A

Assessing Relevance and Trust of the Deep Web Sources and Results Based on Inter-Source Agreement

Raju Balakrishnan, Arizona State University
Subbarao Kambhampati, Arizona State University
Manishkumar Jha, Arizona State University

Deep web search engines face the formidable challenge of retrieving high quality results from the vast collection of searchable databases. Deep web search is a two step process of selecting the high quality sources and ranking the results from the selected sources. Though there are existing methods for both the steps, they assess the relevance of the sources and the results using the query-result similarity. When applied to the deep web these methods have two deficiencies. First is that they are agnostic to the correctness (trustworthiness) of the results. Secondly, the query based relevance does not consider the importance of the results and sources. These two considerations are essential for the deep web and open collections in general. Since a number of deep web sources provide answers to any query, we conjuncture that the agreements between these answers are helpful in assessing the importance and the trustworthiness of the sources and the results. For assessing source quality, we compute the agreement between the sources as the agreement of the answers returned. While computing the agreement, we also measure and compensate for the 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. For ranking results, we analyze the second order agreement between the results. Further extending SourceRank to multi-domain search, we propose a source ranking sensitive to the query domains. Multiple domain specific rankings of a source are computed, and these ranks are combined for the final ranking. We perform extensive evaluations on online and hundreds of Google Base sources spanning across domains. The proposed result and source rankings are implemented in the deep web search engine *Factal*. We demonstrate that the agreement analysis tracks source corruption. Further, our relevance evaluations show that our methods improve precision significantly over Google Base and the other baseline methods. The result ranking and the domain specific source ranking are evaluated separately.

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

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Deep web search, web trust, source rank, web database search, deep web integration, database integration, agreement analysis.

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 of web databases in size [Madhavan et al. 2006]. Searching the deep web has been identified as the next big challenge in information management [Wright 2008]. The

This research is supported by ONR grant N000140910032, NSF grant IIS-0905672 and two Google research awards.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

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 the deep web tuples. Realizing this approach consists of two broad steps—selecting the high quality sources, and ranking the best results returned by these sources at the top.

Although these steps received some attention in the context of text and relational databases (c.f. [Fuhr 1999; Nie and Kambhampati 2004; Bender et al. 2005; Shokouhi and Zobel 2007; Callan et al. 1995; Bhalotia et al. 2002; Ipeirotis and Gravano 2004]) existing approaches are focused on assessing the relevance 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 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 assessment 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, to 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 critical for the deep web like any other uncontrolled collection, 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 evaluate trustworthiness of sources based on local measures; since the 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 has to be evaluated in terms of the endorsement of the source by other sources. We deal with the problem of assessing trustworthiness and importance in the deep web by selecting sources based on their agreement; and extend the method for ranking the results returned by the 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 method from the surface web, like PageRank [Brin and Page 1998] or authorities and hubs [Kleinberg 1999]. 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, 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 importance and trust problems mentioned above. Importance is considered, since 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 [Google Products 2011]. A global relevance assessment based on the agreement of the results would thus have ranked the classic Godfather high. Similarly, regarding trust, the corruption is captured by agreement as the query answers from the legitimate sources are likely to disagree with the incor-

rect 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 Section 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 in agreement computation is that different web databases may represent the same entity syntactically differently [Cohen 1998]. To solve this, we combine record linkage models with entity matching techniques for accurate and speedy agreement computation. Further, attribute matchings are weighted by the computed attribute importance. The second challenge 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 keyword query. To address this, we adapt query based sampling methods used for text databases [Callan and Connell 2001].

Combating Source Collusion: Databases may enhance their SourceRank by colluding with each other. This is similar to the link-spam in the surface web. 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 query 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.

Extensions: While the source selection sensitive to the trust and importance is the main contribution of our work, we also undertake the related problems of ranking results and the topic-sensitive source analysis as two extensions described below:

Extension 1. Ranking Results: After selecting the quality sources based on the SourceRank, the results returned by the sources need to be combined and ranked. Within a source there may be variance among the quality of records, especially for user generated web 2.0 databases (e.g. youtube, craiglist etc.). Hence considering trustworthiness and importance is crucial for ranking results due to the same reasons elucidated for sources above. Since tuples are ranked during the query time, time to compute the ranking should be minimal. A simple agreement based method is to rank in the order of first order agreements—i.e. the sum of the agreements by other tuples. Going one level deeper, a second order agreement will consider the common friends two tuples have, in addition to the mutual agreement. As we compute higher and higher order agreements, the accuracies are likely to increase. However computation timings increase as well, since computation takes more iterations (please refer to Section 6 for computational details). We use second order agreement as a favorable balance between the time and accuracy.

Extension 2. Topic Sensitive Source Selection: A straightforward idea for extending SourceRank for multi-topic deep web search is a weighted combination with query similarity, like PageRank [Brin and Page 1998]. On the other hand, agreement by sources in the same topic (domain) is likely to be much more indicative of the importance of a source than endorsement by out of domain sources. Significantly, sources might have data relevant to multiple topics. The importance of the source might vary across those topics. For example, Barnes & Noble might be quite good as a book source but might not be a good movie source (even though it has information about both topics). These problems are noted for surface web (c.f. Haveliwala [2003]), but are more critical for the deep web since sources are even more likely to cross topics/domains

than single web pages. To account for this, we assess the domain-specific quality of the sources and evaluate the improvement.

To adapt the SourceRank for multiple-domains, we assess the source quality predominantly based on the endorsement from the same domain. For this, we use different sampling query sets for different domains. The quality score of the source for a domain solely depends on the answers to the queries in that domain. To rank the sources for a specific user query, a Naïve Bayes Classifier determines the domain of the query. The classifier gives the probability of the query belonging to different domains. These probabilities are used to weight the domain-specific SourceRanks to compute the combined topic sensitive SourceRank (TSR).

Implementation and Evaluation: Evaluations were performed on two sets of data sources–(i) online books and movie databases in TEL-8 repository [UIUC TEL-8 2003] and (ii) large number of books and movie sources in Google Base [Google Products 2011]. We performed three separate sets of evaluations for the basic SourceRank, topic specific SourceRank, and result ranking:

SourceRank. SourceRank improves the top-k precision and NDCG of source selection significantly over the existing methods including the Google Base. Trust experiments show that the SourceRank is highly effective in capturing corruption, as the score diminishes almost linearly with the source corruption. Runtime evaluations establish acceptable computation time. Further experiments show that the proposed collusion detection is effective in capturing mirrors and near-mirrors while still being sensitive to the the natural agreements between the sources.

Ranking Result Tuples. We evaluated the ability of the ranking to improve precision as a standalone method, and in combination with SourceRank. We show that the ranking significantly improves the precision, and NDCG over the baselines (Google Base and query similarity), and is very effective in removing corrupted results.

Topic Sensitive SourceRank (TSR). We compare the TSR with (i) Domain oblivious universal SourceRank and (ii) Google Base. In these evaluations on 1440 sources across four popular domains, precision values of TSR shows considerable improvement over that of the baselines.

The overall contributions of the paper are:

(1) An agreement based method to calculate the relevance of the deep web sources based on popularity.
(2) An agreement based method to calculate the trustworthiness of the deep web sources.
(3) Domain independent computation of the agreement between the deep web databases.
(4) A method for detecting collusion between the web databases.
(5) Empirical evaluations a on large number of sources.

Two extensions of the above methods to the problems of ranking the deep-web results and multi-domain source selection are:

(1) Ranking of results considering trust and importance (for ranking retrieved results from sources selected using SourceRank).
(2) Domain sensitive source ranking (for improved source selection based on the assumption that agreement by the sources in the same domain is more indicative of the source-quality in the domain).

The rest of this paper is organized as follows. The next section discusses the related work. Section 3 provides a formal justification for calculating the source reputation based on the agreement of the sources, and presents the SourceRank calculation method. The following section explains the computation of agreement between sources, and describes source sampling. Next, in Section 5 we explain source collusion detection. The following two sections (Section 6 and 7) describe two extension of SourceRank—agreement based ranking of results and the topic specific SourceRank. Section 8 describes the architecture of Factal search engine prototype. In Section 9, SourceRank is evaluated on multiple domains demonstrating the improved precision, trustworthiness, acceptable computation time and effectiveness of collusion detection. Subsequently Section 10 evaluates extensions TSR and the result ranking. Finally we present the conclusions and the possible future work in Section 11.

2. RELATED WORK

Different parts of this paper have been published before. SourceRank won the best poster award for WWW 2010 [Balakrishnan and Kambhampati 2010]. The conference version of the SourceRank paper [Balakrishnan and Kambhampati 2011b], and the source selection demonstration [Balakrishnan and Kambhampati 2011a] were presented at WWW 2011. The extensions to multi-domain deep web search and result ranking are added in this journal version. Thus the journal paper expands the scope from the trust and importance analysis for source selection to overall deep web search. The entire work is part of R Balakrishnan's PhD dissertation [**?**]

The indispensability and difficulty of source selection for the deep web have been recognized previously [Madhavan et al. 2006]. Current relational database selection methods minimize the cost by retrieving the maximum number of distinct records from a minimum number of sources [Nie and Kambhampati 2004]. Cost based web database selection is thus formulated as selecting the least number of databases maximizing number of relevant tuples (coverage). The related problem of collecting source statistics [Nie and Kambhampati 2004; Ipeirotis and Gravano 2004] has also been studied.

