely to be of high quality.

The agreement graph described above provides important guidance in selecting relevant and trustworthy sources. Any source that has a high degree of endorsement by other relevant sources is itself a relevant and trustworthy source. This transitive propagation of source relevance (trustworthiness) through agreement links can be captured in terms of a fixed point computation [11]. In particular, if we view the agreement graph as a markov chain, with sources as the states, and the weights on agreement edges specifying the probabilities of transition from one state to another, then the asymptotic stationary visit probabilities of the markov random walk will correspond to a measure of the global relevance of that source. We call this measure *SourceRank*.

The markov random walk based ranking does satisfy the two desiderata described above. The graph is strongly connected and irreducible, hence the random walk is guaranteed to converge to the unique stationary visit probabilities for every node. This stationary visit probability of a a node is used as the SourceRank of that source.

4. AGREEMENT COMPUTATION AND SAMPLING

If the sources are fully relational and share the same schema, then agreement between two tuples will reduce to equality between them. On the other extreme, if the sources are text databases then the agreement between two items will have to be measured in terms of textual similarity. Deep web sources present an interesting middle ground between the free-text sources in IR, and the fully structured sources in relational databases. Hence to address challenges in agreement computation of deep web results we have to combine and extend methods from both these disciplines. In the following subsection, we will describe agreement computation and sampling sources to compute agreement.

4.1 Computing Agreement

Computing agreement between the sources involves following three levels of similarity computations: (a) attribute value similarity (b) tuple similarity, and (c) result set similarity.

(a) Attribute value similarity: If the different web databases were using common domains[1] for the names, calculating agreement between the databases is trivial. But unfortunately, assumption of common domains rarely holds in web databases [14]. For example, the title and casting attributes of tuples referring to the same movie returned from two databases are shown in Table 1(a) and 1(b). Identifying the semantic similarity between these tuples is not straightforward, since the titles and actor lists show wide syntactic variation.

The textual similarity measures work best for scenarios involving web databases with no common domains [14]. Since this 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 [15]. SoftTF-IDF measure is similar to the normal TF-IDF measure. But instead of considering only exact same words in two documents to calculate similarity, SoftTF-IDF also considers occurrences of similar words.

Formally, let v_i and v_j be the values compared, and $\mathcal{C}(\theta, v_i, v_j)$ be the set of words for $w \in v_i$ such that there is some $u \in v_j$ with $sim(w, u) > \theta$. Let $D(w, v_j) = max_{u \in v_j} sim(w, u)$. The $\mathcal{V}(w, v_i)$ are the normal TF values weighted by $log(IDF)$

[1] common domains means names referring to the same entity are the same for all the databases, or can be easily mapped to each other by normalization

(a)

	Title	Casting
1	Godfather, The: The Coppola Restoration	James Caan / Marlon Brando more

(b)

	Title	Casting
1	The Godfather - The Coppola Restoration Giftset [Blu-ray]	Marlon Brando, Al Pacino

Table 1: Sample tuples returned by two movies databases to the query *Godfather* are shown in Table (a) (tuples from first source) and (b) (tuples from second source).

used in the basic TF-IDF. SoftTFIDF is calculated as,

$$\mathcal{SIM}(v_i, v_j) = \sum_{w \in \mathcal{C}(\theta, v_i, v_j)} \mathcal{V}(w, v_i)\mathcal{V}(u, v_j)D(w, v_j) \tag{6}$$

We used Jaro-Winkler as a secondary distance function *sim* above with an empirically determined $\theta = 0.6$. Comparative studies show that this combination provides best performance for name matching [15]. For pure numerical values (like price) we calculate similarity as the ratio of the difference of values to the maximum of the two values.

(b) Tuple similarity: The tuples are modeled as a vector of bags [14]. 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, then the similarity between the tuples is simply the sum of the similarities between the matching values. The problem of finding this mapping is the well known automated answer schema mapping problem in web databases [29]. We do not assume predefined answer schema mapping, and hence reconstruct the schema mapping based on the attribute value similarities as described below.

