Trust and Profit Sensitive Ranking for the Deep Web and On-line Advertisements

by

Raju Balakrishnan

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2012 by the
Graduate Supervisory Committee:

Subbarao Kambhampati, Chair
Yi Chen
AnHai Doan
Huan Liu

ARIZONA STATE UNIVERSITY

August 2012

ABSTRACT

Ranking is of definitive importance to both usability and profitability of web information systems. While ranking of results is crucial for the accessibility of information to the user, the ranking of online ads increases the profitability of the search provider. The scope of my thesis includes both search and ad ranking.

I consider the emerging problem of ranking the deep web data considering trustworthiness and relevance. I address the end-to-end deep web ranking by focusing on: (i) ranking and selection of the deep web databases (ii) topic sensitive ranking of the sources (iii) ranking the result tuples from the selected databases. Especially, assessing the trustworthiness and relevances of results for ranking is hard since the currently used link analysis is inapplicable (since deep web records do not have links). I formulated a method—namely *SourceRank*—to assess the trustworthiness and relevance of the sources based on the inter-source agreement. Secondly, I extend the SourceRank to consider the topic of the agreeing sources in multi-topic environments. Further, I formulate a ranking sensitive to trustworthiness and relevance for the individual results returned by the selected sources.

For ad ranking, I formulate a generalized ranking function—namely Click Efficiency (CE)—based on a realistic user click model of ads and documents. The CE ranking considers hitherto ignored parameters of perceived relevance and user dissatisfaction. CE ranking guaranteeing optimal utilities for the click model. Interestingly, I show that the existing ad and document ranking functions are reduced forms of the CE ranking under restrictive assumptions. Subsequently, I extend the CE ranking to include a pricing mechanism, designing a complete auction mechanism. My analysis proves several desirable properties including revenue dominance over popular Vickery-Clarke-Groves

(VCG) auctions for the same bid vector and existence of a Nash equilibrium in pure strategies. The equilibrium is socially optimal, and revenue equivalent to the truthful VCG equilibrium. Further, I relax the independence assumption in CE ranking and analyze the diversity ranking problem. I show that optimal diversity ranking is NP-Hard in general, and that a constant time approximation algorithm is not likely.

To my parents for their love and support.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to Subbarao Kambhampati for his advice and support. His excellent advising method combining guidance and intellectual freedom made my PhD experience both productive and exciting. I am extremely thankful for his immense patience in evaluating my research and providing constructive suggestions which motivated me to strive for excellence. I believe that the skills, expertise and wisdom he imparted on me will significantly enable all my future career endeavors.

I would like to thank to my other committee members Yi Chen, AnHai Doan, and Huan Liu for their guidance and support for my research and dissertation. Their exceptional co-operation and guidance significantly enabled a smoother dissertation process and contributed towards the higher quality of my research.

I am thankful to my industrial mentors Ullas Nambiar and Rushi Bhatt for their guidance and collaborations. My big thanks to my collaborators Garrett Wolf, Aravind Krishna Kalavagattu, Manishkumar Jha, Srijith Ravikumar for highly fruitful collaborative research. I would also like to thank my colleagues in Yochan research group—Menkes van den Briel, William Cushing, J Benton, Kartik Talamadupula, Tuan Anh Nguyen, Sushovan De, and Yuheng Hu—for their helpful suggestions and comments on my publications and presentations.

I am grateful to the late Dr. Donald Miller for his support as a teaching assistant during the initial years of my PhD program. I would like to thank many other faculty in Arizona State University, without their courses and instruction this dissertation would not have been possible. Further, I am thankful to all the anonymous reviewers of my publications whose criticism and suggestions helped me to improve the quality of my research.

My special thanks to my father, mother and sister for encouragement and love throughout my life and education. Further, I would like to thank all my friends who made my graduate school a better experience.

My dissertation is partially supported by ONR grant N000140910032 and two Google research awards.

TABLE OF CONTENTS

Page

CHAPTER

CHAPTER Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

Chapter 1
Introduction

Providing the best results at the top of their ranked list is the most important success factor of search engines. Similarly, their profitability depends largely on placing the interesting ads in the top few slots. Improved ranking algorithms are crucial for both these abilities. Result ranking enables the exploitation of the vast ocean of information available in both the surface web (HTML pages) and the deep web (web databases). On the other hand, the paid placement of ads drives the business by generating profit for the multi-billion dollar search engine market. In brief, for today's web with rapidly expanding types of data and its applications, improved ranking algorithms are the single most important problem in enhancing the usability and the profitability. We consider these two related problems in this dissertation: ranking the deep web data and ad ranking.

1.1 Deep Web Ranking

Considering the results ranking for search engines, we address the ranking problems pertaining to the deep web integration systems. The deep web is the collection of millions of databases connected to the web (examples of web databases ranges from popular databases like Amazon and Craiglist to numerous small sales catalogues). The size of the deep web is estimated to be many times of currently searchable surface web [1, 2]. Integrating and searching the deep web is a challenging problem with highly promising implications [1]. Since the deep web contains (semi)structured data records, the semantics implied by the structure can be leveraged for better search. We formulate an end-to-end deep web ranking by addressing the sub-problems of:

Ranking sources: Ranking sources considering trustworthiness and importance of the sources.

Topic-sensitive source ranking: Topic-sensitive analysis of trustworthiness and importance of sources in multi-topic environments.

Ranking Results: Ranking results after retrieval from multiple sources.

The foremost challenge in searching open collections like the deep web is assessing trustworthiness as well as importance of the information. Since anyone may upload any information to open collections, the search is potentially adversarial in nature. Among the many sources, the most trustworthy and relevant set needs to be selected. The previous ranking approaches in the deep web are focused on assessing the relevance based on local measures of similarity between the query and the answers expected from the source (c.f. [3, 4, 5, 6, 7, 8, 9]). 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 especially critical for uncontrolled collections like the deep web, since sources try to artificially boost their rankings. A global relevance measure should consider popularity of a result, as the popular results tend to be relevant. Moreover, it is imprudent to evaluate trustworthiness of sources and results 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 by other sources.

The algorithms like Pagerank [10] and HITS [11] used by the surface web search engines solve this problem by assessing link structure of the web. But link analysis is not applicable to the deep web since there are no hyperlinks between the database tuples. At a high level, we deal with the problem of assessing trustworthiness and importance in the deep web by ranking based on the agreement between different sources. We describe the specific solutions to these three ranking problems—ranking sources, topic-sensitive source ranking, and ranking results—in the three sections below.

1.1.1 Source Ranking

We introduce an agreement based source selection method sensitive to trustworthiness and relevance. Two sources agree with each other if they return the same records in answer to the same query. Important results are likely to be returned by a large number of sources. Hence a global relevance assessment based on the agreement of the results will rank the important results high. Similarly regarding trust, the corruption of results can be captured by agreement, since other legitimate sources answering the same query are likely to disagree with the incorrect results.

The primary challenge in computing agreement is that different web databases represent the same entity in syntactically different ways, making the agreement computation hard [12]. To solve this problem, we combine record linkage models with entity matching methods for accurate and efficient agreement computation.

As in the PageRank, databases may enhance SourceRank by colluding with each other. Differentiating genuine agreement from the collusion increases the robustness of the SourceRank. We devise a method to detect the source dependence and compensate for dependence while computing SourceRank.

1.1.2 Topic-Sensitive Source Ranking

A straightforward idea for extending SourceRank for multi-topic deep web search is a weighted combination with query similarity, like PageRank [10]. On the other hand, agreement by sources in the same topic is likely to be much more indicative of importance of a source than endorsement by out of the topic sources. Significantly, a source might have data relevant to multiple topics. The importance of the source might vary across these topics. For example, Barnes & Noble might be quite good as a book source but might not be as good as a movie source (even though it has information about both the topics). These problems are noted for surface web (c.f. Haveliwala [13]), but are more critical for the deep web since sources are even more likely to cross topics than single web pages. To account for this fact, we extend the deep web source selection by assessing a topic-specific quality metric for the sources and assessing the resulting improvement in search quality.

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

1.1.3 Result Ranking

In a typical deep web integration system, the user enters his queries at the mediator. The mediator will select a subset of sources based on the query and the source ranking, and dispatch the query to the selected sources. Each of the source will return its-own ranked set of results to the query. These ranked sets of results need to be combined and ranked. The result ranking focuses on solving this problem.

Though the sources are selected based on trust and relevance, within a source there may be variance in the quality of records. The variance in quality of results is especially high user generated web 2.0 databases (e.g. youtube, craiglist etc.). Hence similar to the sources, 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. We use second order agreement as a favorable balance between the time and accuracy.

In addition to the experimental evaluations, we implemented a prototype deep-web search engine—namely *Factal*—based on the source and result ranking algorithms described in this dissertation.

1.2 Ad Ranking

Most search engines derive their revenues by displaying a ranked set of ads relevant the user-query. These ads are ranked to primarily to maximize the revenue for the search engines. But the ads have to be relevant to users in addition to be profitable, as users will click only on relevant ads. Hence the ad ranking needs to consider both the relevance and profitability of the ads. Compared to ranking results, this added dimension of profitability in addition to relevance gives rise to interesting problems in ad ranking. In the second part of the dissertation, we consider the problem of ad ranking to maximize the profits for the search engines.

In this dissertation, we develop a complete ad ranking and auction mechanism in three steps. In the first step we propose a unified optimal ranking function based on a generalized click model of the user. In the second step, we develop a complete auction mechanism and analyze the properties. In the third step, we consider the problem of ranking ads considering diversity. The details of these three steps are discussed in the three sections below.

1.2.1 Ranking and Generalizations

Ranking is essentially an optimization of expected utilities based on the click model of users. In general, users browse through ranked lists of results or ads from top to bottom either clicking or skipping the results, or abandoning browsing the list due to impatience or satiation. The goal of the ranking is to maximize the expected relevances (or profits) of clicked results based on the click model of the users. The sort by relevance ranking suggested by Probability Ranking Principle (PRP) has been commonly used for search results for decades [14, 15]. In contrast, sorting by the expected profits calculated as the product of bid amount and Click Through Rate (CTR) is popular for ranking ads [16].

Recent click models suggest that the user click behaviors for both search results and targeted ads are the same [17, 18]. Considering this commonality, the only difference between the two ranking problems is the utilities of entities ranked: for documents the utility is the relevance and for the ads it is the cost-per-click. This suggests possibility of a unified ranking function for results and ads. The current segregation of document and ad ranking as separate areas does not consider this commonality. A unified approach can help to widen the scope of the related research to these two areas, and enable applications of existing ranking functions in one area to isomorphic problems in the other area as we will show below.

In addition to the unified approach, the recent click models consider the following parameters:

1. **Browsing Abandonment:** The user may abandon browsing ranked list at any point. The likelihood of abandonment may depend on the entities the user has already seen [18].

2. **Perceived Relevance:** Perceived relevance is the user's relevance assessment viewing only the search snippet or ad impression. The decision

to click or not depends on the perceived relevance, not on the actual relevance of the results [19, 20].

Though these two considerations are part of the click models [17, 18] how to exploit these parameters to improve ranking is unknown. The current document ranking is based on the simplifying assumption that the perceived relevance is equal to the actual relevance of the document, and ignores the browsing abandonment. The ad placement partially considers perceived relevance, but ignores abandonment probabilities.

We propose a unified optimal ranking function—namely *Click Efficiency (CE)*—based on a generalized click model of the user. CE is defined as the ratio of the standalone utility generated by an entity to the sum of the abandonment probability and click probability of that entity (abandonment probability is the probability for the user to abandon browsing the list after viewing the entity). The sum of the abandonment and click probability may be viewed as the click probability consumed by the entity. We derive the name Click Efficiency based on this view—similar to the definition of the mechanical efficiency of a machine as the ratio of the output to the input energy. We show that sorting in the descending order of CE of entities guarantees optimum ranking utility. We do not make assumptions on the utilities of the entities, which may be assessed relevance for documents or cost per click (CPC) charged based on the auction for ads. On plugging in the appropriate utilities—relevance for documents and CPC for the ads—the ranking specializes to document and ad ranking.

As a consequence of the generality, the proposed ranking will reduce to specific ranking problems on assumptions on the user behavior. We enumerate a hierarchy of ranking functions corresponding to the limiting assumptions on the click model. Most interestingly, some of these special cases correspond to the currently used document and ad ranking functions—including PRP and sort by expected profit described above. Further, some of the reduced ranking functions suggest new rankings for special cases of the click model—like a click model in which the user never abandons the search, or the perceived relevance is approximated as the actual relevance. This hierarchy elucidates interconnection between different ranking functions and the assumptions behind the rankings. We believe that this will help in choosing the appropriate ranking function for a particular user click behavior.

1.2.2 Pricing and Mechanism Design

Ad auctions specify a ranking and a pricing—how much each advertiser is charged. The profit of the search engines depends on ranking as well as pricing. Hence to apply the CE ranking to ad placement, a pricing mechanism has to be associated. We incorporate a second price based pricing mechanism with the proposed ranking. Our analysis establishes many interesting properties of the proposed mechanism. Particularly, we state and prove the existence of a Nash Equilibrium in pure strategies. At this equilibrium the profits of the search engine and the total revenue of the advertisers is simultaneously optimized. Like ranking, this is a generalized auction mechanism, and reduces to the existing Generalized Second Price auction (GSP) and Overture mechanisms under the same assumptions as that of the ranking. Further, the stated Nash Equilibrium is a general case of the equilibriums of these existing mechanisms. Comparing the mechanism properties with that of VCG [21, 22, 23], we show that for the same bid vector search engine revenue for the CE mechanism will be greater or equal to that of VCG. Further, the revenue for the proposed equilibrium is equal to the revenue of the truthful dominant strategy equilibrium of VCG.

1.2.3 Diversity Ranking

Our analysis so far was based on the assumption of parameter independence between the ranked entities. We relax this assumption and analyze the implications based on a specific well known problem—diversity ranking [24, 25, 26]. Diversity ranking tries to maximize the collective utility of top-k ranked entities. For a ranked list, an entity will reduce the residual utility of a similar entity in the list below it. Though optimizing many of the specific ranking functions incorporating diversity is known to be NP-Hard [24], an understanding of why this is an inherently hard problem is lacking. By analyzing a significantly general case, we show that even the very basic formulation of diversity ranking is NP-Hard. Further we extend our proof showing that a constant ratio approximation algorithm is unlikely. As a benefit of the generality of ranking, these results are applicable both for ads and documents.

1.3 Contributions of the Dissertation

The direct impact of the dissertation is on the ranking for the deep web and search ads. In addition to these, we believe that the proposed methods will have wider impact on ranking of open data collections with no explicit links, and other profit-sensitive ranking problems. The specific contributions of the deep web and ad ranking research are described in the following two sections.

1.3.1 Contributions in Deep Web Ranking

The most important contribution in document ranking is a method to assess trustworthiness and relevance in open data collections with no explicit hyperlinks. The basic trust assessment has been augmented by measuring and compensating collusion between the sources. Finally, the method has been extended to multi-topic environments and result ranking, completing an end-to-end ranking for the deep web integration. All methods are evaluated in multiple large scale real-world data sets. In addition to the evaluations, we implemented our methods in a prototype search engine.

In summery, the specific contributions of the dissertation in document ranking are enumerated below.

1. An agreement based method to calculate relevance of the deep web sources based on popularity.

2. An agreement based method to calculate trustworthiness of the deep web sources.

3. Topic independent computation of the agreement between the deep web sources.

4. A method for detecting collusion between the web sources.

5. Formal evaluations on large sets of sources.

6. Ranking of results considering trust and importance.

7. Topic sensitive source ranking.

Though the immediate impact of the dissertation is in deep web integration and search, we believe that the methods used may be extended to source and result ranking of other open data collections.

1.3.2 Contributions in Ad Ranking

The dissertation formulates an optimal ranking and auction strategy considering recent user browsing models. The ranking has been analyzed to illustrate applicability in a wide range of ranking problems. Subsequently, we associate pricing with the ranking to formulate a complete ad-auction mechanism. The properties of the mechanism have been analyzed including the equilibrium properties. Further, the revenues of the search engine and the advertisers are compared with other popular auction mechanisms.

The specific contributions of the dissertation in ad ranking are enumerated below.

1. Unified optimal ranking based on a generalized click model.

2. Optimal ranking considering abandonment probabilities for documents and ads.

3. Optimal Ranking considering perceived relevance of documents and ads.

4. A unified hierarchy of ranking functions and enumerating optimal rankings for different click models.

5. Analysis of general diversity ranking problem and hardness proofs.

6. Design and analysis of a generalized ad auction mechanism incorporating pricing with CE ranking.

7. Proving the existence of a socially optimal Nash Equilibrium with optimal advertisers revenue as well as optimal search engine profit.

8. Proof of search engine revenue dominance over VCG for equivalent bid vectors, and equilibrium revenue equivalence to the truthful VCG equilibrium.

Though the immediate impact is on the ad ranking, we believe that the proposed ranking will have impact on related ranking problems with a profit considerations; like recommendations and daily deals. The illustrated commonality between the ad and document rankings may result in closer integration of these two areas.

1.4 Organization of the Dissertation

The organization of the remaining chapters of the dissertation and brief overview of contents is given below.

Chapter 2 gives the necessary background on both deep web integration and ad-ranking. Typical integration system architecture, existing ranking approaches, challenges in ranking, and fundamentals of ad auctions are described.

Chapter 3 describes the method to assess the relevance and trustworthiness of the sources. Further, the agreement computation between the sources, computing the final source ranks, and source-collusion detection are described. The precision, trustworthiness, and timing of the method are evaluated in multiple data sets

Chapter 4 describes the topic sensitive assessment of the source quality. We describe the topic-wise query classifications and separate evaluations in this chapter.

Chapter 5 describes the ranking of the results returned by the sources and evaluations. The quality of the results and timing are evaluated in separate data sets. Further we describe the architecture of our end-to-end deep web integration prototype—namely *Factal*.