For text databases selection, Callan *et al.* [1995] formulated the CORI algorithm for query specific selection based on relevance. Cooperative and non-cooperative text database sampling [Callan and Connell 2001; Ipeirotis and Gravano 2004] and selection considering coverage and overlap to minimize the cost [Si and Callan 2003; Shokouhi and Zobel 2007] have also been addressed. As we mentioned in the introduction, none of these relational or text databases selection methods consider trust and importance of the databases.

Centralized warehousing approaches have been tried for integrating parts of the deep web. Google Product Search [Google Products 2011] works on Google Base (an open repository for products) which contains data from a large number of web databases. In a different surfacing approach of extending the search to web databases, Google crawls and indexes parts of the data in popular sources as html pages, disregarding the structure [Madhavan et al. 2008]. SourceRank can also be used in warehousing approaches to assess individual tuples based on their source lineage (indeed, we adopt this method for our evaluations on Google Base in Section 9.3).

The problem of ranking database tuples for keyword search in databases has been addressed [Bhalotia et al. 2002; Chaudhuri et al. 2004]. The focus of these papers is on relevance assessment of tuples for keyword search in a single database. The problems of trust and importance are not considered. Improving web database search relevance by exploiting the search results from a surface web search engine was attempted by Agrawal *et al.* [2009]. Their paper considers the relevance assessment for search in a

single database, and does not consider the trust problem. Further, the paper assumes the availability of high-quality web search results on the same topics as a reference.

Combining multiple retrieval methods for text documents has been used for improved accuracy [Croft 2000]. Lee [1997] 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.* [2004] proposed trust rank, an extension of page rank that considers trustworthiness of hyperlinked pages. Kurland and Lee [2005] proposed a re-ranking approach based on centrality on a language-model induced graph. Agreement on hidden variables between several learners has been used to achieve tractable learning time for joint learning [Liang et al. 2008].

Many of the related problems in deep web integration and search have been addressed. A number of methods are used for schema mapping of form interfaces of different web databases [Madhavan et al. 2005; Wang et al. 2004a; He and Chang 2003]. The sampling problem of web databases has been explored [Dasgupta et al. 2007; Wang and Lochovsky 2003]. A number of methods have been tried for record linkage [Koudas et al. 2006; Fellegi and Sunter 1969]. Completion and expansion of autonomous web database records at query time has been attempted [Gummadi et al. 2011; Wolf et al. 2009].

A probabilistic framework for trust assessment based on agreement of web pages for question answering has been presented by Yin *et al.* [2008], and Yin and Tan [2011]. Galland *et. al.* [Galland et al. 2010] did an experimental comparison of several fixed point methods to compute trustworthiness of binary facts (true or false). These frameworks however do not consider the influence of relevance on agreement, multiple correct answers to a query, record linkage and non-cooperative sources; thus limiting their usability for the deep web.

Dong *et al.* [2009; 2010] extend the basic idea of Yin *et al.* [2008] by computing source dependence and using a different accuracy model. In this work, source copying is detected based on completeness, accuracy and formatting [Dong et al. 2010]. Deep web collusion is however more than having the same data (hence data copying), since collusion manifests in data and ranking as discussed in Section 5. Further, extending Dong *et al.*'s method to deep web is hard as the access is limited to keyword search, and retrieving the entire data set is difficult. As we shall see, the collusion detection in the deep web needs to address different constraints including multiple true values, non-cooperative sources, and ranked answers. Our collusion detection approach accounts for these additional difficulties.

Clustered analysis of trust for multi-group environments has been attempted by Gupta *et al.* [2011]. Gupta and Han [2011] give a comprehensive survey of network based trust analysis which incidently also includes detailed discussions of SourceRank [Balakrishnan and Kambhampati 2011b].

3. SOURCERANK: TRUST AND RELEVANCE RANKING OF SOURCES

In this section we formalize our 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 two-step SourceRank calculation process: (i) creating a source graph based on the agreement between the sources and (ii) assessing the source reputation on the source graph.

3.1. Agreement as Endorsement

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 two sources return tuples r_1 and r_2 independently

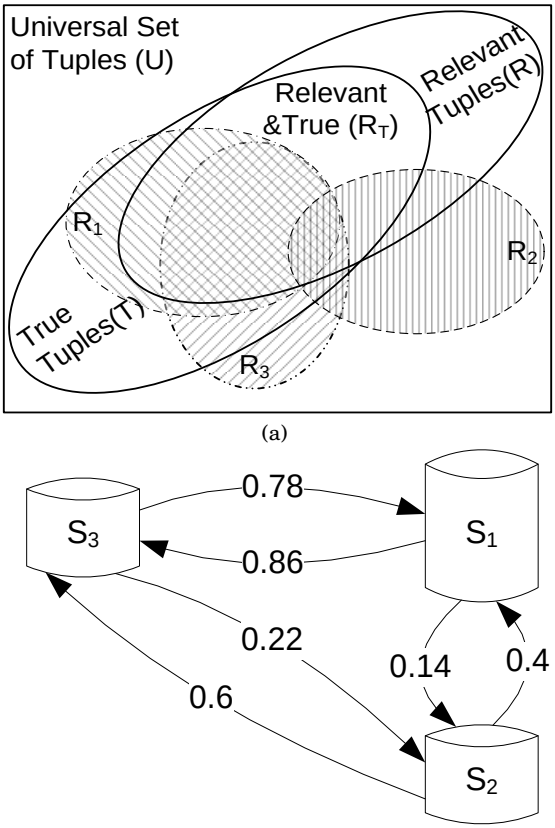


Fig. 1. (a) Model for explaining why agreement implies trust and relevance. Universal set U is the search space, R_T is the intersection of trustworthy tuple set T and relevant tuple set R (R_T is unknown). R_1, R_2 and R_3 are the result sets of three sources. (b) A sample agreement graph of the three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

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 returned by two sources, and $P_A(f_1, f_2)$ be the agreement probability of the 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 $P_A(f_1, f_2) = \frac{1}{10^4}$ ($\frac{1}{10^4-3}$ to be precise). 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. In Figure 1(a) R_1, R_2 and R_3 are the result sets returned by three independent sources. The result sets are the best effort estimates of R_T (assuming a good number of genuine sources). Typically the result sets from individual sources would contain a fraction of the relevant and trustworthy tuples from R_T, and a fraction of the 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. To improve the robustness of SourceRank against source dependence, we assess and compensate for the source collusion 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 Figure 1(b). The vertices represent the sources, and the 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 $a = A(R_1, R_2)$ ($0 \leq a \leq 1$) be the agreement between the results sets (calculated as described in Section 4). Correspondingly, the agreement graph has 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 is estimated based on the results returned in response to the sampling 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. Smoothing links account for the unseen samples. That is, even when there is no agreement between the sampled result sets used to calculate the links, there is a non-zero probability for some of the results to agree on queries not used for sampling. This probability corresponding to unseen samples is accounted by the smoothing links. 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 $R_{2}q$ 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 in our experiments. Empirical studies like Gleich *et al.* [Gleich et al.

2010] may help more accurate estimation. These smoothing links strongly connect the 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 the sum of the out edge weights from the vertex. This normalization allows us to interpret the edge weights as the transition probabilities of a random walk.

3.3. Calculating SourceRank

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

(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 by a source with a higher in-degree should be regarded more highly than endorsement by a source with lower 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 higher quality.

The agreement graph provides important guidance in selecting relevant and trustworthy sources. Any source having 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 [Brin and Page 1998]. 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 correspond to a measure of the global relevance of the 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 the source.

The SourceRank may be combined with query similarity based score of the source (please refer to Section 9.2 for details) for the final ranking as,

$$Score = \alpha \times querySim + (1 - \alpha) \times SourceRank \tag{6}$$

where $1 \geq \alpha \geq 0$ is a proportionality constant.

4. AGREEMENT COMPUTATION AND SAMPLING

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

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

(a)

	Title	Casting
1	The Godfather - The Coppola Restoration Giftset [Blu-ray]	Marlon Brando, Al Pacino
2	The Godfather - The Coppola Restoration Giftset DVD	Marlon Brando et al.

(b)

Fig. 2. Sample tuples returned by two movies databases to the query *Godfather* are shown in Table (a) (tuples from the first source) and (b) (tuples from the second source). Note that semantically same entities are represented in syntactically differently.

(a) Attribute value similarity: If the different web databases were using common domains for the names,[1] calculating agreement between the databases is trivial. But unfortunately, the assumption of common domains rarely holds in web databases [Cohen 1998]. For example, the title and casting attributes of tuples referring to the same movie returned from two databases are shown in Table 2(a) and 2(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 [Cohen 1998]. Since this challenge of matching attribute values is a name matching problem, we calculate the agreement between attribute values using SoftTF-IDF with Jaro-Winkler as the similarity measure [Cohen et al. 2003]. SoftTF-IDF measure is similar to the normal TF-IDF measure, but instead of considering only the 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 $log(IDF)$ used in the basic TF-IDF. SoftTFIDF is calculated as,

$$\mathcal{SIM}(v_i, v_j) = \sum_{w \in \mathcal{C}(\theta, v_i, v_j)} \mathcal{V}(w, v_i)\mathcal{V}(u, v_j)D(w, v_j) \tag{7}$$

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 [Cohen et al. 2003]. For pure numerical values (like price) we calculate the similarity as the ratio of the difference of values to the maximum of the two values.