The complexity of similarity computation between the attribute values (i.e. building edges and weights in Figure 2) of two tuples t_1 and t_2 is $O(|t_1||t_2|)$ (this is equal to the number of attribute value comparisons required). After computing these edges, a single attribute value in t_1 may be similar to multiple attributes in t_2 and *vice versa*. The optimal matching should pick the edges (matches) such that the sum of the matched edge weights would be maximum.

$$S_{opt}(t, t') = \arg\max_{M} \sum_{(v_i \in t, v_2 \in t') \in M} \mathcal{SIM}(v_1, v_2) \tag{7}$$

Note that this problem is isomorphic to the well known "*maximum weighted bipartite matching problem*". The Hungarian algorithm gives the lowest time complexity for the maximum matching problem, and is $O(V^2 log(V) + VE)$ (in the context of our agreement calculation problem, V is the number attribute values to be matched, and E is the number of similarity values). Since E is $O(V^2)$ for our problem the overall time complexity is $O(V^3)$.

Running time is an important factor for calculating agreement at the web scale. Considering this, instead of the $O(V^3)$ optimal matching discussed above, we use the $O(V^2)$ greedy matching algorithm as a reasonable balance between time complexity and performance. To match tuples, say t_1 and t_2 in Figure 2, the first attribute value of t_1 is greedily matched against the most similar attribute value of t_2. Two attributes values are matched only if the similarity exceeds a threshold value (we used an empirically determined threshold of 0.6 in 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 selected by this greedy matching step are shown in solid lines in Figure 2. The 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 empirically determined threshold of similarity.

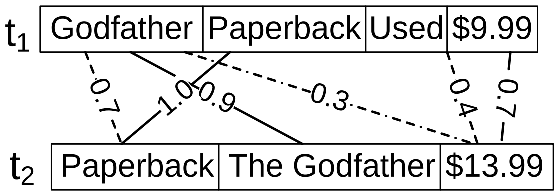


Figure 2: Example tuple similarity calculation. The dotted line edges denote the similarities computed, and the solid edges represent the matches picked by the greedy matching algorithm.

The Fellagi-Saunter record linkage model [23] suggests that the attribute values occurring less frequently are more indicative of the semantic similarity between the tuples. For example, two entities with the common title *The Godfather* are more likely to be denoting same book than two entities with common format *paperback*). To account for this, we weight the similarities between the matched attributes in the step above as

$$S(t, t') = \frac{\sum_{v_i, v_j \in M} w_{ij} \mathcal{SIM}(v_i, v_j)}{\sqrt{\sum_{v_i, v_j \in M} w_{ij}^2}} \tag{8}$$

where v_i,v_j are attribute values of t and t' respectively, and $w_{i,j}$ is the weight assigned to the match between v_i and v_j based on the mean inverse document frequency of the tokens in v_i and v_j. Specifically, the w_{ij}'s are calculated as,

$$w_{ij} = log\left(\frac{\sum_k \mathcal{IDF}_{ik}}{|v_i|}\right) log\left(\frac{\sum_l \mathcal{IDF}_{jl}}{|v_j|}\right) \tag{9}$$

where v_i is the i^{th} attribute value and $\mathcal{IDF}_{ik}$ is the inverse document frequency of the k^{th} token of the i^{th} attribute value. This is similar to the weighting of terms in TFIDF.

(c) Result Set Similarity: The agreement between two result sets R_{1q} and R_{2q} from two sources for a query q is defined as,

$$A(R_{1q}, R_{2q}) = \arg\max_{M} \sum_{(t \in R_{1q}, t' \in R_{2q}) \in M} S(t, t') \tag{10}$$

where M is the optimal matched pairs of tuples between R_{1q} and R_{2q} and $S(t, t')$ are as calculated in Equation 8. Since this is again a bipartite matching problem similar to Equation 7, we use a greedy matching. The 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 the most similar unmatched tuple in R_{2q} and so on. The agreement between the two result sets is calculated as the sum of the agreements between the matched tuples. The agreement thus calculated is used in the Equation 4.

We calculate agreement between the top-k (with $k = 5$) answer sets of the each query in the sampled set described in the subsection below. We stick to top-k results since most web information systems focus on providing best answers in the top few positions (a reasonable strategy given that the users rarely go below the top few 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 the answer set sizes vary as some sources return less than k results to some queries.

4.2 Sampling Sources

Web databases are typically non-cooperative, i.e. they do not share the statistics about the data they contain, or allow access to the entire set of data. Thus, the agreement graph must be computed over a sampled set. In this section we describe the sampling strategy used for our experiments on web databases (see Section 6). 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 [12].