Chapter 6 describes our entire contributions to ad-ranking. The chapter describes the optimal ranking function and ranking taxonomy showing applicability to related ranking problems. Further we associate a pricing with the ranking to formulate a complete auction mechanism. The properties of the mechanism are analyzed in detail and compared with other popular ad-auctions.

Chapter 7 lists the background research in the area and describes the connections and differences to my research.

Chapter 8 presents our conclusions and possible future extensions to the dissertation.

While short proofs are presented in the respective chapters, longer proofs are presented separately in the Appendix.

Chapter 2

Background

This chapter describes the background on information integration and ad auctions required to easily understand the rest of the dissertation. We start by discussing deep web search scenario and typical approaches. Further, we describe considerations and challenges in ranking sources and multi-topic ranking for the deep web. Subsequently, we elucidate additional challenges in ranking the deep web results.

Section 2.5 gives an overview of ad ranking and describes a the profit and relevance considerations. Subsequently, we describe ad auction mechanisms, explain importance of Nash-equilibrium, and describe the popular auction mechanisms.

2.1 Searching the Deep Web

Searching the deep web has been identified as the next big challenge in information management [1]. There have been multiple approaches for searching the deep web with varying levels of difficulties and effectiveness. Source and result ranking problems we address in this dissertation are common to all these approaches.

2.1.1 Approaches

The easiest of these approaches—generally called *surfacing approach*—is commonly used by current search engines [27]. For surfacing, the deep web records are crawled by using the sampling queries and indexed like any other HTML page. The structure of the records is not considered for the search. The advantage of surfacing is easiness, ability to leverage on existing surface-web indexing and search capabilities, and ability to handle search volumes independent of capacities of individual sources. On the other hand, this method has the disadvantages of losing the semantics implied by the structure of the records, need for centralized storage, and difficulties to maintain data coherence.

Another approach to deep web search is building a centralized record warehouse—like the Google Base [28]. In this approach, records from number of databases are collected in a central structured warehouse. The advantages are maintaining the structure and ability to handle search volumes irrespective of the capacities of the individual databases. Disadvantages of this approach are need for a large centralized repository, difficulties in maintaining data coherence, and difficulties in retrieving the data in structured form.

The third, and presumably the most scalable approach that has emerged is data integration. In this approach is explained in Figure 2.1, the user enters the query at a central mediator. In response to the user query, the mediator selects a subset of sources and issues the query. The answers returned by different sources are aggregated and re-ranked before returning to the user. The advantage of this approach is the ability to consider structure for improved search, dispensability of a central storage, and ability to leverage distributed search capability of individual sources. Considering the scale, built-in search capabilities of individual sources, and dynamic nature of data, integration may be the most promising approach. The disadvantages are added technical challenges in source selection—as query volume should be within the source capacity—and complexity in sampling of sources.

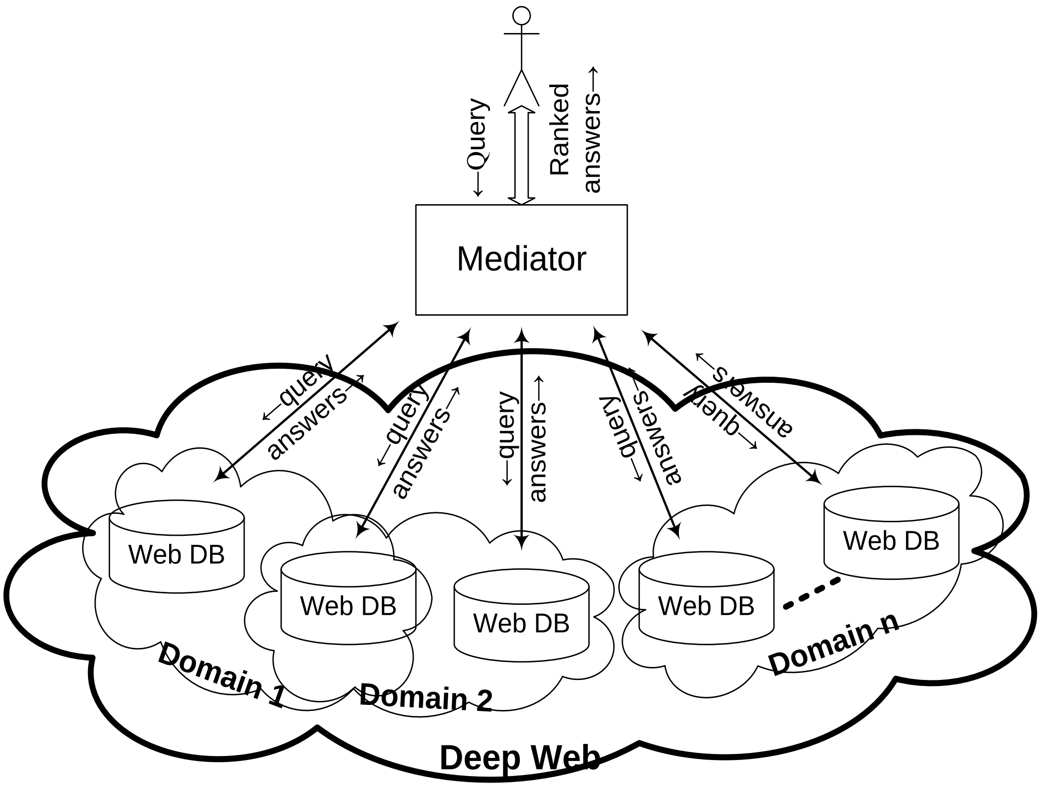


Figure 2.1: Deep web integration scenario. The sources belong to different domains (topics). In response to the user query, the mediator selects a subset of sources and issues the query. The answers returned by different sources are aggregated and re-ranked before returning to the user.

2.1.2 Ranking

For all the three approaches described above, source and result ranking problems are of paramount importance. The first phase of ranking—source ranking—orders sources based on the quality to select the best sources. In centralized approaches like surfacing and warehousing, the source quality assessment combined with the lineage (source of origin) gives an estimate of the data quality. After identifying high quality sources, the relevant records are fetched from these data source. In a centralized approach, this fetching may be based on an index. For the integration approach, the fetching is based on the distributed search by sending queries to the selected sources followed by parsing of the returned results. After fetching the results, the second phase is to rank the result returned by the selected sources to formulate the final ranked list. Another possibility—especially for the centralized approaches—is to combine these two phases to formulate a single score for the results combining source and result quality scores.

An additional challenge in the source quality assessment in the deep web is that the sources are segregated into multiple topics. Intuitively, the quality of a source may vary from domain to domain. For example, a source containing both books and movies may give high quality book results but low quality movie results. Hence it is the best to assess the source quality with respect to a domain. The query topic may be identified using a classifier, and source qualities for query topic may be used for ranking sources. This domain specific source selection incurs the query classification problem, in addition to the problem of assessing topic specific source quality.

We discuss these three problems of source ranking, result ranking, and domain-sensitive source ranking in the following three sections.

2.2 Ranking Sources

Most of the deep web sources are relational databases accessible by keyword queries. Sources generally implement a keyword search and a ranking. Users enters keyword queries in web forms and result pages containing relevant answers are returned in HTML.

To give a brief overview on deep web data, though the data is stored as structured tuples, results are generally wrapped in HTML for presentation. Unlike the static surface web pages, the pages corresponding to the deep web records are created dynamically at search time. The HTTP links to these dynamic pages are rare. Information extraction from the deep web is facilitated by the structure of the records. In spite of wrapping in the HTML, deep web results of the same source generally follow a repetitive template making it easier to extract the structure [29]. Many deep web sources allows only basic keyword search. Obtaining more information like statistics on contents and accessing complete data is hard at best. Another difficulty in analyzing the deep web data is that the same entities often represented (named) differently in different sources [12]. This non-uniform naming makes it hard to identify the same entities across different sources.

As we discussed in the introduction, basic idea of our ranking is to assess the quality of the deep web sources based on inter-sources agreement. Due to the nature of the deep web data described, realizing agreement based ranking faces multiple challenges:

1. **Computing Agreement:** The primary problem in computing agreement is that different web databases represent the same entity syntactically differently, making the agreement computation hard [12]. Solving this problem requires a combination of the record linkage in databases with inexact matching methods in information retrieval and named entity matching.

2. **Non-Cooperative Sources:** Most web databases are *non-cooperative*, as the access is limited to top-k answers to a keyword query. Hence the source quality assessment is restricted to limited keyword based sampling.

3. **Combating Source Collusion:** Like PageRank, the databases may enhance ranks by colluding with each other for increasing mutual endorsement. Differentiating genuine agreement between the sources from

the collusion increases the robustness of the ranking. We need effective methods to distinguish genuine agreement between the sources from source collusion.

We discuss the details of our methods to deal with these problems in Chapter 3.

2.3 Multi-Topic Ranking

Deep web sources are spread across multiple topics, as shown in Figure 2.1. A straight forward idea to extend the source ranking to multiple topics is to combine a domain-oblivious SourceRank to a query specific relevance assessment of sources. For example, for the query *godfather*, all the relevant sources containing the keyword may be identified based on a index on the sample data. After this, one or more of the sources among these relevant sources may be selected based on a static source ranking.

Our approach is to have one static source ranking for each possible search topic. The advantage is that the source quality is sensitive to the query-topic. In this approach, a source quality metric is computed for every topic in the search space. The sources which are members of more than one topic will have multiple quality scores corresponding to each domain. The query is classified into one or more topic classes at query time. Based on this query classification we combine the source scores from query-topic classes to form the final source ranking. For example, if the query *godfather* is likely to be a movie query with probability 0.6 and a book query with probability 0.4, we combine movie and book SourceRank scores of every sources with weights 0.6 and 0.4 respectively for the combined ranking.

This approach of topic specific search entails multiple challenges:

- **Query classification:** The membership of each query in topics need to be determined. This classification is time-critical since it is a query time process.

- **Multiple source rankings:** Instead of computing a single ranking for each source, multiple ranking for a source need to be computed corresponding to each domain of the source. This incurs different additional sampling challenges and computation time.

We describe the details of multi-topic source ranking in in Chapter 4.

2.4 Ranking Results

After ranking sources and retrieving the results, the final stage of searching the deep web is to combine and rank the results. A straightforward approach is to rank the results by keyword similarity. But the keyword similarity has the deficiency of disregarding the importance of trustworthiness of results as we mentioned in the introduction.

A seemingly related problem is merging of multiple ranked lists [6]. An important difference of our problem is that the sources do not return the ranking scores of the results. Further, weighing in the original order in which the sources returned the results may not be desirable, since the deep web ranking may be adversarial. Hence we target to evaluate the result quality from scratch based on a global approach.

The unique challenges in ranking results are:

1. **Computation Time:** Since the ranking of results has to be at the query time, the ranking time is directly added to the response time of the search engine. Since faster response is a critical success factor for the search engines, the ranking must be fast.

2. **Importance of Result Lineage:** Though the quality of the results depends on the source of origin, the degree of this dependence vary for different sources. For example a web 2.0 source (sources with user generated content like Youtube and Craiglist) the result quality may vary widely for different results. On the other hand, a closed database like Amazon would have more uniform quality. Deciding proper weight for the lineage in tuple quality evaluation may be challenging.

3. **Diversity Vs Uniformity:** Diversifying the results for ambiguous queries is likely to increases the overall relevance of the result sets. Determining the right amount of diversification of results is hard [25].

In this dissertation, the essential condition of acceptable computation timing is addressed. We leave the other two problems for the future research.

2.5 Ranking Ads

In this section we briefly describe ad ranking and pricing in a high level. Generally, search engines list sponsored search ads along with the organic search results. The difference between the sponsored results (ads) and organic results is that the advertisers pay for the user visits, whereas organic results are displayed free of cost. In general, advertisers bid for clicks in a per-click basis. Ads are selected and ranked based on the query of the user and bid value per click of the advertisers. Search ads need to be relevant to the users to maximize number of clicks. Simultaneously, the bids on the ads need to be high as well for maximal revenue from each click.

The overall ad ranking scenario is shown in Figure 2.2. The three parties involved are search engine, users and advertisers. The search engine acts as an intermediary between the users and the advertiser. Advertisers place bids for each keyword indicating their valuation of the clicks. Based on the bid values

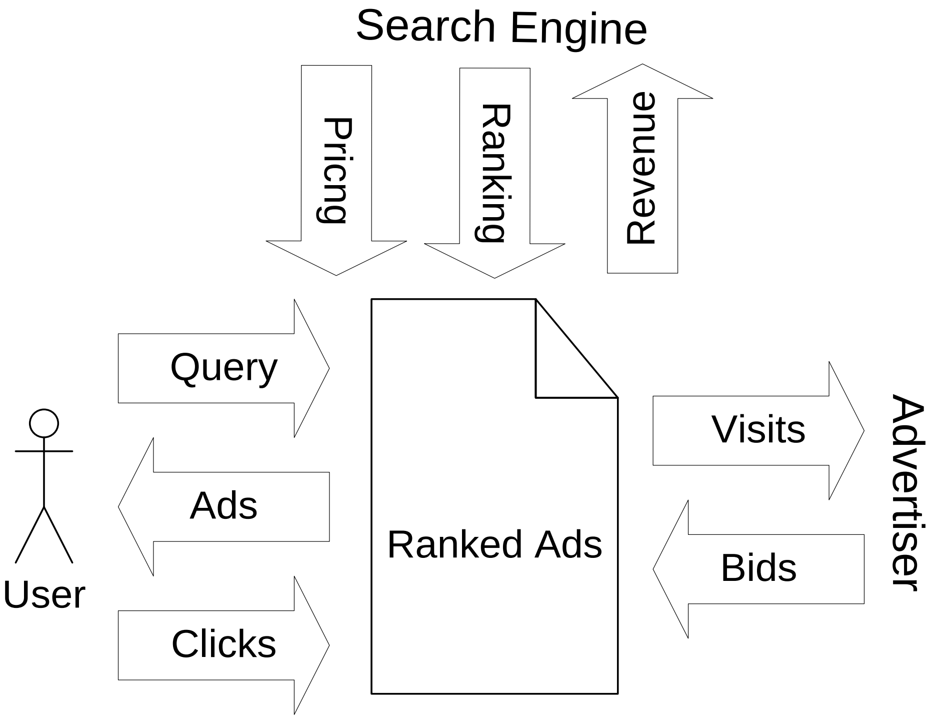


Figure 2.2: Ad ranking scenario. In response to the user query, search engine displays a ranked set of ads based on the bids and relevances of ads. User may click on these ads and visit the advertisers. Advertisers pay an amount (generally on a per-click basis) depending on their bids and the pricing strategy of the search engine.

and the pricing schema, search engine decides how much each advertiser has to pay for a click. When a user issues a query, ads relevant to the query are shown to the user in an order determined by the ranking schema. The user browses through this list of ads—clicking or skipping. The search engine gets revenues by charging the advertisers for the clicks. The advertisers get their revenue by the possible purchases of goods or services by the clicked users.

To give an overview of the dynamics of ad auctions, search engine decides the ranking of the ads based on the relevance and bid amounts. Generally, the primary objective of the ranking is to maximize the profit for the search engine. The number of clicks on an ad depends on the position and the relevance of the ad. More relevant the ad appears to the users, more likely the ad to be clicked. The users infer the relevance of the ad from the ad snippet (displayed title and short description) and decide to click or not to click. This user inferred relevance of the ad from the snippet is called perceived relevance. Perceived relevance may be different from the actual relevance—the relevance of the URL the ad is pointing to. Higher the perceived relevance of the ad to the user, higher the click probability of the ad.

The positional dependence of the click probability of an ad is captured by the click models [30]. Click models describe a general pattern in which the users browse the ads. Most models agree that the users start from the top ads, and progress downwards. This essentially means that it is profitable to place ads with higher bids and relevances higher up in the list to maximize profits. From the advertiser's point of view, it is better to be higher in the ranked list to receive more clicks. In general, ad rankings are formulated such that the positions of ads go up with the bid values to encourage the advertisers to bid high.

While ranking decides placing of ads pricing decides how much each advertiser pays for the clicks. Advertisers place keyword bids based on their click valuations. Based on the bid amount and pricing schema, search engines decide the pay-per-click (PPC)—amount the advertisers pay to the search engine for each click. PPC may be different from the bid of the advertiser. For example, the CPC is equal to the next higher bid for the keyword for second price auctions.

Ranking and pricing have to consider multiple factors for optimizing profit. A major challenge is considering both relevance and bids for ranking, and how to combine these two quantities to optimize profit. Another aspect is considering the mutual influence between the ads—i.e. effect of an ad in the list on the click probability of other ads. Further, the advertisers will keep experimenting by changing their bids to maximize the profit. In addition to the immediate profit, ranking and pricing may have to consider effects like change in profits for the advertisers, easiness to decide a bid amount etc. affecting long term search engine profits.

2.6 Ad Auctions

The ranking and pricing together compose an auction mechanism. In this section we describe the adversarial nature of ad auctions, Nash equilibriums, and popular auction mechanisms.

The advertisers may change their bids hundreds of times a day to increase profits. The advertiser's profit is number of clicks times difference of click value and PPC. The position of an ad—hence the number of clicks—increases with the increase in the bid amount. On the other hand, the PPC tends to increase with the bid amount as well. Every advertiser has to optimize his bid considering these two conflicting effects on profits. Further, position of the advertiser depends on bids of the other advertisers also. Since an advertiser do not know the bids of other advertisers, he has to resort to try and test—changing his bids and checking the resulting position. As the other advertisers change their bids, the optimal bid for the advertiser will change as well. As every advertiser tries to optimize bids, the advertisers are in a constant competition for higher positions resulting in ever changing bid values [31].