(b) Tuple similarity: Tuples are modeled as a vector of bags [Cohen 1998]. The process of matching between two tuples is illustrated in Figure 3. 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

[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

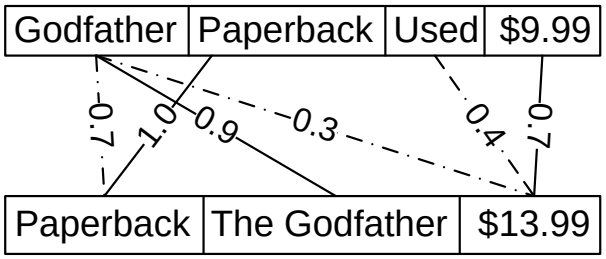


Fig. 3. Example tuple similarity calculation. The dotted line edges denote the similarities computed, and the solid edges represent the matches.

databases [Wang et al. 2004b]. 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 computing weights in Figure 3) 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 the edges, a single attribute value in t_1 may be found to 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.

$$S_{opt}(t,t') = \arg\max_{M} \sum_{(v_i \in t, v_2 \in t') \in M} \mathcal{SIM}(v_1, v_2) \tag{8}$$

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, 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 matching, 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 3, 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 3. The agreements between the tuples are calculated as the sum of the similarities of the individual matched values. The two tuples are considered matching if they exceed an empirically determined threshold of similarity.

The Felligi-Saunter record linkage model [Koudas et al. 2006] 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 denote same book than two entities with the common format *paperback*. To account for this, we weight the similarities between the matched attributes in the step above as

$$S(t,t') = \frac{\sum_{v_i,v_j \in M} w_{ij} \mathcal{SIM}(v_i, v_j)}{\sqrt{\sum_{v_i,v_j \in M} w_{ij}^2}} \tag{9}$$

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,

$$w_{ij} = log\left(\frac{\sum_k \mathcal{IDF}_{ik}}{|v_i|}\right) log\left(\frac{\sum_l \mathcal{IDF}_{jl}}{|v_j|}\right) \tag{10}$$

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 TF-IDF.

(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,

$$A(R_{1q}, R_{2q}) = \arg\max_{M} \sum_{(t \in R_{1q}, t' \in R_{2q}) \in M} S(t, t') \tag{11}$$

where M is the optimal matched pairs of tuples between R_{1q} and R_{2q} and $S(t, t')$ are as calculated as in Equation 9. Since this is again a bipartite matching problem similar to Equation 8, we use a greedy matching. The first tuple in R_{1q} is matched greedily against the most similar tuple 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 Equation 4.

We calculate agreement between the top-k (with $k = 5$) answer sets of each query in the sample set. 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. 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 of their contents, or allow access to the entire data set. Thus, the agreement graph must be computed over samples. In this section we describe our sampling strategy. We assume a minimal form based query interface allowing keyword queries; similar to the query based sampling used for the non-cooperative text databases [Callan and Connell 2001].

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 [NYT Top Books 2010]. For the sampling query set of movie domain, we use 200 randomly selected movies from the second edition of New York Times movie guide [NYT Movie Guide 2010].

As keyword queries for sampling, we use partial titles of the books/movies. We generate sampling queries by randomly deleting words from titles longer 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 by chance 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 the agreement between the sources as the probability of agreement by chance of top-k answers is less for larger answer sets.

(our initial experiments validated this assumption). The choice of deletion probability as 0.5 is based on cross-validation experiments.

We perform a query based sampling of a database by sending queries to the title keyword search field of the source. 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 [Arasu and Garcia-Molina 2003; Hammer et al. 1997; Zhai and Liu 2005], and can be automated (note that parsing is not required for Google Base experiments as structured tuples are returned). Averaging and aggregating over a number of sample queries is likely to increase the robustness of the overall agreement computation against the problems in linking individual records.

5. ASSESSING SOURCE COLLUSION

A potential problem for applying SourceRank is that the sources may clone themselves to boost their rankings. As the SourceRank becomes popular, collusion is likely to be more severe problem as well [Dong et al. 2010]. This is similar to the prevalence of link spam as link analysis became a common ranking method for the surface web. Considering this, we devise a method to measure and compensate for source collusion while computing SourceRank.

We measure the collusion of web databases on top-k answer sets, since agreement is also computed on top-k answers. While computing the agreement graph, we compensate for the source-collusion. Two issues complicating 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. This means that the mere fact that two databases have similar data samples need not necessarily indicate collusion (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*. This observation adds the complexity as even returning similar results on genuine queries does not indicate collusion. 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 behind our idea for collusion detection is that if two sources return the same top-k answers to the queries with a large number of possible answers (e.g. queries containing only stop words), they are possibly colluding. More formally, for two independently ranked sets of answers, the expected agreement between the top-k answers $E(A_k)$ (A_k is the agreement of top-k results) is

$$E(A_k) = \begin{cases} \frac{k}{n}(1-e) & \text{if } k < n \\ (1-e) & \text{otherwise} \end{cases} \tag{12}$$

where top-k answers are used to calculate agreement, the 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. $n \gg k$) 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 the highest document frequencies from the crawl described in 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 two sources on the samples based on genuine queries is multiplied by $(1 - collusion)$ to compute the adjusted agreement. Thus the weight of the edges in

Equation 5 is modified in this collusion-adjusted agreement graph as,

$$w(S_1 \rightarrow S_2) = \beta + (1-\beta) \times \frac{A_Q(S_1, S_2)(1 - collusion)}{|Q|} \tag{13}$$

These adjusted agreements are then used for computing SourceRank for the experiments. We also provide a standalone evaluation of the collusion measure in Section 9.6.

6. RANKING RESULTS FROM SOURCES

After sending queries to the selected sources, the returned results have to be combined and re-ranked. Given the open and adversarial nature of the deep web search, this re-ranking must be prepared to go beyond merging of different rankings. Otherwise sources may manipulate their rankings to improve the global rankings of their-own results, similar to the surface web search engine marketing. More generally, the search engine ranking should ideally be independent of any parameters easily manipulable by the sources to be robust. To support this, we adapt and extend the agreement analysis to result ranking.

We fetch the top$-k$ results (we used $k = 5$ for the system and the experiments) from the selected sources. A preliminary idea for ranking sensitive to importance is basic voting, i.e. counting the number of sources returning each tuple. But this simple voting is infeasible for the deep web due to the non-common domain problem illustrated in Figure 2. Hence we compute the agreement between the tuples as described in Section 4.1. We represent the agreement between the tuples as a graph with individual results as vertices. We do not consider the similarity between the tuples returned by the same source for the result-agreement graph. This is to prevent a source from boosting rank of a tuple by returning multiple copies of the tuple.

In the result-agreement graph, a simple ranking is by the first order agreements—i.e. the sum of the in-degrees of the tuples. We step one level deeper, and consider second order agreement. Second order agreement of two tuples considers the common friends of the tuples, in addition to the direct similarity between them. More precisely, second order agreement considers the number of other tuples similar to both of them.

Let the result-agreement graph be represented as a matrix A, where the entry a_{ij} represents the edge weight from the tuple j to the tuple i. We compute the second order agreement matrix as $S = A^T A$ (A is asymmetric). Finally we obtain the score r_i of a tuple t_i as the sum of the values the i^{th} row i.e $r_i = \sum_j s_{ij}$; and the tuples are ranked in the order of r_i. The trustworthiness and relevance of the ranking are evaluated in Section 10.1.

Since the result ranking is performed at the query time, reducing computation time is critical. We decided to use second order agreement against random walk due to the timing consideration. As we compute higher order agreements as in random walk, the accuracies as well as the computation timings tend to increase. We empirically compare the computation timings and precision of random walk and second order agreement in Section 10.1.

7. TSR: EXTENDING SOURCERANK FOR MULTIPLE TOPICS

As we mentioned in the introduction, deep web sources may contain data from multiple domains (topics). The quality of a source may vary significantly across these domains. The quality of a source specific to a topic is best indicated by the agreement by sources in that topic. Haveliwala [Haveliwala 2003] has shown that the topic-specific endorsement improves search for the surface web. This consideration is even more significant for the deep web, since sources contain records very specific to domains (e.g. book databases, movie databases etc.). Hence to customize SourceRank for the

multi-domain deep web, we introduce a topic sensitive SourceRank (TSR). In the next section we describe the sampling and computation procedures for TSR—SourceRank computed primarily based on the agreements by the sources in the same topic. Subsequently in Section 7.2 we describe the soft-classification of user queries into multiple domains.

7.1. TSR Sampling and Computation

For TSR computations we used 1440 sources spanning four domains—Books, Movies, Cameras and Music. Sampling method is the same as described for SourceRank in Section 4.2. Sampling queries are from New York Times best sellers [NYT Top Books 2010] (for books), Open Directory DVD Listing [DMOZ Movies 2011] (for movies), pbase.com [PBase Cameras 2011] (for cameras), and top-100 albums in 1986-2010 [Wiki Top Music 2011] (for music).

Each source has one TSR score corresponding to each domain. TSR for a domain is solely based on the source crawls using queries of that domain. For example, the agreement graph (described in Section 4.1) for movie TSRs is computed based on the answers to the movie queries by every source (we do not classify sources into domains). On this agreement graph, we compute the source score as the static visit probability of a weighted Markov random walk on the graph, as described in Section 3.3.

7.2. Topical Classification of Queries

Depending on the target domain user has in mind for the query, we need to use the TSR of the right domain to rank the sources. For example, we need to select a source based on the movie TSR for a movie query like "The Godfather Trilogy". the challenge of course is that the query topic is not declared a priori. In the following paragraphs we describe our classification approach that uses a Naïve Bayes Classifier (NBC).