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 the book domain, we randomly select 200 books from the New York Times yearly number one book listing from the year 1940 to 2007 [3]. For the sampling query set of movie domain, we use 200 random movies from the second edition of New York Times movie guide [4].

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

We perform a query based sampling of database by sending the queries to the title keyword search fields of the sources. 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 [7], and can be automated. (parsing is not required for Google Base experiments as structured tuples are returned.)

5. ASSESSING SOURCE COLLUSION

We measure the collusion of web databases on top-k answer sets, since agreement is also computed on top-k answers. Two issues that complicate collusion detection are (i) even non-colluding databases in the same domain may contain almost the same data. For example, many movie sources may contain all Hollywood movies. (ii) top-k answers from even non-colluding databases in the same domain are likely to be similar. For example, two movie databases are likely to return all three movies in Godfather trilogy for the query *Godfather*. The collusion measure should not classify these genuine data and ranking correlations as collusion. On the other hand, mirrors or near-mirrors with same data and ranking functions need to be identified.

The basic intuition we use for collusion detection is that if two sources return the same top-k answers to the queries with large number of possible answers (e.g. queries containing only stop words), they are possibly colluding. More formally, for two ranked sets of answers, the expected agreement between top-k answers $E(A_k)$ is

$$E(A_k) = \begin{cases} \frac{k}{n}(1-e) & \text{if } k < n \\ (1-e) & \text{otherwise} \end{cases} \tag{11}$$

where top-k answers are used to calculate agreement, size of the answer set is n, and e is the error rate due to approximate matching. This means that for queries with large number of answers (i.e. $n \gg k$ as k is fixed) the expected agreement between two independent sources is very low. As a corollary, if the agreement between two sources on a large answer query is high, they are likely to be colluding.

To generate a set of queries with large answer sets, we fetched a set of two hundred keywords with highest document frequencies from the crawl described in the Section 4.2. Sources are probed with these queries. The agreement between the answer sets are computed based on this crawl according to Equation 4. These agreements are seen as a measure of the collusion between the sources. The agreement computed between the same two sources on the samples based on genuine queries described in Section 4.2 is multiplied by $(1 - collusion)$ to get the adjusted agreement. These adjusted agreements are used for computing SourceRank for the experiments below. We also provide a standalone evaluation of collusion measure in Section 6.5.

6. PERFORMANCE EVALUATION

The methods described are implemented as a system namely *Factal* (URL: http://factal.eas.asu.edu/); the system details may be found in [8]. In this section we evaluate the effectiveness of SourceRank (computed based on collusion adjusted-agreement) as the basis for domain specific source selection that is sensitive to relevance and trustworthiness of sources. The top-k precision and discounted cumulative gain (DCG) of SourceRank-base source selection is compared with three existing methods: (i) Coverage based ranking used in relational databases, (ii) CORI ranking used in text databases, and (iii) Google Product search on Google Base.

6.1 Experimental Setup

Databases: We performed the evaluations in two vertical domains—sellers of books and movies (movies include DVD, Blu-Ray etc.). We used three sets of data bases— (i) a set of stand-alone online data sources (ii) hundreds of data sources collected via *Google Base* (iii) a million IMDB records [2].

The databases listed in TEL-8 database list in the UIUC deep web interface repository [5] are used for online evaluations (we remove non-working sources). We used sixteen movie databases and seventeen book databases from the TEL-8 repository. In addition to these, we added five video sharing databases to the movie domain and five library