This dynamically changing bids are likely to reach a state of equilibrium eventually [32]. At an equilibrium stage, no advertiser will be able to improve his profit by changing his bid amounts. Hence no advertiser has an incentive to change his bid unilaterally. In other words, the bid value of every advertiser is a best response to bid values of other advertisers. This equilibrium stage corresponds to a study state in bid values. Such a study state corresponds to a Nash equilibrium in ad auctions. There may be multiple Nash equilibriums for an auction mechanism. The stable revenue from an auction mechanism is likely to be the revenue corresponding to one of the Nash equilibriums. Hence properties and revenues corresponding to the Nash equilibriums are of high interest in mechanism design.

2.6.1 Popular Auction Mechanisms

We discuss two of the most popular auction mechanisms for online ads, which we use as benchmarks later in the dissertation.

2.6.1.1 Vickery-Clarke-Groves Auction (VCG)

In VCG [21, 22, 23] the ads are ranked in the order of the expected revenues. Expected revenue from an ad is equal to the product of the bid amount and click through rate (CTR). CTR is the probability of a user clicking the ad having viewed it. The pricing for an ad is equal to the total lose in revenue to the other advertisers due to its presence in the auction. For example, suppose there are two advertisers. Let us assume that the CTR of both the ads are one for simplicity. The first advertiser bids 3 dollars and wins the top position, and receives, say 10 clicks. The second advertiser bids 2 dollars, and receives 6 clicks. If the first advertisers were not bidding, the second advertiser would have placed in the top position receiving 10 clicks instead of 6. So the total amount charged to the first advertiser is the increase in profit of the second bidder i.e. $4 \times 2 = \$8$. An excellent property of the VCG is that the truth telling is the dominant strategy: profit for every advertiser is the maximum if he bids his true valuation of clicks irrespective of other bids.

2.6.1.2 Generalized Second Price Auction

This is the strategy used by the Google search engine [31]. GSP ranks the ads by the expected profits like VCG. Unlike VCG, GSP uses a simple pricing, in which the pay-per-click is equal to the next highest bid. GSP pricing is different from VCG if there are more than two bidders. Truth telling is not a dominant strategy in GSP. GSP is revenue dominant over VCG, i.e. for any equilibrium in GSP the revenue of the search engine is at least as high as dominant strategy (truth telling) equilibrium of VCG [31].

Chapter 3

Ranking Deep-web Sources

Semantically rich structured data in the deep web is spread across millions of sources. Intuitively, the first step in finding the right information in the deep web is selecting the right sources. While selecting the sources, number of quality attributes like relevance, correctness of the information, topic of the source, response time of the sources etc. need to be considered. Many of these attributes can be estimated by directly borrowing ideas from the surface web search. However, existing methods for evaluating trustworthiness and relevance of data are not applicable to the deep web. Hence we focus specifically on these two problems in this chapter.

As we mentioned above, our method relies on agreement between the sources. We give a formal explanation of why agreement implies trustworthiness and importance. Graph representation of the agreement is described in Section 3.2. Subsequently, we describe random walk based computation of SourceRank—our ranking function. Next section describes three level computation of the agreement and query based sampling of the data sources. Next, we address the robustness of SourceRank. We describe detecting and compensating for the collusion. We evaluate the methods in multiple datasets. The relevance, trustworthiness, timing, and effectiveness SourceRank and of collusion detection are evaluated. The experiments demonstrate effectiveness of the proposed source ranking and collusion detection as well as acceptable computation timings.

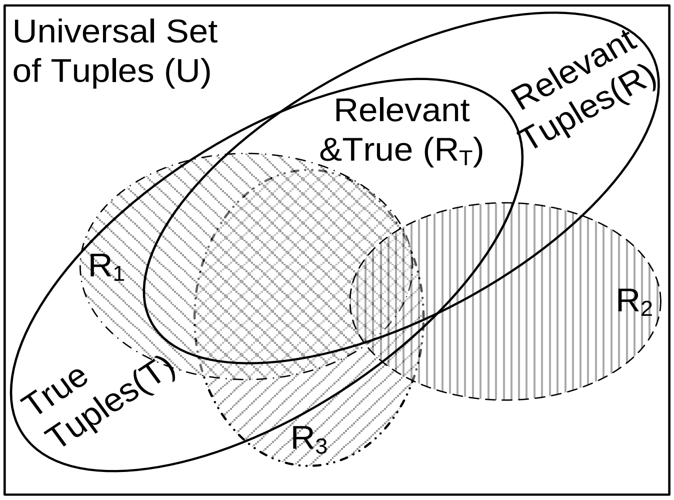


Figure 3.1: Agreement implies trust and relevance. Universal set U is the search space, R_T is the intersection of trustworthy tuple set T and relevant tuple set R (R_T is unknown). R_1, R_2 and R_3 are the result sets of three sources. Since all three result sets are the estimates of R_T, the results agreed by other result sets are likely to overlap more with R_T.

3.1 Agreement as Endorsement

In this section we show that the result set agreement is an implicit form of endorsement. In Figure 3.1 let R_T be the set of relevant and trustworthy tuples for a query, and U be the search space (the universal set of tuples searched). Let r_1 and r_2 be two tuples independently picked by two sources from R_T (i.e. they are relevant and trustworthy), and $P_A(r_1, r_2)$ be the probability of agreement of the tuples (for now think of "agreement" of tuples in terms of high degree of similarity; we shall look at the specific way agreement between tuples is measured in Section 3.4).

$$P_A(r_1, r_2) = \frac{1}{|R_T|} \tag{3.1}$$

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

$$P_A(f_1, f_2) = \frac{1}{|U - R_T|} \tag{3.2}$$

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 3.1 and 3.2 implies

$$P_A(r_1, r_2) \gg P_A(f_1, f_2) \tag{3.3}$$

For example, assume that the user issues the query *Godfather* for the Godfather movie trilogy. Three movies in the trilogy— *The Godfather I*, *II* and *III*—are thus the results relevant to the user. Let us assume that the total number of movies searched by all the databases (search space U) is 10^4. In this case $P_A(r_1, r_2) = \frac{1}{3}$ and $P_A(f_1, f_2) = \frac{1}{10^4}$ (strictly speaking $\frac{1}{10^4-3}$). Similarly the probability of three sources agreeing are $\frac{1}{9}$ and $\frac{1}{10^8}$ for relevant and irrelevant results respectively.

Let us now extend this argument for answer sets from two sources. In Figure 3.1 R_1, R_2 and R_3 are the result sets returned by three independent sources. The result sets are best effort estimates of R_T (assuming a good number of genuine sources). Typically the results sets from individual sources would contain a fraction of relevant and trustworthy tuples from R_T, and a fraction of irrelevant tuples from $U - R_T$. By the argument in the preceding paragraph, tuples from R_T are likely to agree with much higher probability than tuples from $U - R_T$. This implies that the more relevant tuples a source returns, the more likely that other sources agree with its results.

Though the explanation above assumes independent sources, it holds for partially dependent sources as well. However, the ratio of two probabilities (i.e. the ratio of probability in Equation 3.1 to Equation 3.2) will be smaller than that for the independent sources. For added robustness of the SourceRank against source dependence, in Section 3.6 we assess and compensate for the collusion between the sources.

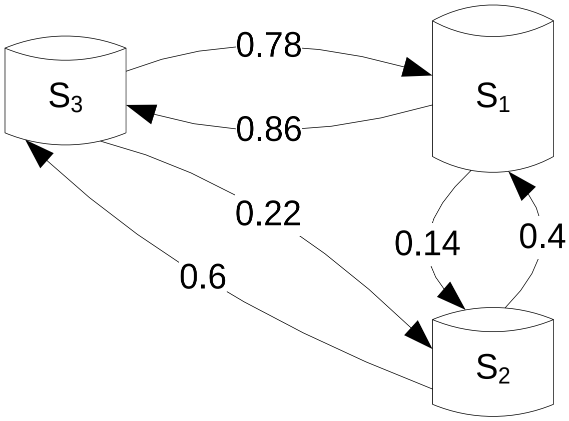


Figure 3.2: A sample agreement graph structure of three sources. The weight of the edge from S_i to S_j is computed by Equation 3.5. The weights of links from every source S_i are further normalized against sum of the weights out links of S_i.

3.2 Creating the Agreement Graph

To facilitate the computation of SourceRank, we represent the agreement between the source result sets as an agreement graph. Agreement graph is a directed weighted graph as shown in example Figure 3.2. In this graph, the vertices represent the sources, and weighted edges represent the agreement between the sources. The edge weights correspond to the normalized agreement values between the sources. For example, let R_1 and R_2 be the result sets of the source S_1 and S_2 respectively. Let $a = A(R_1, R_2)$ $(0 \leq a \leq 1)$ be the agreement between the results sets (calculated as described in Section 3.4). In the agreement graph we create two edges: one from S_1 to S_2 with weight equal to $\frac{a}{|R_2|}$; and one from S_2 to S_1 with weight equal to $\frac{a}{|R_1|}$. The semantics of the weighted link from S_1 to S_2 is that S_1 endorses S_2, where the fraction of tuples endorsed in S_2 is equal to the weight. Since the endorsement weights are equal to the fraction of tuples, rather than the absolute number, they are asymmetric.

As we shall see in Section 3.4, the agreement weights are estimated based on the results to a set of sample queries. To account for the *sampling bias* in addition to the agreement links described above, we also add *smoothing links* with small weights between every pair of vertices. These smoothing links account for the unseen samples. That is, though there is no agreement between the sampled results sets used to calculate the links, there is a non-zero probability for some of the results to agree for queries not used for sampling. This probability corresponding to unseen samples are accounted by smoothing links with small weights. Adding this smoothing probability, the overall weight $w(S_1 \to S_2)$ of the link from S_1 to S_2 is:

$$A_Q(S_1, S_2) = \sum_{q \in Q} \frac{A(R_{1q}, R_{2q})}{|R_{2q}|} \tag{3.4}$$

$$w(S_1 \to S_2) = \beta + (1 - \beta) \times \frac{A_Q(S_1, S_2)}{|Q|} \tag{3.5}$$

where R_{1q} and R_2q are the answer sets of S_1 and S_2 for the query q, and Q is the set of sampling queries over which the agreement is computed. β is the smoothing factor. We set β at 0.1 for our experiments. Empirical studies like Gleich *et al.* [33] 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 sum of the out edge weights from the vertex. This normalization allows us to interpret the edge weights as the transition probabilities for the random walk computations.

3.3 Calculating SourceRank

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

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

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

The agreement graph described above provides important guidance in selecting relevant and trustworthy sources. Any source that has a high degree of endorsement by other relevant sources is itself a relevant and trustworthy source. This transitive propagation of source relevance (trustworthiness) through agreement links can be captured in terms of a fixed point computation [10]. In particular, if we view the agreement graph as a markov chain, with sources as the states, and the weights on agreement edges specifying the probabilities of transition from one state to another, then the asymptotic stationary visit probabilities of the markov random walk will correspond to a measure of the global relevance of that source. We call this measure *SourceRank*.

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

The SourceRank thus obtained may be combined with query similarity based score of the source (please refer to Section 3.7.1.3 for details) to get the

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

(a) Tuples from first source

	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) Tuples from second source

Table 3.1: Sample tuples returned by two movies databases to the query *Godfather* are shown in Table (a) and (b). Note that the tittles and casting referring to same entity syntactically differs from each other.

final ranking score as,

$$Score = \alpha \times querySim + (1 - \alpha) \times SourceRank \tag{3.6}$$

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

3.4 Computing Agreement

If the sources are fully relational and share the same schema and values, the agreement computation between two tuples will reduce to equality between them. On the other extreme, if the sources are text databases then the agreement between two items will have to be measured in terms of textual similarity. Deep web sources present an interesting middle ground between the free-text sources in IR, and the fully-structured sources in relational databases. Hence to address challenges in agreement computation of deep web results we have to combine and extend methods from both these disciplines. Our method of computing agreement between the sources involves following three levels of similarity computations: (a) attribute value similarity (b) tuple similarity, and (c) result set similarity.

3.4.1 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, assumption of common domains rarely holds in web databases [12]. For example, the title and casting attributes of tuples referring to the same movie returned from two databases are shown in Table 3.1(a) and 3.1(b). Identifying the semantic similarity between these tuples is not straightforward, since the titles and actor lists show wide syntactic variation.

The textual similarity measures work best for scenarios involving web databases with no common domains [12]. Since this challenge of matching attribute values is essentially a name matching task, we calculate the agreement between attribute values using SoftTF-IDF with Jaro-Winkler as the similarity measure [34]. SoftTF-IDF measure is similar to the normal TF-IDF measure. But instead of considering only exact same words in two documents to calculate similarity, SoftTF-IDF also considers occurrences of similar words.

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

$$\mathcal{SIM}(v_i, v_j) = \sum_{w \in \mathcal{C}(\theta, v_i, v_j)} \mathcal{V}(w, v_i)\mathcal{V}(u, v_j)D(w, v_j) \tag{3.7}$$

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 [34]. For pure numerical

[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

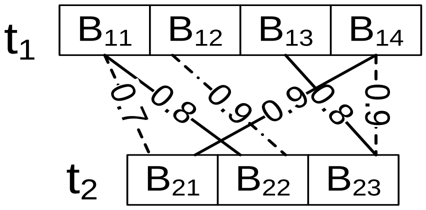


Figure 3.3: Example tuple similarity calculation. The dotted line edges denote the similarities computed, and the solid edges represent the matches picked by the greedy matching algorithm.

values (like price) we calculate similarity as the ratio of the difference of values to the maximum of the two values.

3.4.2 Tuple similarity

The tuples are modeled as a vector of bags [12]. The problem of matching between two tuples based on the vector of bags model is illustrated in Figure 3.3. 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 databases [35]. We do not assume predefined answer schema mapping, and hence reconstruct the schema mapping based on the attribute value similarities as described below.

The complexity of similarity computation between the attribute values (i.e. building edges and weights in Figure 3.3) 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 these edges, a single attribute value in t_1 may be similar to multiple attributes in t_2 and *vice versa*. The optimal matching should pick the edges (matches) such that the sum of the matched edge weights would be maximum.

$$S_{opt}(t,t') = \arg\max_{M} \sum_{(v_i \in t, v_2 \in t') \in M} \mathcal{SIM}(v_1, v_2) \tag{3.8}$$

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 problem, V is the number attribute values to be matched, and E is the number of similarity values). Since E is $O(V^2)$ for our problem the overall time complexity is $O(V^3)$.

Running time is an important factor for calculating agreement at the web scale. Considering this, instead of the $O(V^3)$ optimal matching discussed above, we use the $O(V^2)$ greedy matching algorithm as a reasonable balance between time complexity and performance. To match tuples, say t_1 and t_2 in Figure 3.3, 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 3.3. The agreement between the tuples is calculated as the sum of the similarities of the individual matched values. The two tuples are considered matching if they exceed a empirically determined threshold of similarity.

The Fellagi-Saunter record linkage model [36] suggests that the attribute values occurring less frequently are more indicative of the semantic similarity between the tuples. For example, two entities with the common title *The Godfather* are more likely to be denoting same book than two entities with common format *paperback*). To account for this, we weight the similarities between the matched attributes in the step above as

$$S(t,t') = \frac{\sum_{v_i,v_j \in M} w_{ij} \mathcal{SIM}(v_i,v_j)}{\sqrt{\sum_{v_i,v_j \in M} w_{ij}^2}} \tag{3.9}$$

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,

$$w_{ij} = log\left(\frac{\sum_k \mathcal{IDF}_{ik}}{|v_i|}\right) log\left(\frac{\sum_l \mathcal{IDF}_{jl}}{|v_j|}\right) \tag{3.10}$$

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 TFIDF.

3.4.3 Result Set Similarity

The agreement between two result sets R_{1q} and R_{2q} from two sources for a query q is defined as,

$$A(R_{1q},R_{2q}) = \arg\max_M \sum_{(t \in R_{1q}, t' \in R_{2q}) \in M} S(t,t') \tag{3.11}$$

where M is the optimal matched pairs of tuples between R_{1q} and R_{2q} and $S(t,t')$ are as calculated in Equation 3.9. Since this is again a bipartite matching problem similar to Equation 3.8, we use a greedy matching. The first tuple in R_{1q} is matched greedily against the tuple with highest match in R_{2q}. Subsequently, the second tuple in R_{1q} is matched with the most similar unmatched tuple in R_{2q} and so on. The agreement between the two result sets is calculated as the sum of the agreements between the matched tuples. The agreement thus calculated is used in the Equation 3.4.

We calculate agreement between the top-k (with $k = 5$) answer sets of the each query in the sampled set described in the subsection below. We stick to top-k results since most web information systems focus on providing best answers in the top few positions (a reasonable strategy given that the users rarely go below the top few results). The agreements of the answers to the entire set of sampling queries is used in Equation 3.4 to compute the agreement between the sources. Note that even though we used top-k answers, the normalization against the answer set size in Equation 3.4 is required, since the answer set sizes vary as some sources return less than k results to some queries.

3.5 Sampling Sources

Web databases are typically non-cooperative, i.e. they do not share the statistics of contents, or allow access to the entire data set. Thus, the agreement graph must be computed over a sampled set. In this section we describe the sampling strategy used for our experiments on web databases (see Section 3.7). For sampling, we assume only a form based query interface allowing keyword queries; similar to the query based sampling used for the non-cooperative text databases [37].

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 two hundred books from the New York Times yearly number one book listing from the year 1940 to 2007 [38]. For the sampling query set of movie domain, we use two hundred random movies from the second edition of New York Times movie guide [39].

As keyword queries for sampling, we use partial titles of the books/movies. We generate partial title queries by randomly deleting words from titles of length more than one word. The probability of deletion of a word is set to 0.5.

The use of partial queries is motivated by the fact that two sources are less likely to agree with each other on partial title queries. This is because partial titles are less constraining and thus result in a larger number of possible answers compared to full title queries. Hence agreement on answers to partial queries is more indicative of agreement between the sources 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 to be 0.5 is based on cross-validation experiments.