Training Data: For topic-descriptions to train our classifier, we use query based sampling similar to the sampling described in Section 7.1. The same set of sampling methods and list of queries are used. But instead of generating partial queries by deleting words randomly, we use full titles as queries. Full title query crawl is less noisy and is found to improve classification accuracy.

Classification Steps: Realistically, query classification to domains will be probabilistic at best, since classifying queries to domains is hard. Hence we adopt a soft classification approach using a multinomial NBC with maximum likelihood estimates. For a query q, we compute the probability of membership of q in topic c_i as,

$$P(c_i|q) = \frac{P(q|c_i)P(c_i)}{P(q)} \propto P(c_i) \prod_j P(q_j|c_i) \tag{14}$$

where q_j is the j^{th} term of q.

$P(c_i)$ can be set based on past query logs, but here we assume uniform probabilities for topic-classes. Hence the above equation reduces to,

$$P(c_i|q) \propto \prod_j P(q_j|c_i) \tag{15}$$

$P(q_j|c_i)$ is estimated as the ratio of number of occurrences of q_j in the training data corresponding to c_j to the total number of words.

After computing the topic probabilities of the query, we compute the query specific score of sources by combining the topical scores. For a source s_k, final combined score

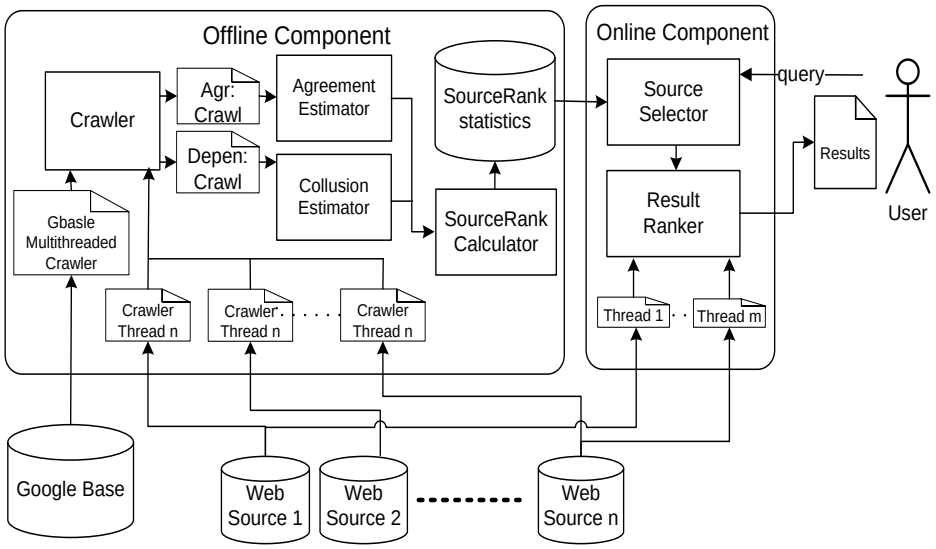


Fig. 4. Factal system architectural diagram. The online component contains processing steps at query time. Both the crawling and search are parallelized. (URL of the system is http://factal.eas.asu.edu).

TSR_{kq} specific to the query is given by,

$$TSR_{kq} = \sum_i P(c_i|q) TSR_{ki} \tag{16}$$

Sources are then ranked based on TSR_{kq} for query q.

8. FACTAL SYSTEM

The proposed source and result rankings are implemented in a vertical search engine named *Factal* (URL: http://factal.eas.asu.edu/). Sources are selected by the SourceRank and the results are ranked by the proposed result ranking.

The system shown in Figure 4 has an offline component and an online component. The offline component crawls the sources and computes the SourceRank. The online component selects the sources to search based on the SourceRank, retrieves and ranks the results at query time. The current prototype searches in the book and the movie domains. Search space contains 22 standalone online sources in each domain, along with 610 book sources and 209 movie sources from the Google Base. Sources are crawled using the sampling method described in Section 4.2 . For online sources one thread per database is used for crawling, and for Google Base we used forty threads (maximum acceptable for Google Base).

To process the queries, the top-k sources with highest SourceRank are selected. We set the value of k at five for the online sources and 10% of the total number of sources for the google base. Queries are dispatched to these sources in parallel spawning a separate thread for each source. Top-5 results are fetched from each source, and the results are combined and presented to the user.

The screenshot in Figure 5 shows the sample results for the query *Godfather* in Factal. The top results refer to the uncorrupted classic *The Godfather* movie, indicating that the proposed source and results rankings are able to handle the trust and importance problems. Please refer to Balakrishnan and Kambhampati [Balakrishnan and Kambhampati 2011a] for further details on Factal.

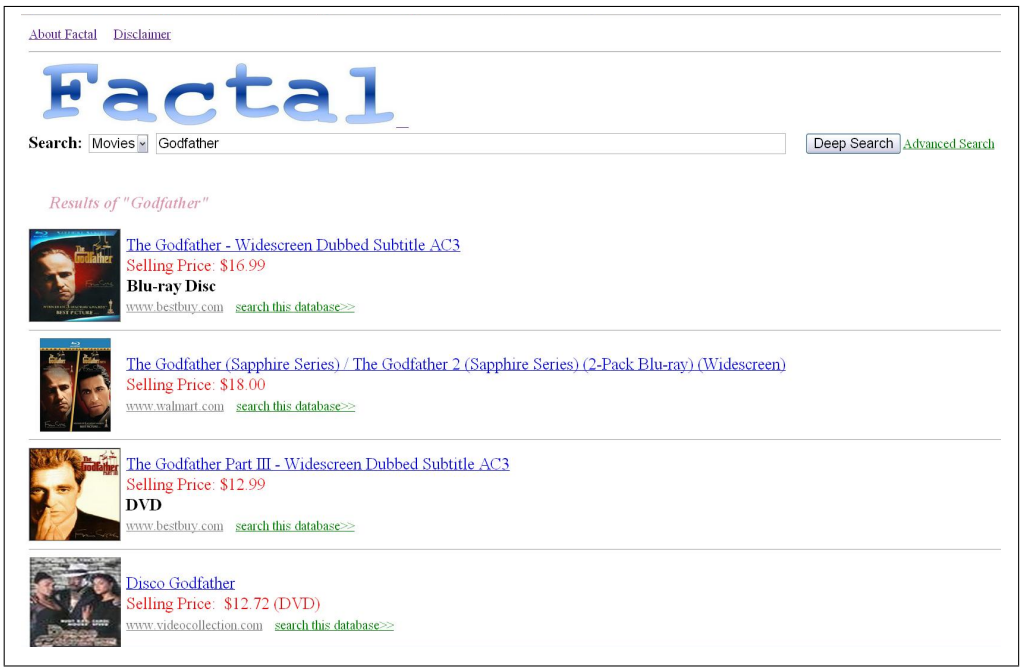


Fig. 5. Sample results of the query *Godfather* in Factal system.

9. SOURCERANK EVALUATION

We evaluate the effectiveness of the domain specific source selection using SourceRank computed based on the collusion adjusted-agreement. The top-k precision and discounted cumulative gain (DCG) of SourceRank-based source selection is compared with three baselines: (i) Coverage based ranking used in relational databases, (ii) CORI ranking used in text databases, and (iii) Google Product search on Google Base.

9.1. 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 databases—(i) a set of standalone online data sources (e.g. Amazon) (ii) hundreds of data sources collected via *Google Base* and (iii) a million IMDB records [IMDB 2011].

The databases listed in TEL-8 database list in the UIUC deep web interface repository [UIUC TEL-8 2003] are used for online evaluations (we used every working source in the repository). We used sixteen movie databases and seventeen book databases. In addition to these, we added five video sharing databases to the movie domain and five library 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 may not make a difference.

Google Base is a data collection from a large number of web databases, with API access to ranked results [Google Products 2011]. The Google Products Search works on Google Base. Each source in Google Base has a source id. For selecting in-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 sources belonging to that particular domain. Thus, we collected a set of 675 book

sources and 209 movie sources for our evaluations. Google Base API is used for sampling, 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 public books list. We generated queries by randomly removing words from the movie/book titles with probability of 0.5 (similar to the sampling queries). We used partial titles as the test queries, since typical web user queries are partial descriptions of objects. The number of queries used in different experiments varies between 50 to 80, so as to attain 95% confidence levels.

9.2. Baseline Methods

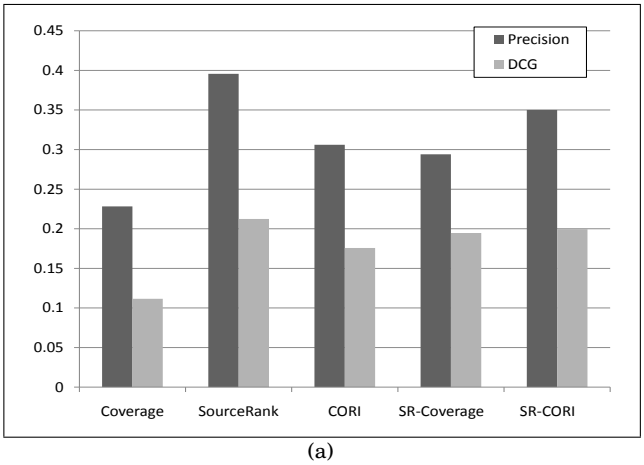
Coverage: Coverage is computed as the mean relevance of the top-5 results to the sampling queries described in Section 4.2. For assessing the relevance of the results, we used the SoftTF-IDF with Jaro-Winkler similarity between the query and the results (recall that the same similarity measure is used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

CORI: Callan *et al.* [Callan and Connell 2001] observed that using highest document frequency terms as crawling queries preforms well. Source statistics are collected using terms with the highest document frequency from the sample crawl (Section 4.2) as crawling queries. Similarly, we used two hundred high frequency queries and used the top-10 results for each query to create resource descriptions for CORI. We used the same parameter values as found to be optimal by Callan *et al.* [Callan et al. 1995]. CORI is used as the baseline, since the later developments like ReDDE [Si and Callan 2003] 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.