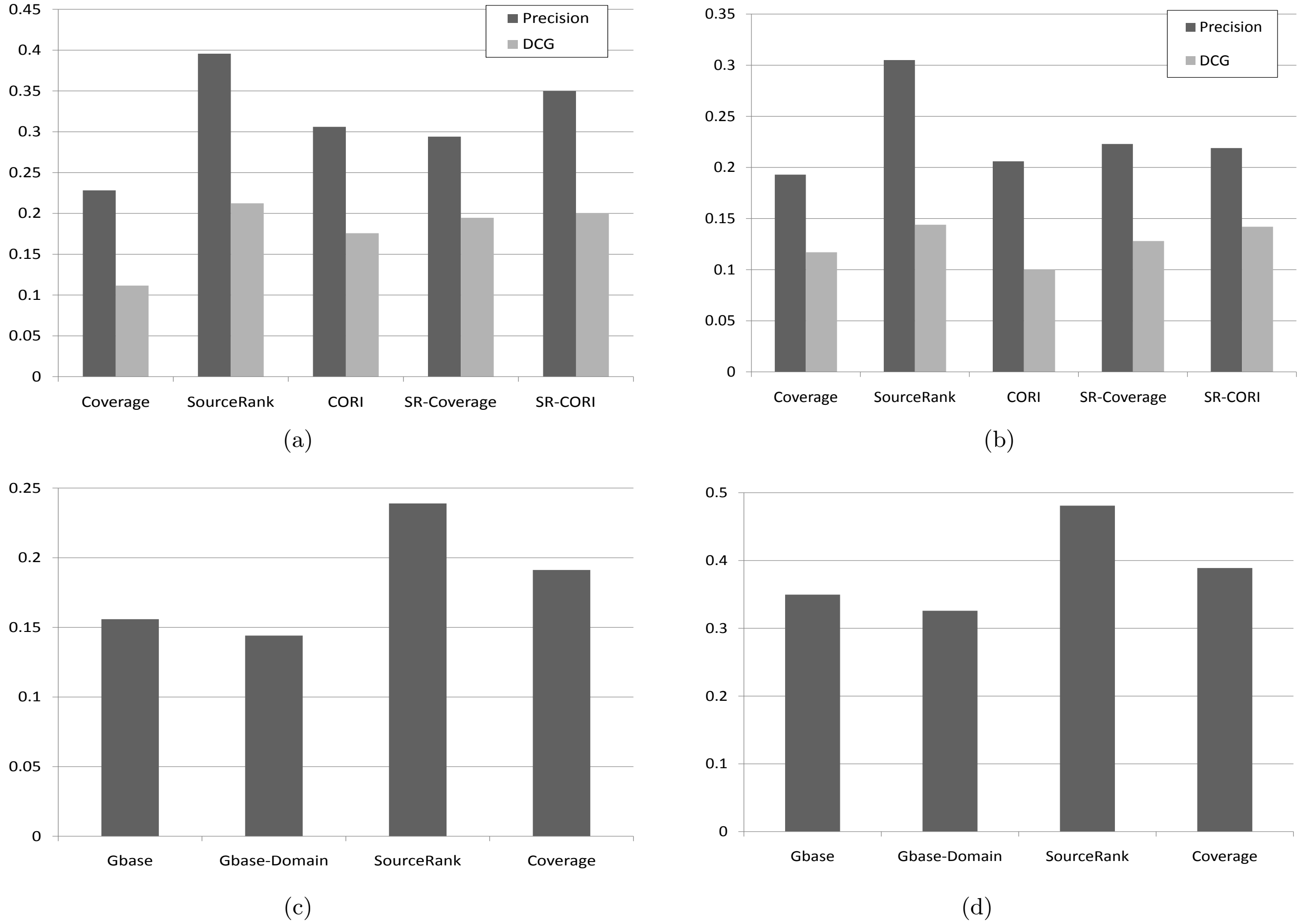


Figure 3: (a-b) Comparison of precision and DCG of top-4 online sources selected by Coverage, SourceRank, CORI, Combination of SourceRank with Coverage (SR-Coverage) and CORI (SR-CORI) for movies (figure a) and books (figure b). (c-d) Comparison of top-5 precision of results returned by SourceRank, Google Base and Coverage for movies (figure c) and books (figure d).

sources to the book domain. These out-of-domain sources are added to increase the variance in source quality. If all sources are of similar quality, different rankings do not make a difference.