We perform a query based sampling of database by sending the queries to the title keyword search fields of the sources. The sampling is automated here, but we wrote our own parsing rules to parse the result tuples from the returned HTML pages. This parsing of tuples has been solved previously [29, 40, 41], and can be automated (parsing is not required for Google Base experiments as structured tuples are returned). This averaging and aggregation over number of queries is likely to increase the robustness of the overall agreement computation against the problems in linking individual records.

3.6 Assessing Source Collusion

A potential problem for applying SourceRank is that sources may make copies of themselves to boost their rankings. As the SourceRank becomes popular, collusion is likely to be more severe problem as well [42]. This is similar to the prevalence of link spam as the link analysis became a common ranking method for the surface web. Considering this, we devise a method to measure and compensate 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 for the improved robustness of SourceRank. Two issues that complicate collusion detection are (i) even non-colluding databases in the same domain may contain almost the same data. For example, many movie sources may contain all Hollywood movies. This means that two databases having similar data samples need not 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 that 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 the collusion detection is that if two sources return the same top-k answers to the queries with large number of possible answers (e.g. queries containing only stop words), they are possibly colluding. More formally, for two 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

$$E(A_k) = \begin{cases} \frac{k}{n}(1-e) & \text{if } k < n \\ (1-e) & \text{otherwise} \end{cases} \tag{3.12}$$

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

To generate a set of queries with large answer sets, we fetched a set of two hundred keywords with highest document frequencies from the crawl described in the Section 3.5. Sources are probed with these queries. The agreement between the answer sets are computed based on this crawl according to Equation 3.4. These agreements are seen as a measure of the collusion between the sources. The agreement computed between the same two sources on the samples based on genuine queries described in Section 3.5 is multiplied by $(1 - collusion)$ to get the adjusted agreement. Thus the weight of the edges in Equation 3.5 is modified in this collusion-adjusted agreement graph as,

$$w(S_1 \rightarrow S_2) = \beta + (1 - \beta) \times \frac{A_Q(S_1, S_2)(1 - collusion)}{|Q|} \tag{3.13}$$

These adjusted agreements are used for computing SourceRank for the experiments below. We also provide a standalone evaluation of collusion measure in Section 3.7.5.

3.7 Evaluations

In this section we evaluate the effectiveness of SourceRank (computed based on collusion adjusted-agreement) as the basis for domain specific source selection sensitive to relevance and trustworthiness. The top-k precision and discounted cumulative gain (DCG) of SourceRank-base source selection are compared with three existing methods: (i) Coverage based ranking used in relational databases, (ii) CORI ranking used in text databases, and (iii) Google Product search on Google Base.

3.7.1 Experimental Setup

We describe the dataset, test queries and baseline methods in our experiments in the following three sections.

3.7.1.1 Databases

We performed the evaluations in two vertical domains—sellers of books and movies (movies include DVD, Blu-Ray etc.). We used three sets of data bases— (i) a set of standalone online data sources (e.g. Amazon) (ii) hundreds of data sources collected via *Google Base* (iii) a million IMDB records [43].

The databases listed in TEL-8 database list in the UIUC deep web interface repository [44] are used for online evaluations (every source in the repository after removing non-working ones). We used sixteen movie databases and seventeen book databases from the TEL-8 repository. In addition to these, we added five video sharing databases to the movie domain and five library sources to the book domain. These out-of-domain sources are added to increase the variance in source quality. If all sources are of similar quality, different rankings do not make a difference.

Google Base is a collection of data from a large number of web databases, with an API-based access to data returning ranked results [28]. The Google Products Search works on Google Base. Each source in Google Base has a source id. For selecting domain sources, we probed the Google Base with a set of ten book/movie titles as queries. From the first 400 results to each query, we collected source ids; and considered them as a source belonging to that particular domain. This way, we collected a set of 675 book sources and 209 movie sources for our evaluations. Sampling is performed through Google Base API's as described in Section 3.5.

3.7.1.2 Test Queries

Test query sets for both book and movie domains are selected from different lists than the sampling query set, so that test and sampling sets are disjoint.

The movie and book titles in several categories are obtained from a movie sharing site and a favorite books list. We generated queries by randomly removing words from the movie/book titles with probability of 0.5—in the same way as described for the sampling queries above. We used partial titles as the test queries, since typical web user queries are partial descriptions of objects. The number of queries are used in different experiments varies between 50 to 80, so as to attain 95% confidence intervals.

3.7.1.3 Baseline Methods

Coverage: Coverage is computed as the mean relevance of the top-5 results to the sampling queries described in Section 3.5 above. For assessing the relevance of the results, we used the SoftTF-IDF with Jaro-Winkler similarity between the query and the results (recall that the same similarity measure is used for the agreement computation).

CORI: To collect source statistics for CORI [6], we used terms with highest document frequency from the sample crawl data describe in Section 3.5 as crawling queries. Callan *et al.* [37] observe that good performance is obtained by using highest document frequency terms in related text databases as queries to crawl. Similarly, we used two hundred high tuple-frequency queries and used top-10 results for each query to create resource descriptions for CORI. We used the same parameters as found to be optimal by Callan *et al.* [6]. CORI is used as the baseline, since the later developments like ReDDE [45] 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.

3.7.2 Relevance Evaluation

This section describes our empirical relevance evaluation. We give the details of manual labeling or results. Subsequently, we describe the experiments on a smaller set of online databases and a larger set of Google Base sources.

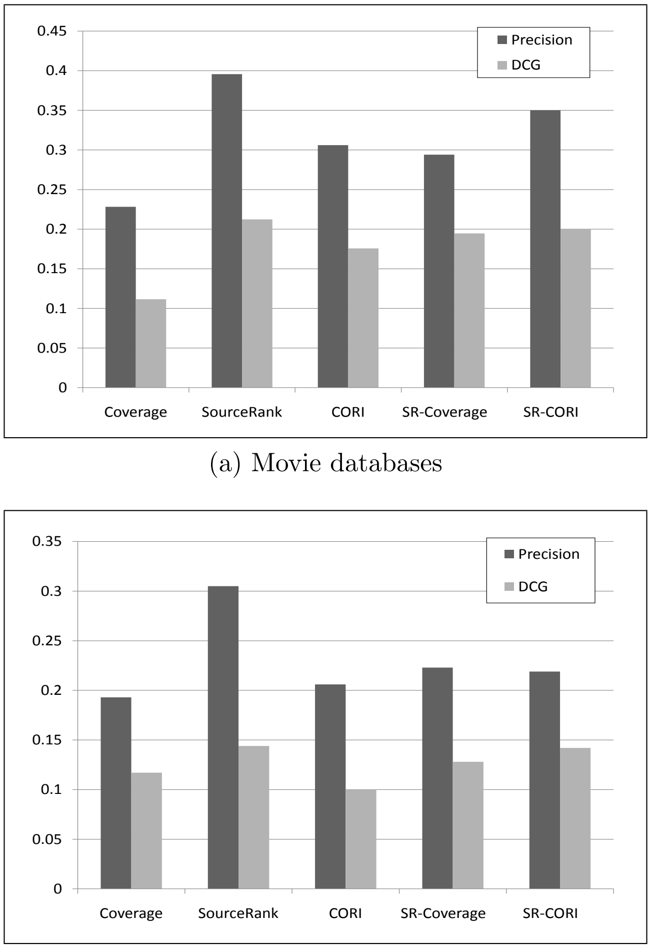
3.7.2.1 Assessing Relevance

To assess the relevance, we used randomly chosen queries from test queries described above in Section 3.7.1. These queries are issued to the top-k sources selected by different methods. The results returned are manually classified as relevant and non-relevant. The first author performed the classification of the tuples, since around 14,000 tuples were to be classified as relevant and irrelevant. The classification is simple and almost rule based. For example, assume that the query is *Wild West*, and the original movie name from which the partial query is generated is *Wild Wild West* (as described in the test query description in Section 3.7.1). 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 classified as irrelevant. Similarly for books, if the result is the queried book to sell, it is classified as relevant and otherwise it is classified as irrelevant. As an insurance against biased classification by the author, we randomly mixed tuples from all methods in a single file; so that the author did not know which method each result came from while he does the classification. All the evaluations are performed to differentiate SourceRank precision and DCG from competing methods by non-overlapping confidence intervals at a significance level of 95% or more.

3.7.2.2 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.

The results of the top-4 source selection experiments in movie and books domain are shown in Figure 3.4a and 3.4b. For both the domains, SourceRank clearly outperforms the Coverage and CORI. For the movie domain, SourceRank increases precision over Coverage by 73.0% (i.e. $((0.395 - 0.228)/_{0.228}) \times 100$) 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%. The SourceRank outperforms standalone CORI and Coverage in both precision and DCG at a confidence level of 95%. Though the primary target of the evaluation is not differentiating SourceRank and combinations, it may be worth mentioning that SourceRank outperforms the combinations at a confidence level more than 90% in most cases. Though this may be counter-intuitive at the first thought, 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. When combining again with the query-relevance based method like CORI, we are possibly over-weighting the query similarity.



(a) Movie databases

(b) Book databases

Figure 3.4: Comparison of precision and DCG of top-4 online sources selected by Coverage, SourceRank, CORI, Combination of SourceRank with Coverage (SR-Coverage) and CORI (SR-CORI) for movies and books .

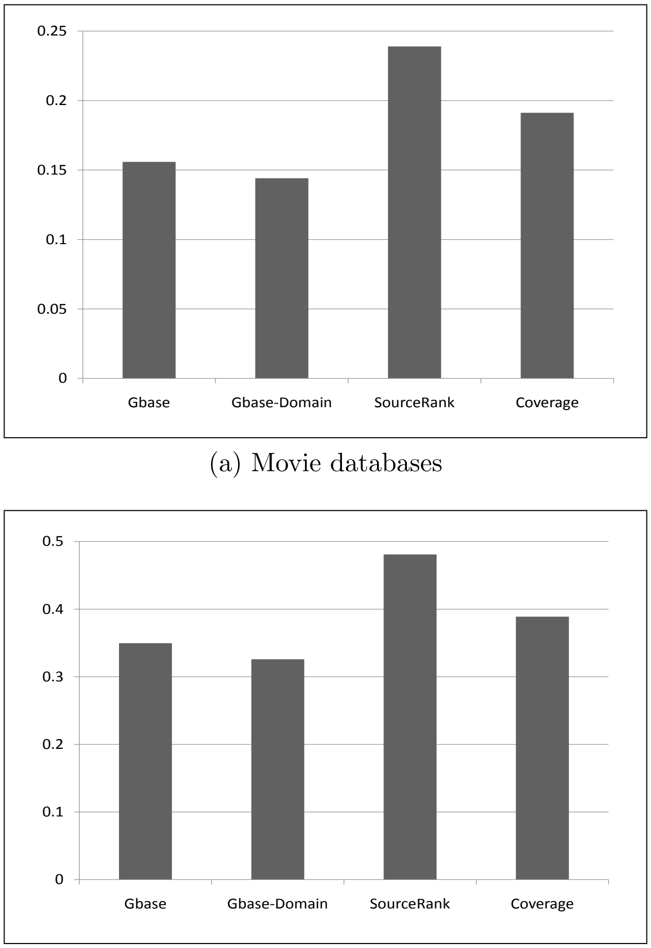
As a note on the seemingly low precision values, these are mean relevance of the top-5 results. Many of the queries used have less than five possible relevant answers (e.g. a book title query may have only paperback and hard cover for the book as relevant answers). But since the web databases always tend to return full first page of results average top-5 precision is bound to be low. For example, for a search engine always returning one relevant result in top−5, the top−5 precision will be only 0.2.

3.7.2.3 Google Base

In these experiments we tested if the precision of Google Base search results can be improved by combining SourceRank with the default Google Base relevance ranking. Google Base tuple ranking is applied on top of source selection by SourceRank and compared with 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 3.7.1. Out of these 675 sources, we selected top-67 (10%) sources based on SourceRank. Google Base is made to query only on this top-67 Sources, and the precision of top-5 tuples is compared with that of Google Base Ranking without this source selection step. Similarly for the movie domain, top-21 sources are selected. DCG is not computed for these experiments since all the results are ranked by Google Base ranking, hence ranking order comparison is not required.

In Figure 3.5a and 3.5b, the *GBase* is 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 3.5b, 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 Cov-



(a) Movie databases

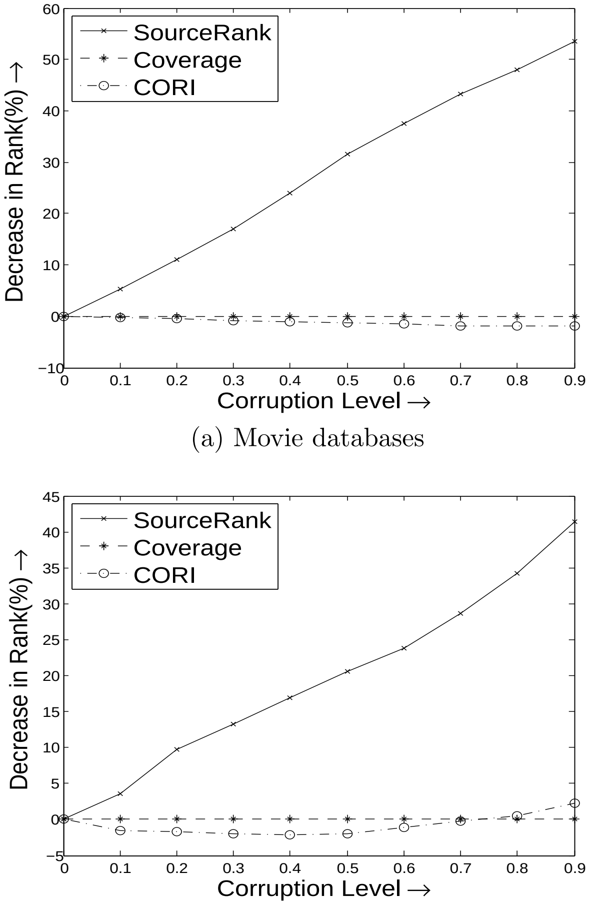
(b) Book databases

Figure 3.5: Comparison of top-5 precision of results returned by SourceRank, Google Base and Coverage for movies and books.

erage respectively. The small difference between the Google Base and Google Base-domain has low statistical significance (below 80%) hence not conclusive.

3.7.3 Trustworthiness Evaluation

In the next set of experiments, we evaluate the ability of SourceRank to eliminate untrustworthy sources. For tuples, corruption in the attribute values not specified in the query manifests as untrustworthy results, whereas mismatch in attributes values specified in the query manifests as the irrelevant results. Since the title is the specified attribute for our queries, we corrupted the at-



(a) Movie databases

(b) Book databases

Figure 3.6: Decrease in ranks of the sources with increasing source corruption levels for movies and books. The SourceRank reduces almost linearly with corruption, while CORI and Coverage are insensitive to the corruption.

tributes 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.

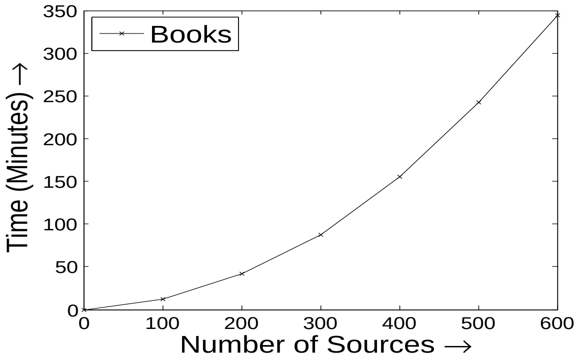


Figure 3.7: Time to compute agreement graph against number of sources.

The results of the experiments for movies and books domain are shown in Figure 3.6. 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 totally 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).

3.7.4 Timing Evaluation

We already know that random walk computation is feasible at web scale [10]. 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 3.7 shows the variation of agreement graph computation time of the 600 of the book sources from Google Base. As expected from time complexity formulae above, the time increases in second order polynomial 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; significantly reducing computation time. Further, note that the agreement graph computation is easy to parallelize. The different processing nodes can be assigned to compute a subset of agreement values between the sources. These agreement values can be computed in isolation—without 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.

3.7.5 Collusion Evaluation

In this section we perform a standalone ground truth evaluation collusion detection and the adjusted agreement described in Section 3.6. 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 [43]. We need to build two databases with varying degree 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 these 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

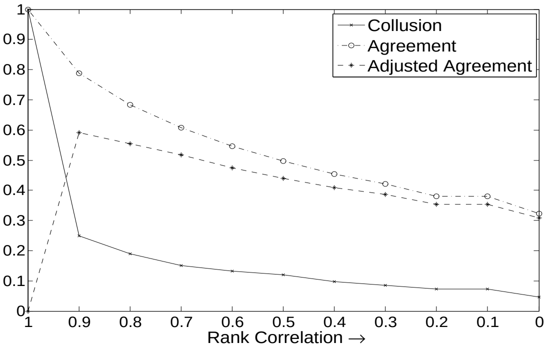


Figure 3.8: Variation of Collusion, Agreement and Adjusted Agreement with rank correlations. Adjusted Agreement is $Agreement \times (1 - collusion)$.

(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 3.8 shows the variation of collusion, agreement, and adjusted agreement with the correlation of the two databases. The correlation is progressively reduced from left to right. At the left, they are complete mirrors with the same ranking and data, and as we go right, the rank correlation decreases. As we observe in the graph, when the databases have the same rankings, the collusion and agreements are the same, making the adjusted agreement zero. This clearly makes the adjusted agreement between mirrors (databases with the same data and ranking) and near mirrors zero. 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 3.6. The method penalizes mirrors and near mirrors, whereas genuine agreement between the sources is kept intact.