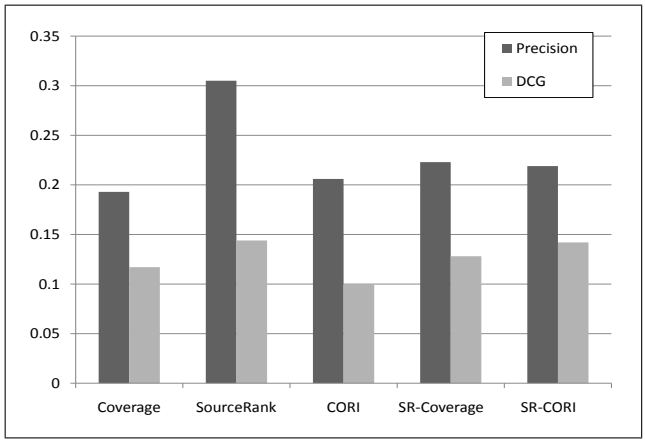
9.3. Relevance Evaluation

Assessing Relevance: Relevance is assessed using randomly chosen queries from the test queries described in Section 9.1. 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, suppose 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 9.1). 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 as irrelevant. Similarly for books, if the result is the queried book to sell, it is classified as relevant and otherwise classified as irrelevant. As an insurance against biased classification by the author, we randomly mixed tuples from all methods; so that the author did not know the method corresponding to the result while classifying. 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 statistical dispersion (measured by mean absolute deviation) of SourceRank scores compared to CORI scores.



(a)



(b)

Fig. 6. 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 (a) movies and (b) books .

The results of the top-4 source selection experiments in movie and books domain are shown in Figure 6(a) and 6(b). For both the domains, SourceRank clearly outperforms Coverage and CORI. For the movie domain, SourceRank increases precision over Coverage by 73.0% (i.e. $((0.395 - 0.228)/_{0.228}) \times 100$) and over CORI by 29.3%. DCG@5 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%. SourceRank outperforms standalone CORI and Coverage in both precision and DCG at a confidence level of 95%. Though the primary aim of the evaluation is not on differentiating SourceRank and combinations, we would like to mention that SourceRank outperformed the combinations at confidence levels exceeding 90% in most cases. Though this may be counter-intuitive at first, keep in mind that the selected sources return the results based on the query based relevance. Hence the results from SourceRank-only source selection implicitly account for the query similarity. Combining again with the query-relevance based method like CORI may be over-weighting query similarity.

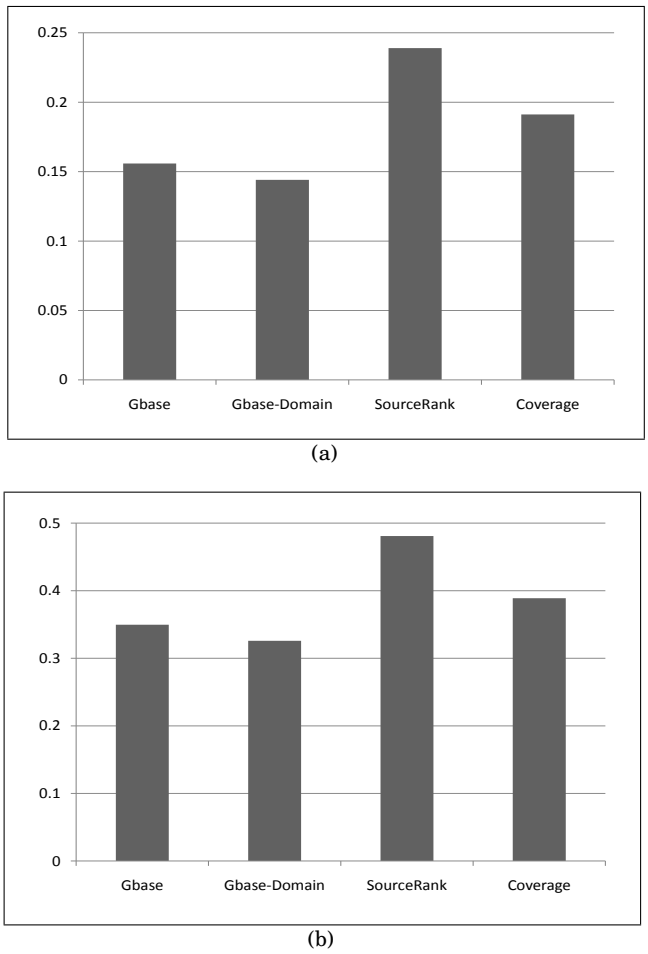
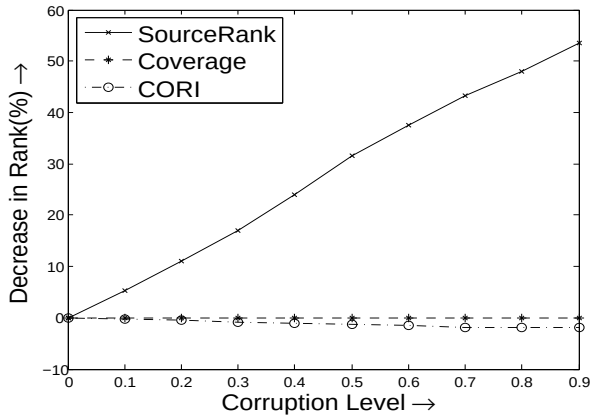


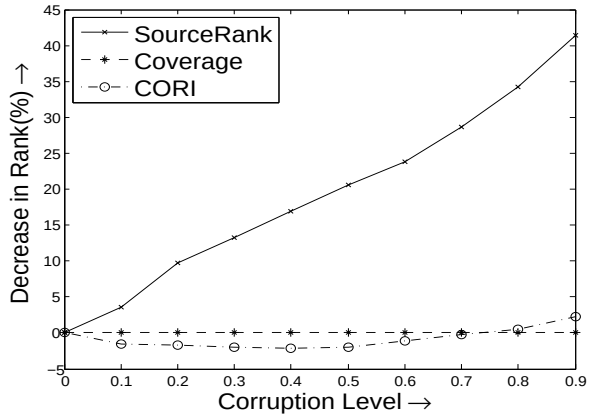
Fig. 7. Comparison of top-5 precision of results returned by SourceRank, Google Base and Coverage for (a) movies and (b) books.

As a note on the seemingly low precision values, these measure 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 the full first page of results the average top-5 precision is bound to be low. For example, if a search engine always returns one relevant result in top-5, the top-5 precision will be only 0.2.

Google Base: 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 the top of the source selection by SourceRank and compared with the standalone 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 9.1. Out of these 675 sources, we selected the top-67 (10%) sources based on SourceRank. Google Base is made to query only on this top-67 Sources, and the



(a)



(b)

Fig. 8. Decrease in the ranks of the sources with increasing source corruption levels in (a) movies and (b) books domains. SourceRank reduces almost linearly with corruption, while CORI and Coverage are insensitive to the corruption.

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

In Figure 7(a) and 7(b), the *GBase* denotes the standalone 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 7(b), Google Base is made to search only on the 675 book domain sources used in our experiments. For the plots labeled SourceRank and Coverage, first top-10% sources are selected using SourceRank and Coverage; and then the results retrieved from the selected sources are ranked by Google Base. 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 Google Base and Google Base-domain has low statistical significance (below 80%) and hence is not conclusive.

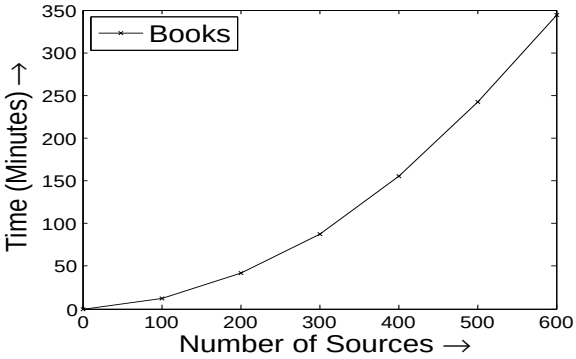


Fig. 9. Time to compute agreement against number of sources.

9.4. Trustworthiness Evaluation

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 attribute 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 8. The coverage and CORI are oblivious of the corruption, and do not lower rank of the corrupted sources. Significantly, this susceptibility to corruption is a deficiency of any query similarity based relevance assessment, since they are completely insensitive to 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.).

9.5. Timing Evaluation

We know that random walk computation is feasible at web scale [Brin and Page 1998]. 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 9 shows the variation of agreement graph computation time over 600 of the book sources from Google Base. As expected from time complexity formulae above, the time increases in quadratic time. Considering that the agreement computation is offline, the deep web scale computation should be feasible. In practice, sources in widely separated domains are not likely to show any significant agreement. Hence we may avoid computing agreement between all pairs of sources based on the domains, thereby significantly reducing computation time. Further, the agreement graph computation is

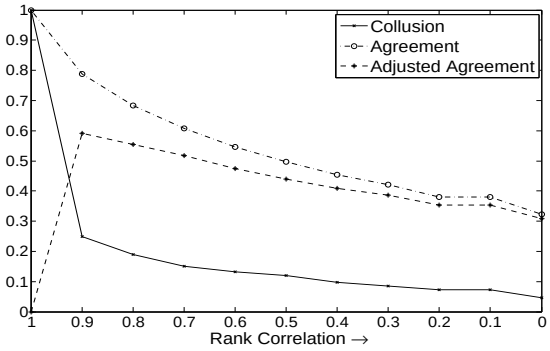


Fig. 10. Variation of Collusion, Agreement and Adjusted Agreement with rank correlations. Adjusted Agreement is $Agreement \times (1 - collusion)$.