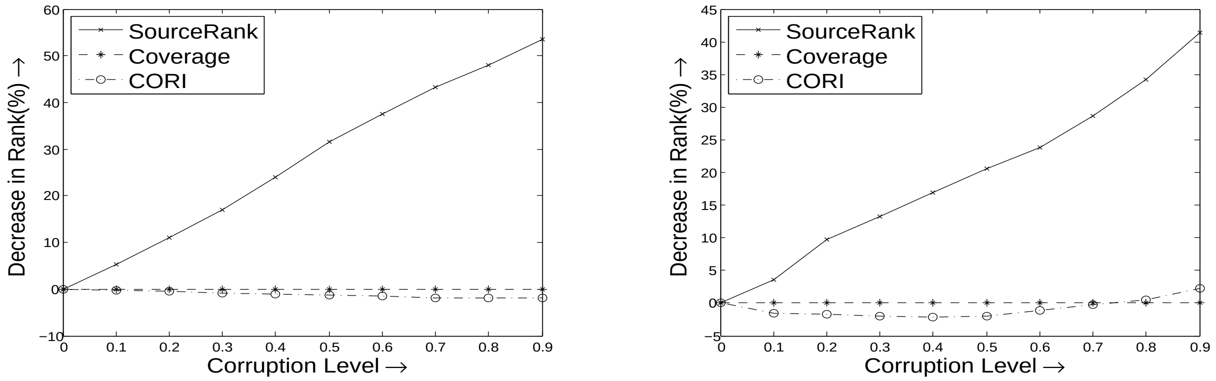
Google Base is a collection of data from a large number of web databases, with an API-based access to data returning ranked results [1]. The Google Products Search works on Google Base. Each source in Google Base has a source id. For selecting domain sources, we probed the Google Base with a set of ten book/movie titles as queries. From the first 400 results to each query, we collected source ids; and considered them as a source belonging to that particular domain. This way, we collected a set of 675 book sources and 209 movie sources for our evaluations. Sampling is performed through Google Base API's as described in Section 4.2.

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 book titles in several categories are obtained from a movie sharing site and a favorite books list. We 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 typical web user queries are partial descriptions of objects. The number of queries are used in different experiments varies between 50 to 80, so as to attain 95% confidence levels.

6.2 Baseline Methods

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 the relevance of the results, we used the SoftTF-IDF with Jaro-Winkler similarity between the query and the results (recall that the same similarity measure is used for the agreement computation).

CORI: To collect source statistics for CORI [13], we used terms with highest document frequency from the sample crawl data describe in Section 4.2 as crawling queries. Callan *et al.* [12] observe that good performance is obtained by using highest document frequency terms in related text databases as queries to crawl. Similarly, we used two hundred high tuple-frequency queries and used top-10 results for each query to create resource descriptions for CORI. We used the same parameters as found to be optimal by Callan *et al.* [13].



(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.

CORI is used as the baseline, since the later developments like ReDDE [28] depend on database size estimation by sampling, and it is not demonstrated that this size estimation would work on the ranked results from web sources.

6.3 Relevance Evaluation

Assessing Relevance: To assess the relevance, we used randomly chosen queries from test queries described above in Section 6.1. These queries are issued to the top-k sources selected by different methods. The results returned are manually classified as relevant and non-relevant. The first author performed the classification of the tuples, since around 14,000 tuples were to be classified as relevant and irrelevant. 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 the test query description in Section 6.1). If the result tuple refers to the movie *Wild Wild West* (i.e. DVD, Blu-Ray etc. of the movie), then the result is classified as relevant, otherwise it is classified to be irrelevant. Similarly for books, if the result is the queried book to sell, it is classified as relevant and otherwise it is classified as irrelevant. As an insurance against biased classification by the author, we randomly mixed tuples from all methods in a single file; so that the author did not know which method each result came from while he does the classification. All the evaluations are performed to differentiate SourceRank precision and DCG from competing methods by non-overlapping confidence intervals at a significance level of 95% or more.

Online Sources: We compared mean top-5 precision and DCG of top-4 Sources (we avoided normalization in NDCG since ranked 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. The higher weight for CORI in CORI-SourceRank combination is to compensate for the higher dispersion of SourceRank compared to CORI.

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, SourceRank clearly outperforms the Coverage and CORI. For the movie domain, SourceRank increases precision over Coverage by 73.0% (i.e. $\frac{0.395-0.228}{0.228} \times 100$) and over CORI by 29.3%. DCG of SourceRank is higher by 90.4% and and 20.8% over Coverage and CORI respectively. For the books domain, SourceRank improves both precision and DCG over CORI as well as Coverage by approximately 30%. The SourceRank outperforms stand-alone CORI and Coverage in both precision and DCG at a confidence level of 95%. Though the primary target of the evaluation is not differentiating SourceRank and combinations, it may be worth mentioning that SourceRank outperforms the combinations at a confidence level more than 90% in most cases. This is not surprising, since the sources selected return the results based on query based relevance. Hence the results from SourceRank-only source selection implicitly account for the query similarity (keep in mind that CORI and Coverage select sources based on query relevance).

As a note on the seemingly low precision values, these are mean relevance of the top-5 results. Many of the queries used have less than five possible relevant answers (e.g. a book title query may have only paperback and hard cover for the book as relevant answers). But since the web databases always tend to return full first page of results average top-5 precision is bound to be low.