3.8 Chapter Summery

The sheer number and uncontrolled nature of the sources in the deep web leads to significant variability among the sources, and necessitates a more robust measure of relevance sensitive to source popularity and trustworthiness. To this end, we proposed SourceRank, a global measure derived solely from the degree of agreement between the results returned by individual sources. SourceRank plays a role akin to PageRank but for data sources. Unlike PageRank however, it is derived from implicit endorsement (measured in terms of agreement) rather than from explicit hyperlinks. For added robustness of the ranking, we assess and compensate for the source collusion while computing the agreements. Our comprehensive empirical evaluation shows that SourceRank improves relevance sources selected compared to existing methods and effectively removes corrupted sources. We also demonstrated that combining SourceRank with Google Product search ranking significantly improves the quality of the results.

Chapter 4

Topic-Sensitive Source Ranking

Deep web sources may contain data from multiple topics (domains). For such multi-domain sources, the quality of the data in different domains may vary significantly. For example, Amazon may return high quality results for books, but may return low quality results for furniture. The quality of a source specific to a topic is best indicated by the agreement by sources in the domain. Haveliwala [13] has shown that the topic-specific endorsement improves search for the surface web. This consideration is likely to be more significant for the deep web, since sources contain records very specific to domains (e.g. books, movies etc.). Hence to customize SourceRank for the multi-domain deep web, we introduce topic sensitive SourceRank (TSR). In this chapter we describe the sampling and computation for TSR—SourceRank computed primarily based on the agreement by the sources in the same topic. We start by describing topical sampling of sources and TSR computation. Section 4.3 describes the soft-classification of user queries into multiple domains. Subsequently we describe the system architecture, and empirically compare TSR with existing measures and topic-oblivious SourceRank.

4.1 Topical Sampling of Sources

Unlike the SourceRank, the sampling for the topical SourceRank is domain specific. We used different sampling queries for different domains. For example, the TSR for movies is computed based on the movie sampling queries. All other details of sampling is similar to the SourceRank sampling described in Section 3.5

For TSR computations we used sources spanning across four domains—Books, Movies, Cameras and Music. Sampling method is same as described for SourceRank in Section 3.5. Sampling queries are from New York Times best sellers [38] (books), Open Directory DVD Listing [46] (movies), pbase.com [47] (cameras), and top-100 albums in 1986-2010 [48] (music)

Similar to the SourceRank sampling, words are deleted from titles with 0.5 probability to get the partial key word queries. All these queries are sent to every source and *top-k* (we used $k = 5$) answers returned are collected. Note that the sources are not explicitly classified into topics. The idea is that if a source gives high quality answers for queries in a topic, the other sources in the topic are likely to agree with that source. After tuples are retrieved, we compute the agreement between the sources as described below.

4.2 Computing Topic Sensitive Ranking

For the Topic-sensitive SourceRank (TSR), a source-quality score is computed for each topic of the source. We compute the source quality score for a topic based solely on the source crawls corresponding to the sampling queries of the topic. For example, for computing movie TSRs, we compute the agreement graph (described in Section 3.2) based on the crawl obtained by using the movie sampling queries described above in Section 4.1. After generating the agreement graph, source quality score for this topic are computed based on the static visit probability of a weighted Markov random walk on the graph as described in Section 3.3.

The acceptability of computation timings of TSR is directly inferable from the computation of the SourceRank. The first step of computing TSR—computing the agreement graph—is shown to be scalable in Section 3.7.4. The only difference for the TSR is that we have multiple source graphs, one corresponding to each topic. Hence the total time to compute the graphs increases linearly with the number of topics. The random walk computation is widely used [10] and known to be scalable. Besides, note that the TSR computation is offline, and does not add to the valuable query time. We do not perform separate timing experiments for TSR.

Depending on the target topic of the query, we need to use the right topic TSRs to select the best sources. For example, we need to select sources ranking higher in the movie TSR for a movie query. Realistically, the membership of a query in a topic will be probabilistic. The section below describes combining topic TSRs depending on the probability of membership of the query in different topics.

4.3 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 sources. For example, we need to select source based on the movie TSR for a movie query like "The Godfather Trilogy". The first step in query processing is to identify the query-topic i.e. the likelihood of the query belonging to topic-classes. We formulate this as a soft-classification problem. For a user query q and a set of representative topic-classes $c_i \in C$, the goal is to find the probability of topic membership of q in each of these topics c_i. A Naïve Bayes Classifier (NBC) is used for this topical query classification. We describe training data and classification approach in the sections below.

4.3.1 Training Data

For topic-descriptions to train our classifier, we use query based sampling similar to the sampling described in Section 4.1. Same set of sampling methods and list of queries have been used. But instead of generating partial queries by deleting words randomly, we use full titles as queries. Full title query crawls is less noisy and is found to be giving better classification accuracy.

4.3.2 Classification

Realistically, query classification to domains will be probabilistic at best, since deterministically classifying queries to a particular domain 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,

$$P(c_i|q) = \frac{P(q|c_i)P(c_i)}{P(q)} \propto P(c_i)\prod_j P(q_j|c_i) \tag{4.1}$$

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,

$$P(c_i|q) \propto \prod_j P(q_j|c_i) \tag{4.2}$$

$P(q_j|c_i)$ is computed 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 TSR_{kq} specific to the query is given by,

$$TSR_{kq} = \sum_i P(c_i|q)TSR_{ki} \tag{4.3}$$

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

4.4 System Architecture

Figure 4.1 provides an overview of our system. Similar to the SourceRank system, it consists of two main parts. An offline component which uses the

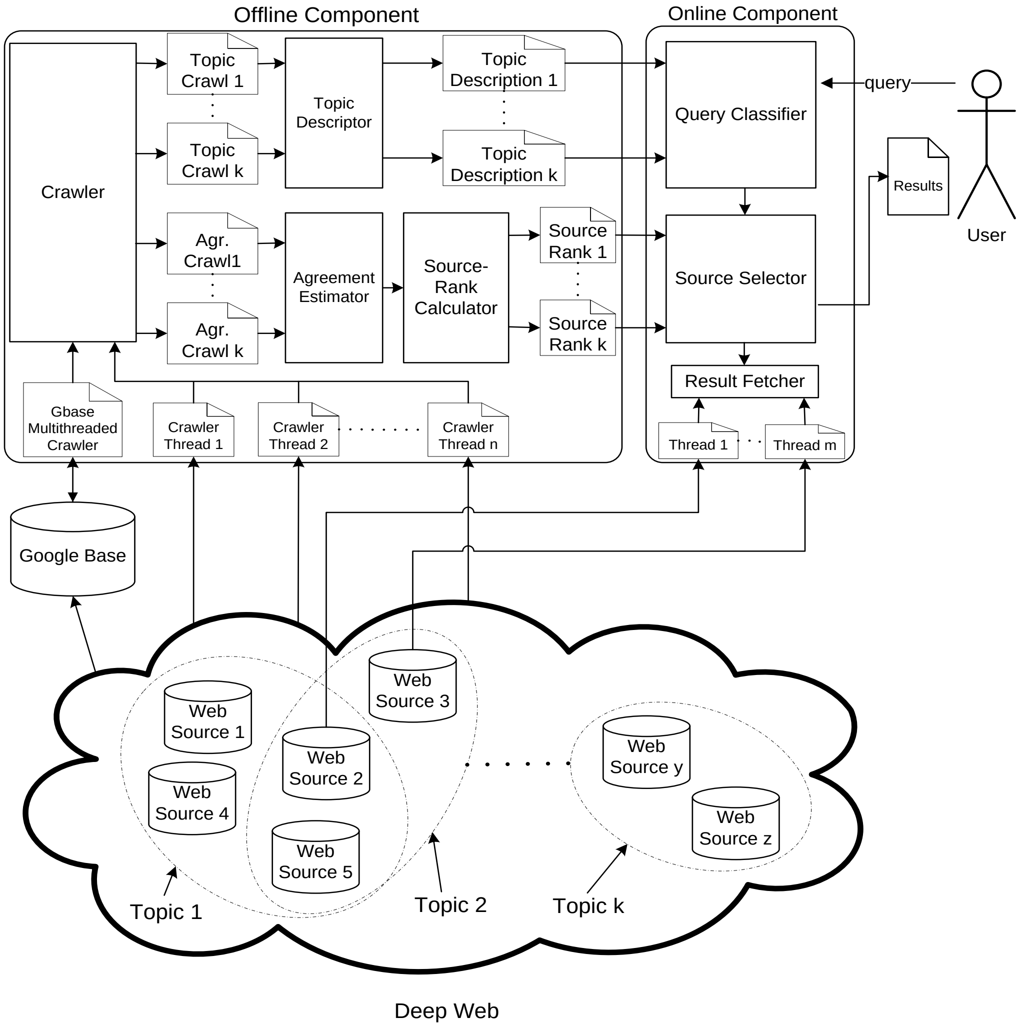


Figure 4.1: Multi-topics deep web integration system combining online query classification and TSR based source selection.

crawled data for computing topic-sensitive SourceRanks and topic-descriptions. The online component consists of a classifier which performs user query-classification using the topic-descriptions. The source selector uses the query-classification information to combine TSRs in order to generate query specific ranking of sources.

The main difference with the SourceRank system described in Section 5.4.1 are the topical-crawling and query classification components in the architecture. The topic crawlers get the samples required for the query classifier training. The agreement crawlers perform the topic specific sampling required for computation of topical source graphs. At the query time, the query classifier classify the query to topics, and source selector ranks the sources by mixing the scores corresponding the query domains.

4.5 Experimental Evaluation

We compared the precision of TSR with other source selection methods. The experiments are performed on a large set of multi-topic deep-web sources. These sources correspond to four representative topic classes - camera, book, movie, and music.

4.5.1 Source Data Set

The deep-web source data was collected from Google Base. For selecting sources for multi-topic deep-web environment, we probed Google Base with a set of 40 queries. These 40 queries contained a mix of camera model names, book, movie, and music album titles. From the first 200 results of each query, we collected the source identifiers and considered them as a source belonging to our multi-topic deep web environment. We collected a total of 1440 deep web sources: 276 camera, 556 book, 572 movie, and 281 music sources.

4.5.2 Test Queries

Test query set contained a mix of queries from all four topic-classes. Test queries were selected such that there is no overlap with the sampling queries. Queries were generated by randomly removing words from camera names, book, movie and music album titles with probability 0.5, similar to the sampling queries described in Section 4.1. Number of test queries are varied for different topics to achieve statistically significant (0.95) difference with baselines.

4.5.3 Baseline Source Selection Methods

TSR is compared with agreement based and query similarity based source selection methods. The agreement based methods consider the source agreement, and hence the trustworthiness and relevance of the sources are taken into account. On the other hand, pure query similarity measures like CORI [6] assesses the source quality based on similarity of content with the user query; hence agnostic to the trust and importance. The CORI and the Undifferentiated SourceRank described below may be considered as the alternative approaches to multi-topic search derived from the existing methods.

The baseline methods used are:

Undifferentiated SourceRank (USR): The USR is computed without differentiating between the domains similar to the single-domain SourceRank. A single agreement graph is created for the entire set sources; using the sampling queries for all the domains. On this undifferentiated graph, a single source quality scores for each source is computed.

CORI: We compared with standalone CORI (described in Section 3.7.1.3) as well as evaluated combination of CORI with agreement based source selection.

Google Base: We compared with two-versions of Google Base. Stand along 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 3.7.2).

4.5.4 Relevance Evaluation

Evaluation is similar to the SourceRank evaluation on Google Base sources. Using our source selection strategies, we selected *top-k* sources for every test query and restricted Google Base query only on these *top-k* sources. We experimented with three different values of *k*—*top-10* sources, *top-5%* and *top-10%* sources—and found that best precision was obtained for *k=10*. We used Google Base's tuple ranking for ordering the resulting tuples and return *top-5* tuples in response to test queries. After ranking the tuples, the methods can be directly compared with each other.

For assessing the relevance, we used the test queries described above. The queries were issued to *top-k* sources selected by different source selection methods. The *top-5* results returned were manually classified as relevant or irrelevant. The classification of query to relevant and irrelevant is performed as described for SourceRank evaluation in Section 3.7

4.5.4.1 Comparison with Query Similarity

We compared TSR with the baselines described above. Instead of using standalone TSR, we combined TSR with query similarity based CORI measure. We experimented with different values of weighted combination of CORI and TSR, and found that $TSR \times 0.1 + CORI \times 0.9$ gives best precision. For rest of this section we denote this combination as $TSR(0.9)$. Note that the higher weightage of CORI compared to TSR is to compensate for the fact that TSR scores have much higher dispersion compared to CORI scores, and not an indication of relative importance of these measures.

Our first set of experiments compare precision of TSR(0.1) with the query similarity based measures i.e. CORI, Google Base and Google Base Dataset. The results for individual domains and the aggregate mean across the domains are illustrated in Figure 4.2. Note that for every domain as well as for the aggregate the improvement in precision by TSR(0.1) considerable as the precision improves up to 85% over baselines.

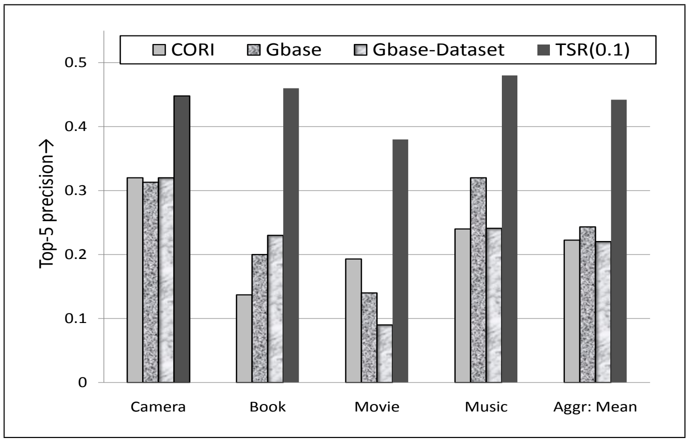


Figure 4.2: Comparison of top−5 precision of TSR(0.1) ($TSR \times 0.1 + CORI \times 0.9$) with query similarity based CORI and Google Base for different domains, and aggregate mean precision across the domains

4.5.4.2 Comparison with Agreement

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. Note that this combination is isomorphic to the linear combination of domain oblivious static PageRank and query similarity for the surface web [10].

The results for individual domains and the mean aggregate are illustrated in Figure 4.3. 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 statistical 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 the ranking towards book domain. Further, we analyzed comparable performance of domain independent USR and domain specific USR(0.1) for three

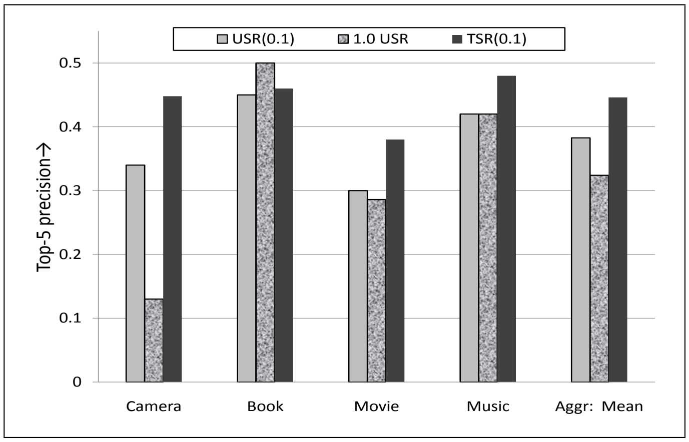


Figure 4.3: Comparison of top−5 precision of TSR(0.1) ($TSR \times 0.1 + CORI \times 0.9$) with agreement based USR and USR(0.1) ($0.1 \times USR + 0.9 \times CORI$)

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 USR.

4.6 Chapter Summery

We attempted multi-topic source selection sensitive to trustworthiness and importance for the deep web. Although SourceRank is shown to be effective in solving this problem in single topic environments, there is a need for extending SourceRank to multiple-topics. We introduced topic-sensitive SourceRank (TSR) as an efficient and effective technique for evaluating source importance in a multi-topic deep web environment. We combined TSR source selection with a Naïve Bayes Classifier for queries to build our final multi-topic deep web search system. Our experiments on more than thousand sources spanning across multiple topics show that a TSR-based source selection is highly effective in extending SourceRank for multi-topic deep web search. TSR is able to significantly out-perform query similarity based retrieval selection models.

Comparison with agreement-based source selection models showed that TSR improves precision over topic oblivious SourceRank.

Chapter 5

Ranking Results and System Architecture

For the end-to-end deep web integration and search, the returned results by selected sources have to be combined and re-ranked. Given the open and adversarial nature of the deep web sources, this re-ranking must be prepared to go beyond merging of different rank lists. Otherwise, sources may manipulate their rankings to improve the global rankings of their-own results, similar to the current search engine marketing methods. More generally, the search engine ranking ideally be independent of any parameters easily manipulable by the sources to be robust. To support this, we formulate a result ranking method—namely *TupleRak*—based on the agreement analysis.

5.1 Building Result Agreement Graph

We compute the result quality at the query time. Query time computation increases the search response time, compared to pre-computing quality for the entire search space of records (i.e. similar to the surface web search). However, unlike the surface web, a centralized index and pre-computing is infeasible for the deep web. Number of difficulties including hardness of crawling the full data set of non-cooperative sources, size of the deep web amounting to many times of the surface web [2], and the dynamic content make pre-computation infeasible.

We fetch the top$-k$ results (we used $k = 5$ for our system and experiments) from the selected sources for ranking. A primary idea for ranking sensitive to importance is the basic voting by counting number of sources returning each tuple. But this simple voting is infeasible for the deep web due to the noncommon domain problem illustrated in Figure 3.1. Hence we compute the agreement between the tuples as described in Section 3.4. We represent the agreement between the tuples as a graph with individual results as vertices. Note that 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.

5.2 Computing Ranking Scores

In the result-agreement graph, as simple ranking is in the order of first order agreements—i.e. the sum of the in-degrees of the tuples. Stepping one level deeper, second order agreement considers the common friends two tuples have. As we compute higher and higher order agreements, the accuracies as well as the computation timings increase. Since the result ranking is at the query time, lower computation time is important. We empirically compared precisions and convergence of second order agreement and random walk. For fifty test queries, the mean number of iterations to converge for random walk was found to be 16.4 (note that second order agreement takes two iterations). The difference in precision between the two was statistically insignificant (significance levels less than 0.5). Hence we use second order agreement for reduced computation time.