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 inter-process communication to pass intermediate results between the nodes. Consequently, we will achieve a near-linear reduction in computation time with the number of computation nodes.

9.6. Collusion Evaluation

We performed a standalone ground truth evaluation of collusion detection. 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 [IMDB 2011]. We built two databases with varying degrees of collusion between them. For this, all the records are replicated to create two databases of one million records each. For a query, the set of tuples are fetched based on the keyword match and ranked. To implement ranking, a random score is assigned to each tuple and tuples are sorted on this score (every tuple is present in both the databases). If these scores for a given tuple in two databases are independent random numbers, the rankings are completely independent (hence databases have zero collusion). If the score for a tuple is the same for both the databases, rankings are completely correlated (full collusion or mirrors). To achieve mid levels of correlations between the sources, weighted combinations of two independent random numbers are used for ranking results.

Figure 10 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 cancels the adjusted agreement between mirrors (databases with the same data and ranking) 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. Consequently, mirrors and near mirrors are penalized, whereas genuine agreements between the sources are kept intact.

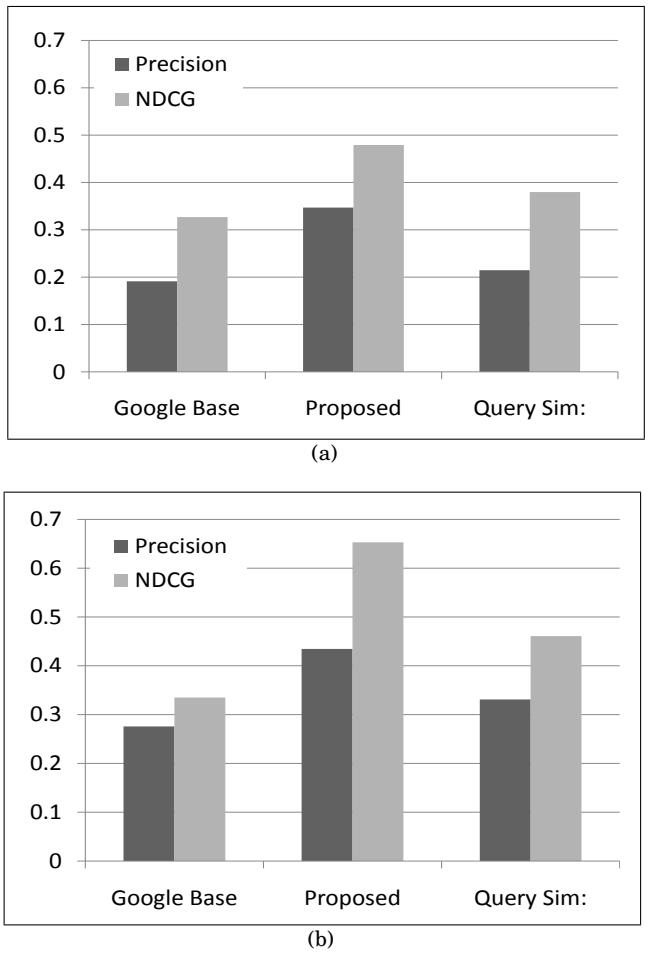


Fig. 11. Comparison of top-5 precisions and NDCG of TupleRank, Query Similarity, and Google Base (a) Without source selection. (b) With SourceRank based source selection.

10. EVALUATING EXTENSIONS

We describe the experimental evaluations of SourceRank extensions in this section. This section explains the evaluation of result ranking followed by evaluations of topic sensitive SourceRank (TSR) in Section 10.2.

10.1. Result Ranking Evaluation

We used 209 movie sources in Google Base described in Section 9 for these experiments. Top-5 precision, NDCG@5 and trustworthiness of results by the proposed ranking are compared with those of (i) relevance measured as the query similarity with tuples (using SoftTFIDF with Jaro-Winkler described in Section 4.1). (ii) the default relevance ranking of Google Base. Further, we compare the precision and computation timings of ranking based on random walk and second order agreement.

Relevance Results: We compared the relevance improvements of the standalone result ranking as well as in combination with SourceRank. Sufficient number of queries

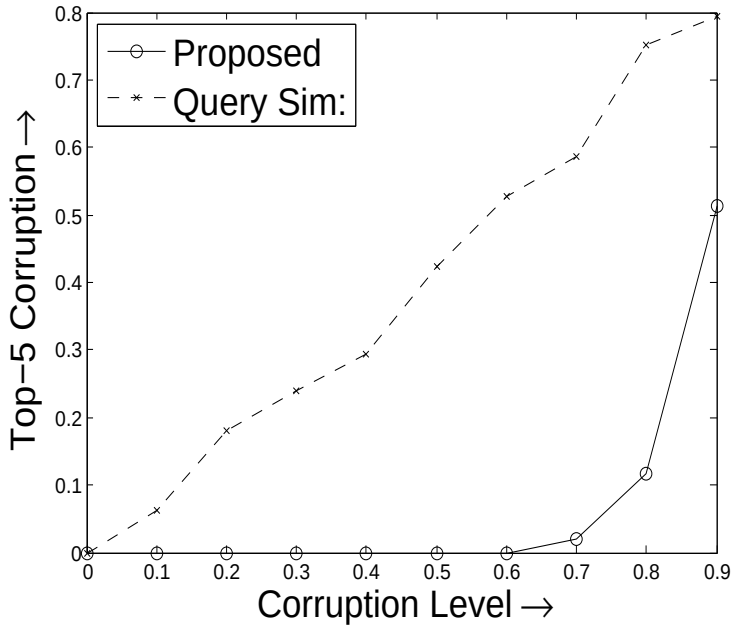


Fig. 12. Corruption of top-5 results of the proposed result ranking and query similarity against the increasing levels of result corruption.

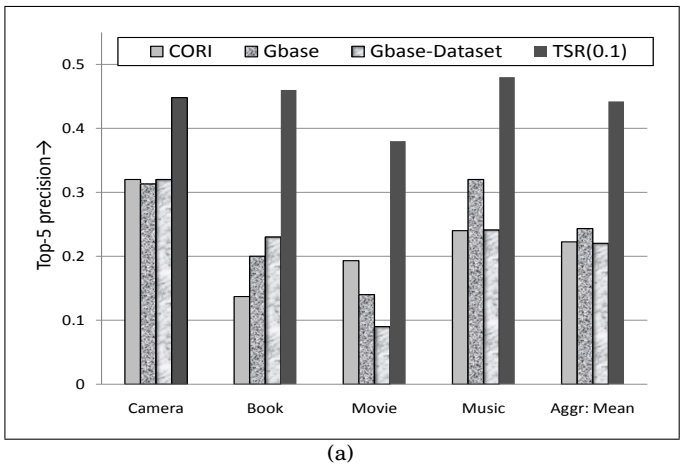
are used to differentiate both NDCG and precision of the proposed ranking with non-overlapping confidence intervals at a significance level of 0.95.

In Figure 11(a), top-5 results from sources are selected for each query. These results are combined and re-ranked using the three ranking methods. The comparison of top-5 precision and NDCG are shown in Figure 11(a). Precision is improved by 81% over Google Base and 61% over query similarity; and NDCG by 46% and 26% respectively over Google Base and query similarity. Note that the apparent difference in accuracy between the query similarity and Google Base is not conclusive as the difference is of low statistical significance.

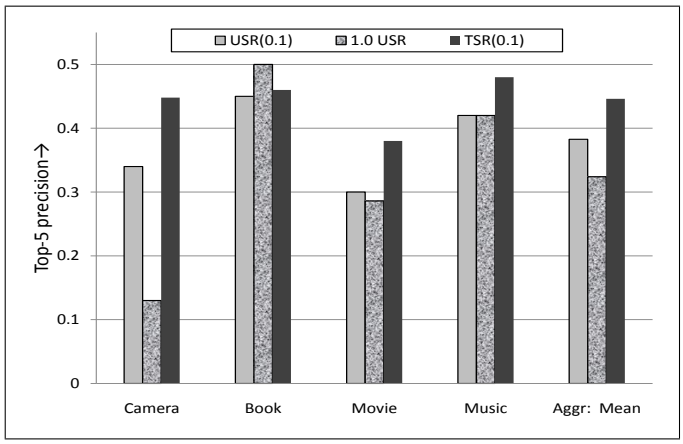
We used top-5 results since most web databases try to provide best precision for the top slots, as very few users go below top results [Richardson et al. 2007]. The ranking is applicable for other values of k as well. One consideration in fixing k is that a larger k will increase the number of tuples to be ranked, thus increasing the ranking time. Another factor is the number of sources searched. In general, as the number of sources increases, fetching fewer top results from each source is sufficient to compose a combined ranked list. Hence depending on the number of sources, ranking time constraints and other application requirements, the value of k may be varied for different searches.

The second set of experiments evaluated precision improvements when result ranking is combined with SourceRank. We selected the top 10% sources using SourceRank, and top-5 results from these selected sources are combined and ranked by the proposed ranking method. For the results shown in Figure 11(b), relevance is improved over Google Base and Query Similarity by 30 to 90%. Not surprisingly, the precision and NDCG of all the methods increase over those without source selection (Figure 11(a)).