Google Base: In these experiments we tested if the precision of Google Base search results can be improved by combining SourceRank with the default Google Base relevance 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 is required for performance comparison, since source ranking cannot be directly compared with the tuple ranking of Google Base. For the book domain, we calculated SourceRank for 675 book domain sources selected as described in Section 6.1. Out of these 675 sources, we selected top-67 (10%) sources based on SourceRank. Google Base is made to query only on this top-67 Sources, and the precision of top-5 tuples is compared with that of Google Base Ranking without this source selection step. Similarly for the movie domain, top-21 sources are

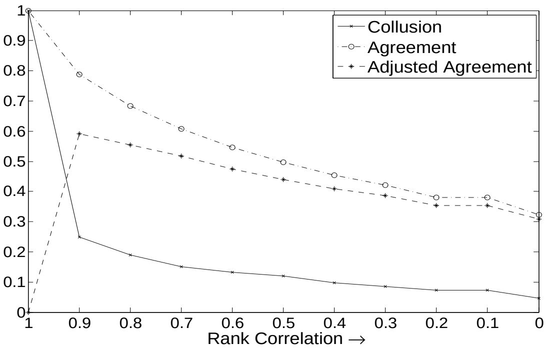


Figure 5: Variation of Collusion, Agreement and Adjusted Agreement with rank correlations. Adjusted Agreement is $Agreement \times (1 - collusion)$.

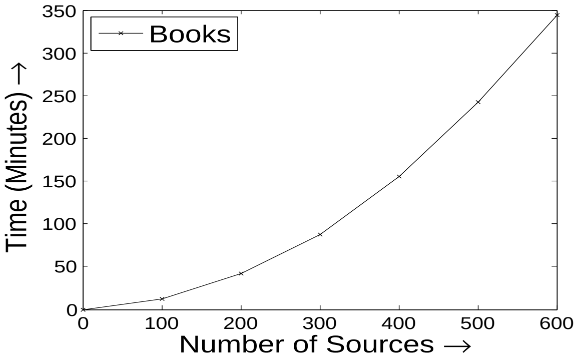


Figure 6: Time to compute agreement against number of sources.

selected. DCG is not computed for these experiments since all the results are ranked by Google Base ranking, hence ranking order comparison is not required.

In Figure 3(c) and 3(d), the *GBase* is the stand-alone Google Base ranking. *GBase-Domain* is the Google Base ranking searching only in the domain sources selected using our query probing. For example, in Figure 3(d), Google Base is made to search only on the 675 book domain sources used in our experiments. The plots 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. SourceRank outperforms all other methods (confidence levels are 95% or more). For the movie domain, SourceRank precision exceeds Google Base by 38% and coverage by 23%. For books the differences are 53% and 25% with Google Base and Coverage respectively. The small difference between the Google Base and Google Base-domain has low statistical significance (below 80%) hence not conclusive.

6.4 Trustworthiness Evaluation

In the next set of experiments, we evaluate the ability of SourceRank to eliminate untrustworthy sources. For tuples, corruption in the attribute values not specified in the query manifests as untrustworthy results, whereas mismatch in attributes values specified in the query manifests as the irrelevant results. Since the title is the specified attribute for our queries, we corrupted the attributes other than the title values of the source crawls. Values are replaced by random strings for corruption. SourceRank, Coverage and CORI ranks are recomputed using these corrupted crawls, 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 runs. Since CORI ranking is query specific, the 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 domain are shown in Figure 4. The Coverage and CORI are oblivious of the corruption, and do not lower rank of the corrupted sources. This is not surprising, since any query based relevance measure would not be able to capture the corruption in the attributes not specified in the query. On the other hand, the SourceRank of the corrupted sources reduces almost linearly with the corruption level. This corruption-sensitivity of SourceRank would be helpful in solving the trust problems we discussed in the introduction (e.g. the solution manual with the same title and low non-existent prices etc).

6.5 Collusion Evaluation

We also performed a stand-alone ground truth evaluation of collusion and adjusted agreement. Since the ground truth—degree of collusion—of the online databases is unknown, these evaluations are performed using controlled ranking functions on a data set of a million records from IMDB [2]. The records are replicated to create two databases of one million each. For a query, the set of tuples are fetched based on the key word match and ranked. To implement ranking, a random score is assigned to each tuple and tuples are sorted on this score. If these scores for a given tuple in two databases are independent random numbers, the rankings are completely independent. If the score for a tuple is the same for both the databases, rankings are completely correlated. To achieve mid levels of correlations, weighted combinations of two independent random numbers are used.