To describe the computation of the second order agreement, 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). Semantically second order agreement captures not just that the two tuples are agreeing, but also that they have common friends (friends are the tuples agreeing with a tuple).

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 tuples are ranked in the order of r_i.

5.3 Evaluations

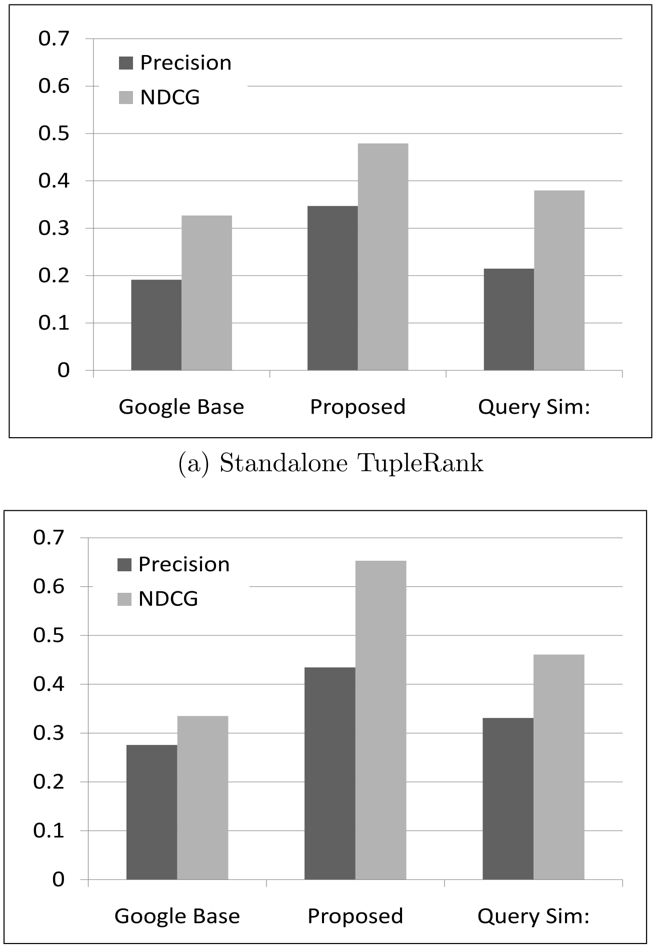
We compared the precision and trustworthiness of the result ranking with existing methods and systems. We start by evaluating standalone result ranking. Further, since result ranking will be used in conjunction with the SourceRank in real systems, we evaluate the residual increase in precision by the result ranking in addition to the improvement by SourceRank.

We used 209 movie sources in Google Base described in Section 3.7.1 for these experiments. The creation of the test query set and the labeling of the results as relevant and irrelevant are performed in the same manner as described in Section 3.7.1 as well. 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 3.4). (ii) the default relevance ranking of Google Base.

5.3.1 Relevance Results

The relevance improvements of the standalone result ranking, and in combination with SourceRank are evaluated in separate experiments. Sufficient number of queries 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 5.1a, 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 5.1a. Precision is improved by 81% over Google Base and 61% over query similarity; and



(a) Standalone TupleRank

(b) SourceRank combined with TupleRank

Figure 5.1: Comparison of top-5 precisions and NDCG of TupleRank, Query Similarity, and Google Base (a) without source selection. (b) with SourceRank based source selection.

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 found to be 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 [49]. The ranking is applicable for other values of k as well. One factor in fixing k is that larger k will increase the number of tuples to be ranked, thus increasing the ranking

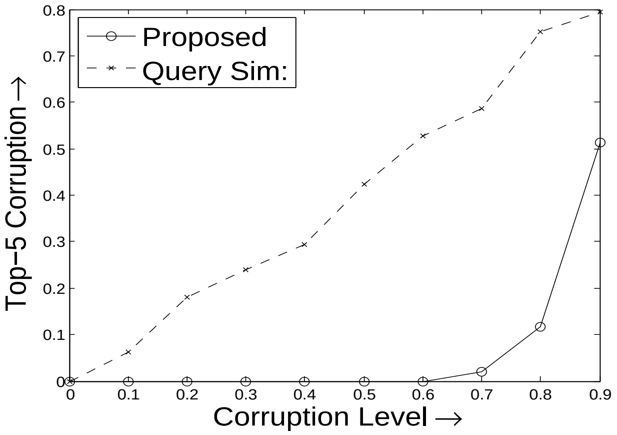


Figure 5.2: Corruption of top-5 results of the proposed result ranking and query similarity with increasing result corruption levels.

time. Another factor is the number of sources searched. As the number of sources increases, fetching fewer top results from each source is sufficient to compose a combined rank list in general. 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 5.1b, relevance is improved over the 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 5.1a).

5.3.2 Trust Results

Similar to the trust evaluation for the SourceRank described in Section 3.7.3, we corrupted a randomly selected subset of tuples by replacing attributes not specified in the query. After corrupting, tuples are ranked using Query Similarity and the proposed ranking. Robustness to corruption of ranking

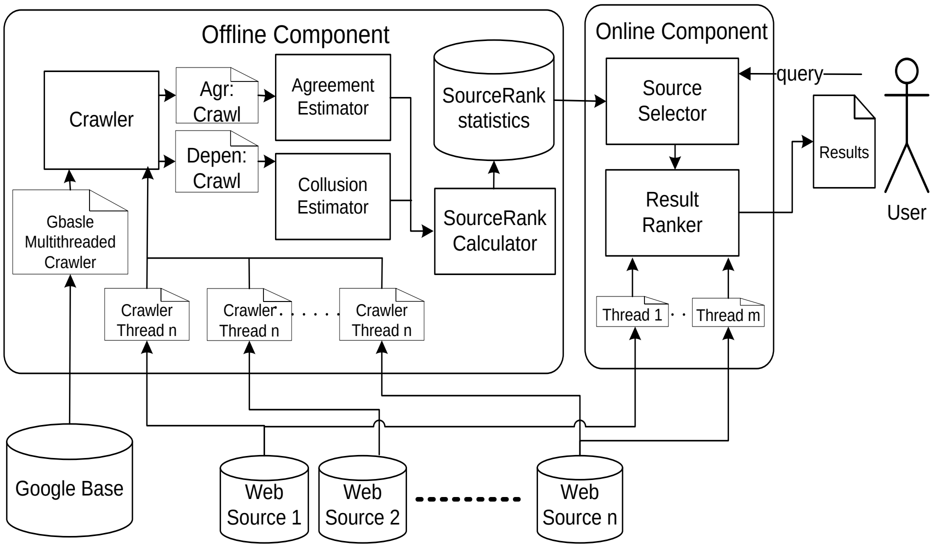
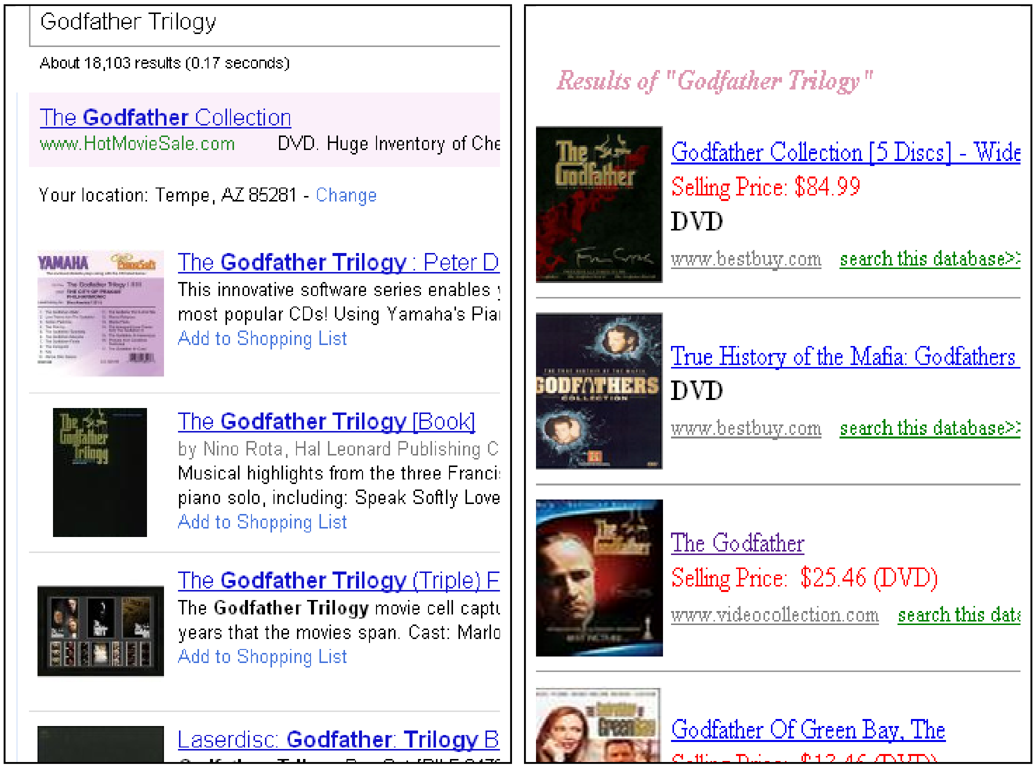


Figure 5.3: System Architectural Diagram. The online component contains processing steps at query time. Both the crawling and search are parallelized. (URL of the system is http://factal.eas.asu.edu).

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 5.2. 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. On the other hand, proposed result ranking is highly robust to corruption, as all corrupted tuples are removed until 70% of the results are corrupted. At higher levels, 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 corruption level one, any ranking method will have all the top-5 tuples corrupted).

5.4 Factal System

The proposed source and result rankings are implemented in a vertical search engine namely *Factal* (URL: http://factal.eas.asu.edu/). Sources are selected



Godfather Trilogy

About 18,103 results (0.17 seconds)

The Godfather Collection
www.HotMovieSale.com DVD. Huge Inventory of Che

Your location: Tempe, AZ 85281 - Change

The Godfather Trilogy : Peter D
This innovative software series enables y
most popular CDs! Using Yamaha's Pia
Add to Shopping List

The Godfather Trilogy [Book]
by Nino Rota, Hal Leonard Publishing C
Musical highlights from the three Franci
piano solo, including: Speak Softly Love
Add to Shopping List

The Godfather Trilogy (Triple) F
The Godfather Trilogy movie cell captu
years that the movies span. Cast: Marlo
Add to Shopping List

Laserdisc: Godfather: Trilogy B

Results of "Godfather Trilogy"

Godfather Collection [5 Discs] - Wide
Selling Price: $84.99
DVD
www.bestbuy.com search this database>>

True History of the Mafia: Godfathers
DVD
www.bestbuy.com search this database>>

The Godfather
Selling Price: $25.46 (DVD)
www.videocollection.com search this data

Godfather Of Green Bay, The

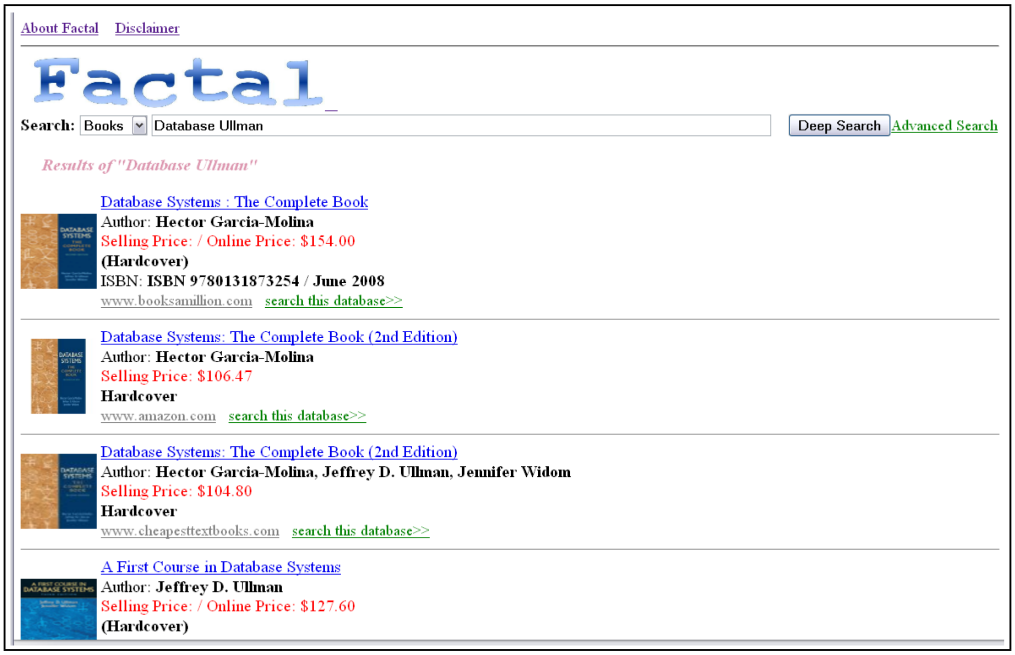
(a) Google product search (b) Factal

Figure 5.4: Comparison of results to the query *Godfather Trilogy* from (a) Google Product Search and (b) Factal. None of the top results of Google Products refer to the classic Godfather, whereas many results in Factal including top result are correct.

using the SourceRank and the results are ranked using the proposed result ranking.

5.4.1 Factal Architecture

The system shown in Figure 5.3 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, retrieve, and rank the results at query time. Factal 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 in the



Results of "Database Ullman"

Database Systems : The Complete Book
Author: **Hector Garcia-Molina**
Selling Price: / Online Price: $154.00
(Hardcover)
ISBN: **ISBN 9780131873254 / June 2008**
www.booksamillion.com search this database>>

Database Systems: The Complete Book (2nd Edition)
Author: **Hector Garcia-Molina**
Selling Price: $106.47
Hardcover
www.amazon.com search this database>>

Database Systems: The Complete Book (2nd Edition)
Author: **Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom**
Selling Price: $104.80
Hardcover
www.cheapesttextbooks.com search this database>>

A First Course in Database Systems
Author: **Jeffrey D. Ullman**
Selling Price: / Online Price: $127.60
(Hardcover)

Figure 5.5: Sample book search results for the query *Database Ullman* in Factal

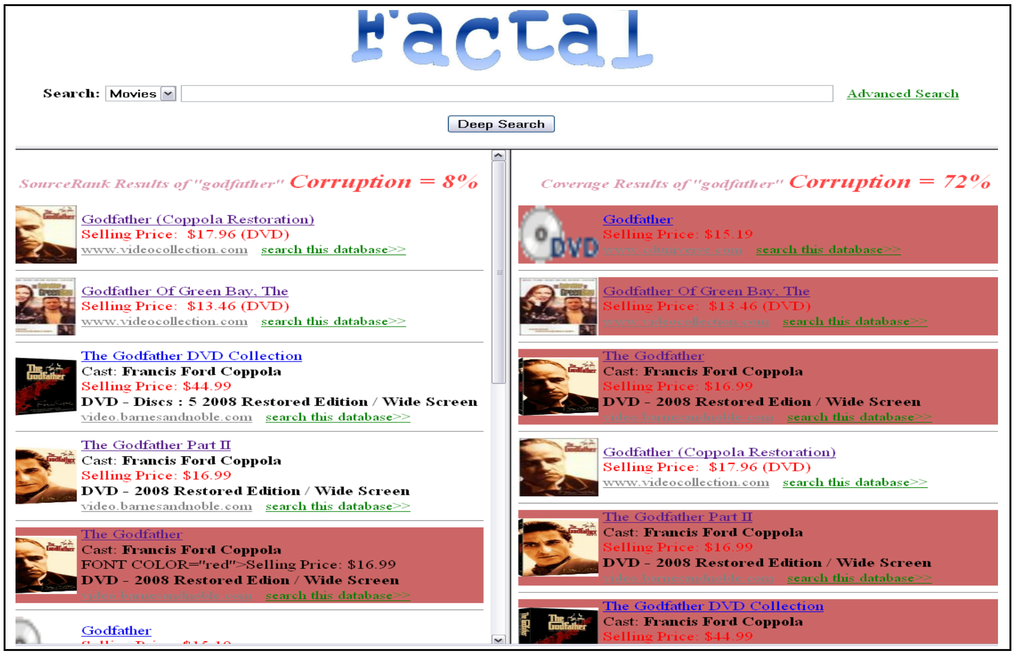
Google Base[1]. Sources are crawled using sampling method described in Section 3.5. For online sources one thread per data base is used for crawling, and for Google Base we used forty threads (maximum acceptable for Google Base).

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 re-ranked using the proposed result ranking.

5.4.2 Demonstration Scenarios

We demonstrate effectiveness of Factal by multiple screenshots. This screenshots include the Factal search results as well as comparison results from our

[1] Google Base API was shutdown lately, Factal search only online databases now.