Trust Results: Similar to the trust evaluation for the SourceRank described in Section 9.4, we corrupted a randomly selected subset of tuples by replacing attributes not specified in the query. After data corruption, the tuples are ranked using Query Sim-



(a)



(b)

Fig. 13. Comparison of top-5 precision of TSR(0.1) ($TSR \times 0.1 + CORI \times 0.9$) with (a) The query similarity based CORI and Google Base for different domains, and aggregate mean precision across the domains (b) Agreement based USR and USR(0.1) ($0.1 \times USR + 0.9 \times CORI$)

ilarity and the proposed ranking. Robustness to corruption of ranking is measured as the number of corrupted tuples in the top-5 results. The experiment is repeated for 50 queries in each corruption level and the results are shown in Figure 12. The query similarity is oblivious to the corruption—as the fraction of corrupted tuples in the top-5 is almost the same as the corruption level. In contrast, the proposed result ranking is highly robust to corruption, as all corrupted tuples are removed until 70% of the results are corrupted. At higher levels, the corruption of the top-5 tuples are bound to increase since there would be less than five uncorrupt tuples for many queries (e.g. at the corruption level one, any ranking method will have all the top-5 tuples corrupted).

Random Walk Comparison: We compared the precision and convergence of the second order agreement and the random walk. The difference in precision between the two was statistically insignificant. For fifty movie queries with no source selection, the top-k precision for the random walk and the second order agreement were 0.161 and

0.153 respectively; with a p-value of 0.42 in a paired ttest.[2] The mean number of iterations to converge for the random walk was 16.4. The second order agreement takes two iterations. Total time to rank for both the methods were similar (around 152 ms), since most of the time was spent on computing the similarity matrix. However, the similarity matrix size and hence the iteration time increases quadratically with the number of results—hence with the number of sources. Further as the number of sources increases, the iterations may become costlier as the computations may no longer be in memory. Consequently, the additional iterations in random walk may take more time for a larger number of sources and results, hence increasing the timing difference between the two methods. Considering all these factors together, we choose second order agreement for the result ranking.

10.2. Topic Sensitive SourceRank Evaluation

Data Set and Test Queries: We evaluated precision of TSR on 1440 sources in Google Base spanning across four topic classes—camera, book, movie, and music. Sources belonging to a domain are collected by query probing, as described for data collection for SourceRank experiments in Section 9.1. We used a total of 276 camera, 556 book, 572 movie, and 281 music sources, with some sources belonging to multiple domains. We do not run the trust experiments for TSR, since the trustworthiness of agreement based source selection is already established in Section 9.4.

To give an informal overview of the nature of these databases we notice that there is considerable variance in the quality of the sources. Many databases do not respond to the majority of queries; and the coverage of around 40% of sources is zero. SourceRank for these databases varies from 1 to 0.07 and coverage varies between one and zero after normalization. The SourceRank of these empty sources is not zero because of the smoothing links. Similar to other open domains, there is a wide variety of data, including some databases returning text documents with thousands of words as answers to some queries, instead of structured tuples.

Test query set contained a mix of queries from all four topic-classes, non-overlapping with the sampling queries. The test queries were generated by removing words in titles as described in Section 9.1. The number of test queries is varied for the different domains to get 0.95 statistical significance.

Baseline Methods: TSR is compared with the following agreement based and query similarity based source selection methods.

> ***Undifferentiated SourceRank (USR)***. The USR does not differentiate between the domains, similar to the single-domain SourceRank. A single agreement graph is created for the entire set of sources; using the sampling queries for all the domains described in Section 7.1. On this graph, a single source quality score for each source is computed.
> ***CORI***. We compared with standalone CORI (described in Section 9.2) and evaluated the combination of CORI with agreement based source selection.
> ***Google Base***. We compared with two-versions of Google Base. Stand alone Google Base and Google Base Dataset—Google Base restricted to search only on our crawled sources similar to SourceRank evaluations above (i.e. GBase-Domain in Section 9.3).

Assessing Relevance: The relevance is evaluated similar to the experiments described in Section 9.3. We selected top-10 sources for every test query and restricted

[2]These experiments were on different data set than the results in Figure 10.1 and are not directly comparable.

Google Base to query only on these sources. The resulting top-5 tuples are classified as relevant or irrelevant.

Evaluation Results: We compare precision of TSR(0.1) (i.e. $TSR \times 0.1 + CORI \times 0.9$) with the query similarity based measures i.e. CORI, Google Base and Google Base Dataset.[3] The results for individual domains and the aggregate means across the domains are illustrated in Figure 13(a). For every domain as well as for the aggregate, the improvements in precision by TSR(0.1) are considerable as the precisions improve up to 85% over competitors.

In the next set of experiments, we compared TSR(0.1) with standalone USR and USR(0.9) (i.e. $USR \times 0.1 + CORI \times 0.9$). Note that USR(0.9)—linear combination of USR with a query specific relevance measure—is a highly intuitive way of extending domain oblivious USR for the multi-domain deep web search.[4] The results for individual domains and the mean aggregate are illustrated in Figure 13(a). For three out of four topic-classes (Camera, Movies, and Music), TSR(0.1) out-performs USR(0.1) and USR with confidence levels 0.95 or more. For books, we found no statistically significant difference between USR(0.1) and TSR(0.1). This may be attributed to the fact that the source set was dominated by large number of good quality book sources, biasing USR ranking towards the book domain. Further we analyzed comparable performance of domain independent USR and domain specific USR(0.1) for three domains: music, movies and books (though this comparison is not the focus of our evaluation). This analysis revealed that there are many multi-domain sources providing good quality results for books, movies and music domains (e.g. Amazon, eBay). These versatile sources occupy top positions in USR returning reasonable results for all these domains.

11. CONCLUSIONS 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 identifying relevant and trustworthy information from the huge collection of sources. Current relevance assessments depend predominantly on query similarity. These query similarity based measures can be easily tampered by the content owner, as the measures are insensitive to the popularity and trustworthiness of the results. These latter considerations are crucial for both selecting sources and ranking results. We propose an approach for assessing trustworthiness and importance of sources as well as results based on the agreement between the results. For selecting sources, 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. Applying the agreement analysis for the results, we compute their trustworthiness and importance based on the second order agreement between the results. Extending SourceRank to a domain sensitive assessment of source quality, we propose Topical-SourceRank: a trust and relevance measure predominantly based on the endorsement of sources in the same domain. We implement the proposed source and result ranking in the deep web search engine prototype Factal (`http://factal.eas.asu.edu`). Our comprehensive empirical evaluation shows that SourceRank improves the relevance of the sources selected compared to exist-

[3]Again the higher weight for CORI is to compensate for the higher dispersion of TSR compared to CORI scores.

[4]This combination is similar to the linear combination of domain oblivious static PageRank and query similarity for the surface web [Brin and Page 1998]

ing methods and effectively removes corrupted sources. We also demonstrated that combining SourceRank with Google Product search ranking significantly improves the quality of the results. Further, our evaluations show that the proposed result ranking effectively improves precision and eliminates corrupted results. After illustrating that agreement captures trust and importance by these experiments, we proceed to compare TSR with domain oblivious SourceRank and the existing methods. The experiments demonstrated the added precision by Topical-SourceRank for multi-domain search.

11.1. Discussion and Future Work

The problems in deep web search are at least as complex as those in surface web search. Though the proposed source and result ranking methods solve some of the important ones, there are many possible areas of future research.

For domains without many redundant sources, (e.g. student database of a university) the agreement based methods may not work. It can be argued that, the need for analyzing trustworthiness and importance is also less in these types of databases. While the keyword match based methods like CORI or Coverage may be sufficient for these types of databases, the performance and improvement of these methods may be further explored.

For topic specific source selection, we currently do not determine source topics explicitly. Different agreement graphs are based on the manually harvested topic-specific sampling queries. It would be interesting to extend this by topical modeling or classification of databases [Gravano et al. 2003; He et al. 2004; Barbosa et al. 2007]. Topical sampling queries may be extracted automatically from the databases belonging to a topic after the classification [Madhavan et al. 2008].

The top result being the most popular one is likely to satisfy most number of users. On the other hand, to satisfy maximum number of users by top-k results, it is best to diversify top-k results. Another direction is to exploit user models, if profiles are available.

Deep web integration systems have to generate wrappers, automatically or semi-automatically [Arasu and Garcia-Molina 2003]. SourceRank and the proposed ranking tuples will add to the extraction errors as well. The extraction errors will be reflected in the same way as wrong attribute values, or as incomplete tuples. The agreement of these sources and results by other correctly extracted sources will decrease. Consequently, the extracted tuples and sources will be ranked down, effectively shielding users from these errors. The validity of this intuitive robustness of the proposed method against extraction errors may be further explored.

REFERENCES

AGRAWAL, S., CHAKRABARTI, K., CHAUDHURI, S., GANTI, V., KONIG, A., AND XIN, D. 2009. Exploiting web search engines to search structured databases. In *Proceedings of World Wide Web*. ACM, 501–510.

ARASU, A. AND GARCIA-MOLINA, H. 2003. Extracting structured data from Web pages. In *Proceedings of SIGMOD*. ACM Press New York, NY, USA, 337–348.

BALAKRISHNAN, R. AND KAMBHAMPATI, S. 2010. SourceRank: relevance and trust assessment for deep web sources based on inter-source agreement. In *Proceedings of World Wide Web*. ACM, 1055–1056.