Figure 5 shows the variation of collusion, agreement, and adjusted agreement with the correlation of the two databases. The correlation is progressively reduced from left to right. At the left, they are complete mirrors with the same ranking and data, and as we go right, the rank correlation decreases. As we observe in the graph, when the databases have the same rankings, the collusion and agreements are the same, making the adjusted agreement zero. This clearly cancels out agreement between mirrors and near mirrors. Even for a small reduction in the rank correlation, the collusion falls rapidly, whereas agreement reduces more gradually. Consequently the adjusted agreement increases rapidly. This rapid increase avoids canceling agreement between the genuine sources. In particular, the low sensitivity of the adjusted agreement in the correlation range 0.9 to 0 shows its immunity to the genuine correlations of databases. At low correlations, the adjusted agreement is almost the same as the original agreement as desired. These experiments satisfy the two desiderata of collusion detection we discussed in Section 5. The method penalizes mirrors and near mirrors, whereas genuine agreement between the sources is kept intact.

6.6 Timing Evaluation

We already know that random walk computation is feasible at web scale [11]. Hence for the timing experiments, we focus on the agreement graph computation time. 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 a 3.16 GHz, 3.25 GB RAM Intel Desktop PC with Windows XP Operating System.

Figure 6 shows the variation of agreement graph computation time of the 600 of the 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. These agreement values can be computed in isolation—without interprocess communication to pass intermediate results between the nodes.

7. CONCLUSION

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 depended on purely query based measures to assess the relevance of a source. The relevance assessment based solely on query similarity is easily 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. For added robustness of the ranking, we assess and compensate for the source collusion while computing the agreements. Our comprehensive empirical evaluation shows that SourceRank improves relevance sources selected compared to existing methods and effectively removes corrupted sources. We also demonstrated that combining SourceRank with Google Product search ranking significantly improves the quality of the results.

8. REFERENCES

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

[2] IMDB movie database. http://www.imdb.com.

[3] New york times best sellers. http://www.hawes.com/number1s.htm.

[4] New york times guide to best 1000 movies. http://www.nytimes.com/ref/movies/1000best.html.

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

[6] S. Agrawal, K. Chakrabarti, S. Chaudhuri, V. Ganti, A. Konig, and D. Xin. Exploiting web search engines to search structured databases. In *Proceedings of WWW*, pages 501–510. ACM, 2009.

[7] A. Arasu and H. Garcia-Molina. Extracting structured data from Web pages. In *Proceedings of SIGMOD*.

[8] R. Balakrishnan and S. Kambhampati. Factal: Integrating Deep Web Based on Trust and Relevance. In *Proceedings of WWW*. ACM, 2011.

[9] 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.

[10] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using BANKS. In *ICDE*, page 0431, 2002.

[11] 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.

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

[13] 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.

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

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

[16] W. Croft. Combining approaches to information retrieval. *Advances in information retrieval*, 7:1–36, 2000.

[17] X. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global detection of complex copying relationships between sources. *Proceedings of the VLDB Endowment*, 3(1), 2010.

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

[19] D. Gleich, P. Constantine, A. Flaxman, and A. Gunawardana. Tracking the random surfer: empirically measured teleportation parameters in PageRank. In *Proceedings of WWW*, 2010.

[20] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of VLDB*, 2004.

[21] P. Ipeirotis and L. Gravano. When one sample is not enough: improving text database selection using shrinkage. *SIGMOD*, pages 767–778, 2004.

[22] J. KLEINBERG. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632, 1999.

[23] N. Koudas, S. Sarawagi, and D. Srivastava. Record linkage: similarity measures and algorithms. In *Proceedings of SIGMOD*, page 803. ACM, 2006.

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

[25] 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.

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

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

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

[29] J. Wang, J. Wen, F. Lochovsky, and W. Ma. Instance-based schema matching for web databases by domain-specific query probing. In *In Proceedings of the VLDB*, pages 408–419. VLDB Endowment, 2004.

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

[31] X. Yin, J. Han, and P. Yu. Truth discovery with multiple conflicting information providers on the web. *TKDE*, 2008.