SourceRank Results of "godfather" ***Corruption = 8%***

Godfather (Coppola Restoration)
Selling Price: $17.96 (DVD)
www.videocollection.com search this database>>

Godfather Of Green Bay, The
Selling Price: $13.46 (DVD)
www.videocollection.com search this database>>

The Godfather DVD Collection
Cast: **Francis Ford Coppola**
Selling Price: $44.99
DVD - Discs : 5 2008 Restored Edition / Wide Screen
video.barnesandnoble.com search this database>>

The Godfather Part II
Cast: **Francis Ford Coppola**
Selling Price: $16.99
DVD - 2008 Restored Edition / Wide Screen
video.barnesandnoble.com search this database>>

The Godfather
Cast: **Francis Ford Coppola**
FONT COLOR="red">Selling Price: $16.99
DVD - 2008 Restored Edion / Wide Screen
video.barnesandnoble.com search this database>>

Godfather

Coverage Results of "godfather" ***Corruption = 72%***

Godfather
Selling Price: $15.19
www.cduniverse.com search this database>>

Godfather Of Green Bay, The
Selling Price: $13.46 (DVD)
www.videocollection.com search this database>>

The Godfather
Cast: **Francis Ford Coppola**
Selling Price: $16.99
DVD - 2008 Restored Edion / Wide Screen
video.barnesandnoble.com search this database>>

Godfather (Coppola Restoration)
Selling Price: $17.96 (DVD)
www.videocollection.com search this database>>

The Godfather Part II
Cast: **Francis Ford Coppola**
Selling Price: $16.99
DVD - 2008 Restored Edition / Wide Screen
video.barnesandnoble.com search this database>>

The Godfather DVD Collection
Cast: **Francis Ford Coppola**
Selling Price: $44.99

Figure 5.6: Comparing trustworthiness of result of SourceRank and baseline methods. The corrupted results are marked as red based on the ground truth.

demonstration system [50]. The screenshots demonstrate improved precision, trustworthiness and importance of the results.

5.4.2.1 Precision and Importance

Figure 5.4 shows the comparison of Factal results with Google Product search. As we mentioned in the introduction, the current deep web search solution has the problem of showing trivial results at the top. In Figure 5.4, none of the top results by Google product search refers to the classic godfather movie or book. On the other hand the top factal results refer to the classic godfather results. This is a direct implication of the fact that the SoruceRank and the proposed result ranking are capable of ranking important results high.

Screenshot in Figure 5.5 shows another example from books domain in Factal. The query *Database Ullman* returns the database book from multiple vendors. Note that even though the diversity is not explicitly considered in the ranking, the results are diverse due to the nature of the search space. Though the titles repeat, these are different vendors providing different prices. This redundancy is beneficial for applications like comparison shopping.

5.4.2.2 Trustworthiness

Figure 5.6 illustrates trustworthiness of the proposed source and result ranking. We set up databases using tuples crawled from Google Base, and corrupted them to varying degrees. Subsequently, we compute SourceRank, Coverage and CORI ranks for each of the databases, and compare the search results from each method. The screenshot shows the layout of the results presented. The corrupted tuples are marked with red background, for an easy interpretation. The left pane shows the results from SourceRank and right pane shows the results from CORI or Coverage—as selected by the user. The corruption levels of the results are displayed separately in addition to the color coding.

The system response time is found to be in the acceptable. For the integrated online search—where the queries are routed to the selected online databases—the responses times were found to be less than a second in most cases. Thus the Factal system demonstrates the feasibility of the integration approach in the deep web, in addition to the effectiveness of the proposed source and result ranking in assessing the trustworthiness and importance of sources and results.

5.5 Chapter Summery

We address the problem of ranking the results returned by the selected sources. Similar to the SourceRank, we propose a method (TupleRank) to rank the results based on the second order agreement. Our evaluations show that the method is effective in capturing the importance and trustworthiness of the results. TupleRank significantly improves both precision and trustworthiness of the results compared to the existing ranking methods.

We implement an end-to-end deep web integration system Factal incorporating both SourceRank and TupleRank. The system architecture and sample results are presented, along with the comparison with the existing systems. Result samples comparing both trustworthiness and relevance of the system are presented. The examples clearly demonstrate the effectiveness of the proposed source and result ranking methods, in addition to the feasibility of an integration based search in the deep web.

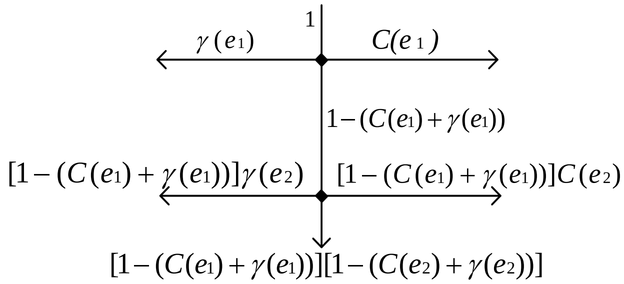
Chapter 6
Ad Ranking

Having described the deep web ranking in the preceding chapters, we consider ranking of sponsored search results. We start by deriving an optimal ranking function based on the user click model. Next, we generalize the ranking to both ads and documents. A taxonomy of rankings based on the specific assumptions on the click models and utilities are presented. Subsequently we associate a pricing with the ranking, designing a complete auction mechanism. We prove existence of a Nash equilibrium in pure strategies for the proposed mechanism. The properties of the equilibria are analyzed and compared with the VCG mechanism. Subsequently we analyze the problem of optimizing ranking considering diversity of the results. We prove that the optimal ranking considering diversity is intractable even for basic formulations of diversity. Finally we run a number of simulation experiments to quantify the difference in profits by the proposed ranking. The experiments suggest considerable profit improvements by the proposed ranking, and confirm the profit predictions by our analysis.

6.1 Click Model

Ranking functions attempt to optimiz utilities based on the click model of users. For our ranking, we assume a basic user click model in which the web user browses the entity list in ranked order, as shown in Figure 6.1. At every result entity, the user may:

1. Click the result entity with *perceived relevance* $C(e)$. We define the perceived relevance as the probability of clicking the entity e_i having seen



1
$\gamma(e_1)$
$C(e_1)$
$1-(C(e_1)+\gamma(e_1))$
$[1-(C(e_1)+\gamma(e_1))]\gamma(e_2)$
$[1-(C(e_1)+\gamma(e_1))]C(e_2)$
$[1-(C(e_1)+\gamma(e_1))][1-(C(e_2)+\gamma(e_2))]$

Figure 6.1: User click model. The labels are the view probabilities and e_i denotes the entity at the i^{th} position

e_i i.e. $C(e_i) = P(click(e_i)|view(e_i))$. Note that the Click Through Rate (CTR) defined in ad placement is the same as the perceived relevance defined here [16].

2. Abandon browsing the result list with *abandonment probability* $\gamma(e_i)$. $\gamma(e_i)$ is defined as the probability of abandoning the search at e_i having seen e_i. i.e. $\gamma(e_i) = P(abandonment(e_i)|view(e_i))$.

3. Go to the next entity in the result list with probability $[1-(C(e_i)+\gamma(e_i))]$

The click model can be schematically represented as a flow graph in Figure 6.1. Labels on the edges refer to the probability of the user traversing them. Each vertex in the figure corresponds to a view epoch (see below), and the flow balance holds at each vertex. Starting from the top entity, the probability of the user clicking the first ad is $R(e_1)$ and probability of him abandoning browsing is $\gamma(e_1)$. The user goes beyond the first entity with probability $1 - (R(e_1) + \gamma(e_1))$ and so on for the subsequent results.

In this model, we assume that the parameters—$C(e_i)$, $\gamma(e_i)$ and $U(e_i)$—are functions of the entity at the current position i.e. these parameters are independent of other entities the user has already seen. We recognize that this assumption is not fully accurate, since the users decision to click the current item or leave search may depend not just on the current item but rather all the items he has seen before in the list. We stick to the assumption for the optimal ranking analysis below, since considering mutual influence of ads can lead to combinatorial optimization problems with intractable solutions. We will show that even the simplest dependence between the parameters will indeed lead to intractable optimal ranking in Section 6.5.

Though the proposed model is intuitive enough, we would like to mention that the model is also confirmed by the recent empirical click models. For example, the General Click Model (GCM) by Zhu *et al.* [18] is based on the same basic user behavior. The GCM is empirically validated for both search results and ads [18]. Further, other click models are shown to be special cases of GCM (hence special cases of the model used in this dissertation). Please refer to Zhu *et al.* [18] for a detailed discussion. These previous works avoids the need for separate model validation, as well as confirm feasibility of the parameter estimation.

6.2 Optimal Ranking

Based on the click model, we formally define the ranking problem and derive optimal ranking in this section. The problem may be state as,

Choose the optimal ranking $E_{opt} = \langle e_1, e_2, .., e_N \rangle$ *of* N *entities to maximize the expected utility*

$$E(U) = \sum_{i=1}^{N} U(e_i) P_c(e_i) \tag{6.1}$$

where N *is the total number of entities to be ranked.*

For the browsing model in Figure 6.1, the click probability for the entity at the i^{th} position is,

$$P_c(e_i) = C(e_i) \prod_{j=1}^{i-1} [1 - (C(e_j) + \gamma(e_j))] \tag{6.2}$$

Substituting click probability P_c from Equation 6.2 in Equation 6.1 we get,

$$E(U) = \sum_{i=1}^{N} U(e_i)C(e_i) \prod_{j=1}^{i-1} [1 - (C(e_j) + \gamma(e_j))] \tag{6.3}$$

The optimal ranking maximizing this expected utility can be shown to be a sorting problem with a simple ranking function:

Theorem 1. *The expected utility in Equation 6.3 is maximum if the entities are placed in the descending order of the value of the ranking function CE,*

$$CE(e_i) = \frac{U(e_i)C(e_i)}{C(e_i) + \gamma(e_i)} \tag{6.4}$$

Proof Sketch: The proof shows that any inversion in this order will reduce the expected profit. CE function is deduced from expected profits of two placements—the CE ranked placement and placement in which the order of two adjacent ads are inverted. We show that the expected profit from the inverted placement can be no greater that the CE ranked placement. Please refer to Appendix A-1 for the complete proof. □

As mentioned in the introduction, the ranking function CE is the utility generated per unit view probability consumed by the entity. With respect to browsing model in Figure 6.1, the top entities in the ranked list have higher view probabilities, and placing ads with greater utility per consumed view probability higher intuitively increases total utilities.

Note that the ordering above does not maximize the utility for selecting a subset of items. The seemingly intuitive method of ranking the set of items by CE and selecting top-k may not be optimal [51]. For optimal selection, the proposed ranking can be extended by a dynamic programming based

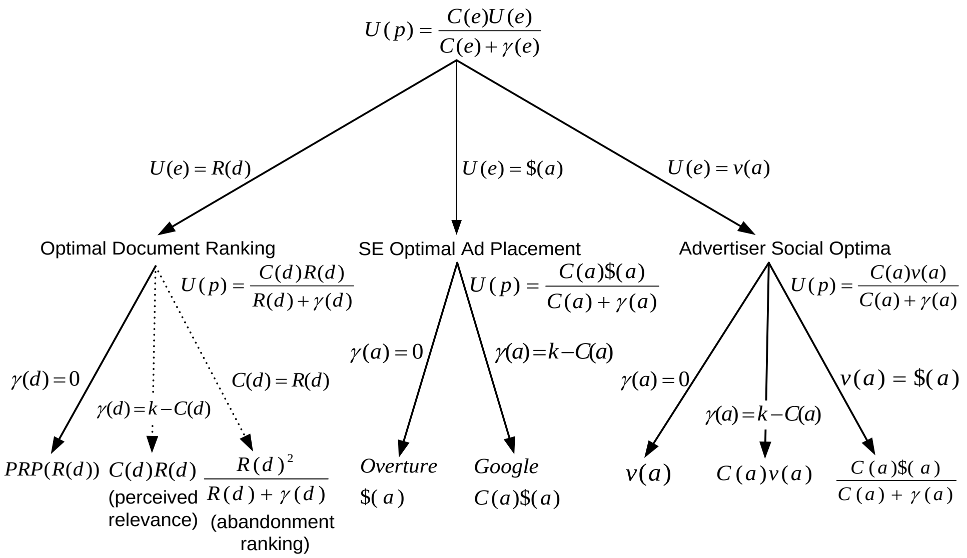


Figure 6.2: Taxonomy of reduced ranking functions of CE . The assumptions and corresponding reduced ranking functions are illustrated. The dotted lines denote predicted ranking functions incorporating new click model parameters.

selection—similar to the method suggested by Aggrawal *et al* [51] for maximizing advertiser's profit. In this dissertation, we discuss only the ranking problem.

6.3 Ranking Taxonomy

As we mentioned before, the CE ranking will can be made applicable to different ranking problems by plugging in the corresponding utilities. For example, if we plug in relevance as utility ($U(e)$ in Equation 6.4), the ranking function is for the documents, whereas if we plug in cost per click of ads, the ranking function is for ads. Further, we may assume specific constraints on one or more of the three parameters of CE ranking (e.g. $\forall_i \gamma(e_i) = 0$). Through these assumptions, CE ranking will suggest a number of reduced ranking functions with specific applications. These substitutions and reductions can be enumerated as a taxonomy of ranking functions.

We show the taxonomy in Figure 6.2. The three top branches of the taxonomy ($U(e) = R(d)$, $U(e) = \$(a)$, and $U(e) = v(a)$ branches) are for document ranking, ad ranking maximizing search engine profit, and ad ranking maximizing advertisers revenue respectively. These branches correspond to the substitution of utilities by document relevance, CPC, and private value of the advertisers. The sub-trees below these branches are the further reduced cases of these three main categories. The solid lines in Figure 6.2 denote the already known functions, while the dotted lines are the new ranking functions suggested by CE ranking. Sections 6.3.1,6.3.2, and 6.3.3 below discuss the further reductions of document ranking, search engine optimal ad ranking, and social optimal ad ranking respectively.

6.3.1 Document Ranking

For the document ranking the utility of ranking is the probability of relevance of the document. Hence by substituting the document relevance—denoted by $R(d)$—in Equation 6.4 we get

$$CE(d) = \frac{C(d)R(d)}{C(d) + \gamma(d)} \tag{6.5}$$

This function suggests the general optimal relevance ranking for the documents. We discuss some intuitively valid assumptions on user model for the document ranking and the corresponding ranking functions below. The three assumptions discussed below correspond to the three branches under Document Ranking subtree in Figure 6.2.

6.3.1.1 Sort by Relevance (PRP)

We elucidate two sets of assumptions under which the $CE(d)$ in Equation 6.5 will reduce to PRP.

First assume that the user has infinite patience, and never abandons results (i.e. $\gamma(d) \approx 0$). Substituting this assumption in Equation 6.5,

$$CE(d) \approx \frac{R(d)C(d)}{C(d)} = R(d) \tag{6.6}$$

which is exactly the ranking suggested by PRP.

In other words PRP is optimal for scenarios in which the user has infinite patience and never abandons checking the results (i.e. the user leaves browsing the results only by clicking a result).

The second set of slightly weaker assumptions under which the $CE(d)$ will reduce to PRP are:

1. $C(d) \approx R(d)$.

2. Abandonment probability $\gamma(d)$ is negatively proportional to the document relevance i.e. $\gamma(d) \approx k - R(d)$, where k is a constant between one and zero. This assumption corresponds to the intuition that the higher the perceived relevance of the current result, the less likely is the user abandoning the search.

Now $CE(d)$ reduces to,

$$CE(d) \approx \frac{R(d)^2}{k} \tag{6.7}$$

Since this function is strictly increasing with $R(d)$, ordering just by $R(d)$ results in the same ranking as suggested by the function. This implies that PRP is optimal under these assumptions also.

We should note that abandonment probability decreasing with perceived relevance is a more intuitively valid assumption than the infinite patience assumption above.

6.3.1.2 Ranking Considering Perceived Relevance

Recent click log studies have been used to effectively assess perceived relevance of document search snippets [19, 20]. Is it still an open question as ever as to how to use the perceived relevance for improved document ranking is unknown. We show that depending on the nature of abandonment probability $\gamma(d)$, the optimal ranking considering perceived relevance differs.

If we assume that $\gamma(d) \approx 0$ in Equation 6.5, the optimal perceived relevance ranking is the same as that suggested by PRP as we have seen in Equation 6.6.

On the other hand, if we assume that the abandonment probability is negatively proportional to the perceived relevance ($\gamma(d) = k - C(d)$) as above, the optimal ranking considering perceived relevance is

$$CE(d) \approx \frac{C(d)R(d)}{k} \propto C(d)R(d) \tag{6.8}$$

i.e. sorting in the order of the product of document relevance and perceived relevance is optimal under these assumptions. The assumption of abandonment probabilities negatively proportional to relevance is more realistic than infinite patience assumption as we discussed above. This discussion shows that by estimating nature of abandonment probability, one would be able to decide on the optimal perceived relevance ranking.

6.3.1.3 Ranking Considering Abandonment

We now consider the ranking considering abandonment probability $\gamma(d)$, with the assumption that the perceived relevance is approximately equal to the actual relevance. In this case $CE(d)$ becomes,

$$CE(d) \approx \frac{R(d)^2}{R(d) + \gamma(d)} \tag{6.9}$$

Clearly this is not a strictly increasing function with $R(d)$. So the ranking considering abandonment is different from PRP ranking, even if we assume that the perceived relevance is equal to the actual relevance. The abandonment ranking becomes same as PRP on the assumption that $\forall_d \gamma(d) = 0$, .

6.3.2 Optimal Ad Ranking for Search Engines

For the paid placement of ads, the utility of ads to the search engine are Cost Per Click (CPC) of ads. Hence, by substituting the CPC of the ad—denoted by $\$(a)$ —in Equation 6.4 we get

$$CE(a) = \frac{C(a)\$(a)}{C(a) + \gamma(a)} \tag{6.10}$$

Thus this function suggests the general optimal ranking for the ads. Please recall that the perceived relevance $C(a)$ is the same as the Click Through Rate (CTR) used for ad placement [16].

In the following subsections we demonstrate how the general ranking presented reduces to the currently used ad placement strategies under appropriate assumptions. We will show that they all correspond to the specific assumptions on the abandonment probability $\gamma(a)$. These two functions below correspond to the two branches under the SE Optimal ad placement subtree in Figure 6.2.

6.3.2.1 Ranking by Bid Amount

The sort by bid amount ranking was used by Overture Services (and was later used by Yahoo! for a while after their acquisition of Overture). Assuming that the user never abandons browsing (i.e. $\forall_a \gamma(a) = 0$), Equation 6.10 reduces to

$$CE(a) = \$(a) \tag{6.11}$$

This means that the ads are ranked purely in terms of their payment. To be precise, Overture ranks by bid amount, which is different from payment in a second price auction (since payment will be next higher bid amount). But both will result in the same ranking.