BALAKRISHNAN, R. AND KAMBHAMPATI, S. 2011a. Factal: Integrating Deep Web Based on Trust and Relevance. In *Proceedings of World Wide Web*. ACM.

BALAKRISHNAN, R. AND KAMBHAMPATI, S. 2011b. Sourcerank: Relevance and trust assessment for deep web sources based on inter-source agreement. In *Proceedings of World Wide Web*. ACM, 227–236.

BARBOSA, L., FREIRE, J., AND SILVA, A. 2007. Organizing hidden-web databases by clustering visible web documents. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*. IEEE, 326–335.

BENDER, M., MICHEL, S., TRIANTAFILLOU, P., WEIKUM, G., AND ZIMMER, C. 2005. Improving collection selection with overlap awareness in P2P search engines. *SIGIR*, 67–74.

BHALOTIA, G., HULGERI, A., NAKHE, C., CHAKRABARTI, S., AND SUDARSHAN, S. 2002. Keyword searching and browsing in databases using BANKS. In *ICDE*. 0431.

BRIN, S. AND PAGE, L. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems 30,* 1-7, 107–117.

CALLAN, J. AND CONNELL, M. 2001. Query-based sampling of text databases. *ACM TOIS 19,* 2, 97–130.

CALLAN, J., LU, Z., AND CROFT, W. 1995. Searching distributed collections with inference networks. In *Proceedings of ACM SIGIR*. ACM, NY, USA, 21–28.

CHAUDHURI, S., DAS, G., HRISTIDIS, V., AND WEIKUM, G. 2004. Probabilistic ranking of database query results. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 888–899.

COHEN, W. 1998. Integration of heterogeneous databases without common domains using queries based on textual similarity. *ACM SIGMOD Record 27,* 2, 201–212.

COHEN, W., RAVIKUMAR, P., AND FIENBERG, S. 2003. A comparison of string distance metrics for name-matching tasks. In *IIWeb Workshop*.

CROFT, W. 2000. Combining approaches to information retrieval. *Advances in information retrieval 7*, 1–36.

DASGUPTA, A., DAS, G., AND MANNILA, H. 2007. A random walk approach to sampling hidden databases. In *Proceedings of SIGMOD*. ACM Press New York, NY, USA, 629–640.

DMOZ Movies 2011. Open directory project movies. http://www.dmoz.org/Arts/Movies/Titles/.

DONG, X., BERTI-EQUILLE, L., HU, Y., AND SRIVASTAVA, D. 2010. Global detection of complex copying relationships between sources. *Proceedings of the Very Large Databases Endowment 3,* 1.

DONG, X., BERTI-EQUILLE, L., AND SRIVASTAVA, D. 2009. Integrating conflicting data: the role of source dependence. In *PVLDB*.

FELLEGI, I. AND SUNTER, A. 1969. A theory for record linkage. *Journal of the American Statistical Association 64,* 328, 1183–1210.

FUHR, N. 1999. A Decision-Theoretic Approach to Database Selection in Networked IR. *ACM Transactions on Information Systems 17,* 3, 229–249.

GALLAND, A., ABITEBOUL, S., MARIAN, A., AND SENELLART, P. 2010. Corroborating information from disagreeing views. In *Proceedings of Web search and data mining*. WSDM '10. 131–140.

GLEICH, D., CONSTANTINE, P., FLAXMAN, A., AND GUNAWARDANA, A. 2010. Tracking the random surfer: empirically measured teleportation parameters in PageRank. In *Proceedings of World Wide Web*.

Google Products 2011. Google Products. http://www.google.com/products.

GRAVANO, L., IPEIROTIS, P., AND SAHAMI, M. 2003. QProber: A system for automatic classification of hidden-Web databases. *ACM Transactions on Information Systems 21,* 1, 1–41.

GUMMADI, R., KHULBE, A., KALAVAGATTU, A., SALVI, S., AND KAMBHAMPATI, S. 2011. Smartint: using mined attribute dependencies to integrate fragmented web databases. In *Proceedings of World Wide Web*. ACM, 51–52.

GUPTA, M. AND HAN, J. 2011. Heterogeneous network-based trust analysis: A survey.

GUPTA, M., SUN, Y., AND HAN, J. 2011. Trust analysis with clustering. In *Proceedings of World Wide Web*. ACM, 53–54.

GYÖNGYI, Z., GARCIA-MOLINA, H., AND PEDERSEN, J. 2004. Combating web spam with trustrank. In *Proceedings of Very Large Databases*.

HAMMER, J., GARCIA-MOLINA, H., CHO, J., ARANHA, R., AND CRESPO, A. 1997. Extracting Semistructured Information from the Web. In *Proceedings of the Workshop on Management of Semistructured Data*. Tucson, Arizona: ACM, 18–25.

HAVELIWALA, T. 2003. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE transactions on knowledge and data engineering*, 784–796.

HE, B. AND CHANG, K. 2003. Statistical schema matching across web query interfaces. In *Proceedings of SIGMOD*. ACM, 217–228.

HE, B., TAO, T., AND CHANG, K. 2004. Organizing structured web sources by query schemas: a clustering approach. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM, 22–31.

IMDB 2011. IMDB movie database. http://www.imdb.com.

IPEIROTIS, P. AND GRAVANO, L. 2004. When one sample is not enough: improving text database selection using shrinkage. *SIGMOD*, 767–778.

KLEINBERG, J. 1999. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM 46,* 5, 604–632.

KOUDAS, N., SARAWAGI, S., AND SRIVASTAVA, D. 2006. Record linkage: similarity measures and algorithms. In *Proceedings of SIGMOD*. ACM, 803.

KURLAND, O. AND LEE, L. 2005. Pagerank without hyperlinks: structural re-ranking using links induced by language models. In *Proceedings of ACM SIGIR*. ACM, 306–313.

LEE, J. 1997. Analyses of multiple evidence combination. In *ACM SIGIR Forum*. Vol. 31. ACM, 276.

LIANG, P., KLEIN, D., AND JORDAN, M. 2008. Agreement-based learning. *Advances in Neural Information Processing Systems 20*, 913–920.

MADHAVAN, J., BERNSTEIN, P., DOAN, A., AND HALEVY, A. 2005. Corpus-based schema matching. In *Data Engineering, 2005. ICDE 2005. Proceedings.* 57–68.

MADHAVAN, J., HALEVY, A., COHEN, S., DONG, X., JEFFERY, S., KO, D., AND YU, C. 2006. Structured Data Meets the Web: A Few Observations. *Data Engineering 31,* 4.

MADHAVAN, J., KO, D., KOT, Ł., GANAPATHY, V., RASMUSSEN, A., AND HALEVY, A. 2008. Google's deep web crawl. *Proceedings of Very Large Databases Endowment 1,* 2, 1241–1252.

NIE, Z. AND KAMBHAMPATI, S. 2004. A Frequency-based Approach for Mining Coverage Statistics in Data Integration. *Proceedings of ICDE*, 387.

NYT Movie Guide 2010. New york times guide to best 1000 movies. http://www.nytimes.com/ref/movies/1000best.html.

NYT Top Books 2010. New york times books best sellers. http://www.hawes.com/number1s.htm.

PBase Cameras 2011. Pbase camera list. http://www.pbase.com/cameras.

RICHARDSON, M., DOMINOWSKA, E., AND RAGNO, R. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of World Wide Web*. ACM, 521–530.

SHOKOUHI, M. AND ZOBEL, J. 2007. Federated text retrieval from uncooperative overlapped collections. In *Proceedings of the ACM SIGIR*. ACM.

SI, L. AND CALLAN, J. 2003. Relevant document distribution estimation method for resource selection. In *Proceedings of ACM SIGIR*. ACM New York, NY, USA, 298–305.

UIUC TEL-8 2003. UIUC TEL-8 repository. http://metaquerier.cs.uiuc.edu/repository/datasets/tel-8/index.html.

WANG, J. AND LOCHOVSKY, F. 2003. Data extraction and label assignment for web databases. In *Proceedings of World Wide Web*. ACM, 187–196.

WANG, J., WEN, J., LOCHOVSKY, F., AND MA, W. 2004a. Instance-based schema matching for web databases by domain-specific query probing. In *Proceedings of Very Large Databases*. VLDB Endowment, 408–419.

WANG, J., WEN, J., LOCHOVSKY, F., AND MA, W. 2004b. Instance-based schema matching for web databases by domain-specific query probing. In *In Proceedings of Very Large Databases*. VLDB Endowment, 408–419.

Wiki Top Music 2011. Best selling albums worldwide. http://en.wikipedia.org/wiki/List_of_best-selling_albums_worldwide.

WOLF, G., KALAVAGATTU, A., KHATRI, H., BALAKRISHNAN, R., CHOKSHI, B., FAN, J., CHEN, Y., AND KAMBHAMPATI, S. 2009. Query processing over incomplete autonomous databases: query rewriting using learned data dependencies. *The Very Large Databases Journal 18,* 5, 1167–1190.

WRIGHT, A. 2008. Searching the deep web. *Commmunications of ACM*.

YIN, X., HAN, J., AND YU, P. 2008. Truth discovery with multiple conflicting information providers on the web. *TKDE*.

YIN, X. AND TAN, W. 2011. Semi-supervised truth discovery. In *Proceedings of World Wide Web*. ACM, 217–226.

ZHAI, Y. AND LIU, B. 2005. Web data extraction based on partial tree alignment. In *Proceedings of World Wide Web*. ACM, 76–85.