When $\gamma(a) = 0$, we essentially have a user with infinite patience who will keep browsing downwards until he finds the relevant ad. So, to maximize profit, it makes perfect sense to rank ads by bid amount. More generally, for small abandonment probabilities, ranking by bid amount is near optimal. Note that this ranking is isomorphic to PRP ranking discussed above for document ranking, since both rank based only on utilities.

6.3.2.2 Ranking by Expected Profit

Google and Microsoft are purported to be placing the ads in the order of expected profit based on product of CTR ($C(a)$ in CE) and bid amount ($\$(a)$) [52]. The ranking is part of the well known Generalized Second Price (GSP) auction mechanism. If we approximate abandonment probability as negatively proportional to the CTR of the ad (i.e. $\forall_a \gamma(a) = k - C(a)$) , the Equation 6.10 reduces to,

$$CE(a) \approx \frac{\$(a)R(a)}{k} \propto \$(a)R(a) \tag{6.12}$$

This shows that ranking ads by their standalone expected profit is near optimal as long as the abandonment probability is negatively proportional to the relevance. To be accurate, Google mechanism—GSP—uses bid amount of the advertisers (instead of CPC in Equation 6.12) for ranking. We will show that both will result in the same ranking by an order preserving property of the GSP pricing in Section 6.4. Note that this ranking is isomorphic to the perceived relevance ranking of the documents discussed above.

6.3.3 Revenue Optimal Ad Ranking

An important property of the auction mechanism is the expected revenue—which is the sum of the profits of the advertisers and the search engine. To analyze advertisers' profit, a private value model is commonly used. Each advertiser is assigned with a private value for the click equal to the expected revenue from the click. Advertisers pay a fraction of this revenue to the search engine depending on the pricing mechanism. The profit for advertisers is the difference between the private value and payment to the search engine. Profit for the search engine is the payment from the advertisers. Consequently, the revenue is the sum of the profits of all the parties—search engine and the advertisers.

The Advertiser Social Optima branch in Figure 6.2 corresponds to the ranking to maximize total revenue. Private value of advertisers a_i is denoted as—$v(a_i)$. By substituting the utility by private values in Equation 6.4 we get,

$$CE(d) = \frac{C(a)v(a)}{C(a) + \gamma(a)} \tag{6.13}$$

If the ads are ranked in this order, the ranking will guarantee maximum revenue.

In Figure 6.2 the two left branches of revenue maximizing subtree (labeled $\gamma(a) = 0$ and $\gamma(a) = k - C(a)$) correspond respectively to the assumption of no abandonment, and abandonment probabilities being negatively proportional to the click probability. These two cases are isomorphic to the Overture and Google ranking discussed in Section 6.3.2 above. We discuss further on revenue maximizing ranking in conjunction with a pricing mechanism in Section 6.4

The revenue optimal ranking is not directly implementable as search engines do not know the private value of the advertisers. But this ranking is useful in analysis of auctions mechanisms. Further, the search engine may try to effectuate this order through auction mechanism equilibriums as we demonstrate in Section 6.4.

6.4 Extending to Auctions

We have shown that CE ranking maximizes the profits for search engines for given CPCs in Section 6.3.2. In ad placement, the net profit of ranking to the search engine can only be analyzed in association with a pricing mechanism. To this end, we introduce a pricing to be used with the CE thus designing a full auction mechanism. Subsequently, we analyze the properties of the mechanism.

Let us start by describing the dynamics of ad auctions describe in Chapter 2 briefly. The search engine decides the ranking and pricing of the ads based on the bid amounts of the advertisers. Generally the pricing is not equal to the bid amount of the advertiser, but is instead derived based on the bids [32, 31, 53]. In response to these ranking and pricing strategies, the advertisers (more commonly, the software agents of the advertisers) may change their bids to maximize their profits. They may change bids hundreds of times a day. Eventually, the bids will stabilize at a fixed point where no advertiser can increase his profit by unilaterally changing his bid. This set of bids corresponds to a Nash Equilibrium of the auction mechanism. Hence the expected profits of a search engine will be the profits corresponding to the Nash Equilibrium.

The next section discusses the properties of any mechanism based on the user model in Figure 6.1—independent of the ranking and pricing strategies. In Section 6.4.2, we introduce a pricing mechanism and analyze its properties including the equilibrium.

6.4.1 Pricing Independent Properties

In this section we illustrate properties arising based on the user browsing model in Figure 6.1, not assuming any pricing or ranking strategy. One of the basic results is

Remark 1. *In any equilibrium the payment by the advertisers is less than or equal to their private values (i.e. individual rationality of the bidders is maintained).*

If this is not true, this advertiser may opt out from the auction by bidding zero and increase the profit, thereby violating the assumption of an equilibrium.

Remark 2. *In any equilibrium, the price paid by an advertiser increases monotonically as he moves up in the ranking unilaterally.*

From the browsing model, click probability of the advertisers is non-decreasing as he moves up in the position. Unless the price increases monotonically, advertiser can increase his profit by moving up, violating the assumption of an equilibrium.

Note that the proposed model is a general case of the positional auctions model by Varian [54]. Positional auctions assume static click probabilities for each position independent of the other ads. We assume realistic dynamic click probabilities depending on the ads above. Due to these externalities, the model is more complex and does not hold many of the properties derived by Varian [54] (e.g. monotonically increasing values and prices with positions).

Remark 3. *Irrespective of the ranking and pricing, the sum of revenues of the advertisers is upper bounded at*

$$E(V) = \sum_{i=1}^{N} v(a_i)C(a_i)\prod_{j=1}^{i-1}[1-(C(a_j)+\gamma(a_j))] \tag{6.14}$$

when the advertisers are ordered by $\frac{C(a)v(a)}{C(a)+\gamma(a)}$*. Further, this is an upper bound for the search engine profit.*

This result directly follows from the Advertisers Social Optima branch in Figure 6.2, and Equation 6.13.

The revenue is shared among the advertisers and search engine. For each click, dvertisers get a revenue equal to the private value $v(a)$ and pay a fraction equal to the CPC (set by the search engine pricing strategy) to the search engine. The total payoff for the search engine is the sum of the payments by the advertisers. Conversely, total payoff to the advertisers is the difference between the total revenue and payoff to the search engine. Since the suggested order above in Remark 3 maximizes revenue, which is the sum of the payoffs of all the players (search engine and the advertisers), this is a socially optimal order and the revenue realized is the socially optimal revenue.

A corollary of the social optimality combined with the individual rationality result in Remark 1 is that,

Remark 4. *The quantity $E(V)$ in Remark 3 is an upper bound for the search engine profit irrespective of the ranking and pricing mechanism.*

Social optimal revenue can be realized only if the ads are in the descending order of $\frac{C(a)v(a)}{C(a)+\gamma(a)}$. Social optimum is desirable for search engines, since this will increase the payoffs for advertisers for the same CPC. Increased payoffs will increase the advertiser's incentive to advertise with the search engine and will increase business for the search engine in the long term.

Since search engines do not know the private value of the advertisers (note that search engine perform the ranking), social optimal ranking based on private values is not directly feasible. We need to design a mechanism having an equilibrium coinciding with the social optimality. This will motivate advertisers towards bids coinciding with social optimal ordering. In addition to social optimality, it is highly desirable for the mechanism to be based on CE ranking to simultaneously maximize advertiser's revenue and search engine profit. In the following section we propose such a mechanism using CE ranking and prove the existence of an equilibrium in which the CE ranking coincides with the socially optimal allocation.

6.4.2 Pricing and Equilibrium

In this section, we define a pricing strategy to use with the CE ranking, and analyze the properties of the resulting mechanism.

For defining the pricing strategy, we define the pricing order as the decreasing order of $w(a)b(a)$, where $w(a)$ is,

$$w(a) = \frac{C(a)}{C(a) + \gamma(a)} \tag{6.15}$$

In this pricing order, we denote the i^{th} advertiser's $w(a_i)$ as w_i, $C(a_i)$ as c_i, $b(a_i)$ as b_i, and the abandonment probability $\gamma(a_i)$ as γ_i for convenience. Let $\mu_i = c_i + \gamma_i$. For each click, advertiser a_i is charged with a price p_i (CPC) equal to the minimum bid required to maintain its position in the pricing order,

$$p_i = \frac{w_{i+1} b_{i+1}}{w_i} = \frac{b_{i+1} c_{i+1} \mu_i}{\mu_{i+1} c_i} \tag{6.16}$$

Substituting p_i in Equation 6.10 for the ranking order, CE of the i^{th} advertiser is,

$$CE_i = \frac{p_i c_i}{\mu_i} \tag{6.17}$$

This proposed mechanism preserves the pricing order in the ranking order as well, i.e.

Theorem 2. *The order by $w_i b_i$ is the same as the order by CE_i for the auction i.e.*

$$w_i b_i \geq w_j b_j \iff CE_i \geq CE_j \tag{6.18}$$

Proof is given in the Appendix A-2. This order preservation property implies that the final ranking is the same as that based on bid amounts. i.e. ads can be ranked based on the bid mounts instead of CPCs. After the ranking, the CPCs can be decided based on this ranking order. A corollary of this order preservation is that the CPC is equal to the minimum amount the advertisers have to pay to maintain his position in the ranking order.

Further we show below that any advertisers' CPC is less than or equal to his bid.

Lemma 1 (Individual Rationality)**.** *The payment p_i of any advertiser is less or equal to his bid amount.*
Proof.

$$p_i = \frac{b_{i+1}c_{i+1}\mu_i}{\mu_{i+1}c_i} = \frac{b_{i+1}c_{i+1}}{\mu_{i+1}}\frac{\mu_i}{c_i b_i}b_i \leq b_i(\text{since } CE_i \geq CE_{i+1})$$

□

This means advertisers will never have to pay more than his bid, similar to GSP. This property makes it easy for the advertiser to decide his bid, as he may bid up to his click valuation. He will never have to pay more than his revenue irrespective of bids of other advertisers.

Interestingly, this mechanism also is a general case of the existing mechanisms, as in the case of CE ranking. In particular, the mechanism reduces to GSP (Google mechanism) and Overture mechanisms on the same assumptions on which CE ranking reduces to respective rankings (described in Section 6.3.2).

Lemma 2. *The mechanism reduces to Overture ranking with second price auction on the assumption $\forall_i \gamma_i = 0$*

Proof. This assumption implies

$$\begin{aligned} w_i &= 1 \\ &\Rightarrow p_i = b_{i+1} \text{ (second price auction)} \\ &\Rightarrow CE_i = b_{i+1} \equiv b_i \text{ (i.e. ranking by } b_{i+1} \text{ is equivalent to ranking by } b_i\text{)} \end{aligned}$$

□

Lemma 3. *The mechanism reduces to GSP on assumption* $\forall_i \gamma_i = k - c_i$

Proof. This assumption implies

$$\begin{aligned} w_i &= c_i \\ &\Rightarrow p_i = \frac{b_{i+1}c_{i+1}}{c_i} \text{ (GSP pricing)} \\ &\Rightarrow CE_i = \frac{b_{i+1}c_{i+1}}{k} \equiv \frac{b_i c_i}{k} \text{ (by Theorem 2)} \\ &\propto b_i c_i \end{aligned}$$

□

This in conjunction with Theorem 2 implies that GSP ranking by $c_i b_i$ (i.e. by bids) is the same as the ranking by $c_i p_i$ (by CPCs).

Now we will look at the equilibrium properties of the mechanism. We start by noticing that truth telling is not a dominant strategy. This trivially follows from the proof that GSP is a special case of the proposed mechanism. It is well known that for GSP truth telling is not a dominant strategy [31]. Hence we center our analysis on Nash Equilibrium conditions.

Theorem 3 (Nash Equilibrium)**.** *Without loss of generality assume that the advertisers are ordered in the decreasing order of* $\frac{c_i v_i}{\mu_i}$ *where* v_i *is the private* *value of the i^{th} advertiser. The advertisers are in a pure strategy Nash Equilibrium if*

$$b_i = \frac{\mu_i}{c_i}\left[v_i c_i + (1-\mu_i)\frac{b_{i+1}c_{i+1}}{\mu_{i+1}}\right] \tag{6.19}$$

This equilibrium is socially optimal as well as optimal for search engines for the given CPC's.

Proof Sketch: The inductive proof shows that for these bid values, no advertiser can increase his profit by moving up or down in the ranking. The full proof is given in Appendix A-3. □

We do not rule out the existence of multiple equilibria. The stated equilibrium is particularly interesting, due to the simultaneous social optimality and search engine optimality.

The following remarks show that the equilibria of other placement mechanisms are reduced cases of the proposed CE equilibrium, as a natural consequence of its generality. The stated equilibrium reduces to equilibriums in Overture mechanism and GSP under the same assumptions under which the ranking reduces to respective rankings.

Remark 5. *The bid values*

$$b_i = v_i c_i + (1-c_i)b_{i+1} \tag{6.20}$$

are a pure strategy Nash Equilibrium in Overture mechanism. This corresponds to the substitution of the assumption $\forall_i \gamma_i = 0$ (i.e. $\mu_i = c_i$) in Theorem 3.

The proof follows from Theorem 3 as both pricing and ranking is shown to be a special case of our proposed mechanism.

Similarly for GSP,

Remark 6. *The bid values*

$$b_i = v_i k + (1-k) b_{i+1} c_{i+1} \tag{6.21}$$

are a pure strategy Nash Equilibrium in GSP mechanism.

This equilibrium corresponds to the substitution of the assumption $\forall_i \gamma_i = k - c_i$ $(1 \geq k \geq 0)$ in Theorem 3. Since this is a special case, the proof for Theorem 3 is sufficient.

6.4.3 Comparison with VCG mechanism

We compare the revenue and equilibrium of CE mechanism with those of VCG [21, 22, 23]. VCG auctions combine an optimal allocation (ranking) with VCG pricing. VCG payment of a bidder is equal to the reduction of revenues of other bidders due to the presence of the bidder. A well known property is that VCG pricing with any socially optimal allocation has truth telling as the dominant strategy equilibrium.

In the context of online ads, a ranking optimal with respect to the bid amounts is socially optimal ranking for VCG. This optimal ranking is $\frac{b_i c_i}{\mu_i}$; as directly implied by the Equation 1 on substituting b_i for utilities. Hence this ranking combined with VCG pricing has truth telling as the dominant strategy equilibrium. Since $b_i = v_i$ at the dominant strategy equilibrium, ranking is socially optimal for an advertiser's true value as suggested in Equation 6.13.

The CE ranking function is different from VCG since CE ranking by payments optimizes search engine profits. On the other hand, VCG ranks by bids optimizing the advertiser's profit. But Theorem 2 shows that for the pricing used in CE, ordering of CE is the same as that of VCG. This order-preserving property facilitates the comparison of CE with VCG. The theorem below shows revenue dominance of CE over VCG for the same bid values of advertisers.

Theorem 4 (Search Engine Revenue Dominance). *For the same bid values for all the advertisers, the revenue of the search engine by CE mechanism is greater or equal to the revenue by VCG.*

Proof Sketch: The proof is an induction based on the fact that the ranking by CE and VCG are the same, as mentioned above. Full proof is given in Appendix A-4. □

This theorem shows that the CE mechanism is likely to provide higher revenue to the search engine even during transient times before the bids stabilize on equilibriums.

Based on Theorem 4 we prove revenue equivalence of the proposed CE equilibrium with dominant strategy equilibrium of VCG.

Theorem 5 (Equilibrium Revenue Equivalence). *At the equilibrium in Theorem 3, the revenue of search engine is equal to the revenue of the truthful dominant strategy equilibrium of VCG.*

Proof Sketch: The proof is an inductive extension of the Theorem 4. Please refer to Appendix A-5 for complete proof. □

Note that the CE equilibrium has lower bid values than VCG at the equilibrium, but provides the same profit to the search engine.

6.5 Considering Mutual Influences: Diversity Ranking

An assumption in CE ranking is that the entities are mutually independent as we pointed out in Section 6.1. In other words, the three parameters—$U(e)$, $C(e)$ and $\gamma(e)$—of an entity do not depend on other entities in the ranked list. In this section we relax this assumption and analyze the implications. Since the nature of the mutual influence may vary for different problems, we base our analysis on a specific well known problem—ranking considering diversity [24, 25, 26].

Diversity ranking accounts for the fact that the utility of an entity is reduced by the presence of a similar entity above in the ranked list. This is a typical example of the mutual influence between the entities. All the existing objective functions for the diversity ranking are known to be NP-Hard [24]. We analyze a most basic form of diversity ranking to explain why this is a fundamentally hard problem.

We modify the objective function in Equation 6.1 slightly to distinguish between the standalone utilities and the residual utilities—utility of an entity in the context of other entities in the list—as,

$$E(U) = \sum_{i=1}^{N} U_r(e_i) P_c(e_i) \tag{6.22}$$

where $U_r(e_i)$ denotes the residual utility.

We consider a simple case of diversity ranking problem by considering a set of entities—all having the same utilities, perceived relevances and abandonment probabilities. Some of these entities may be repeating. If an entity in the ranked list is the same as the entity in the list above, residual utility of that entity becomes zero. In this case, it is intuitive that the optimal ranking is to place the maximum number of pair-wise dissimilar entities in the top slots. The theorem below shows that even in this simple case the optimal ranking is NP-Hard.

Theorem 6. *Diversity ranking optimizing expected utility in Equation 6.22 is NP-Hard.*

Proof Sketch: The proof is by reduction from the independent set problem. See Appendix A-6 for the complete proof. □

Moreover, the proof by reduction from independent set problem has more severe implications than NP-Hardness as shown in the following corollary,

Corollary 1. *The constant approximation algorithm for ranking considering diversity is hard.*

Proof: The proof of NP-Hardness theorem above shows that the independent set problem is a special case of diversity ranking. This implies that a constant ratio approximation algorithm for the optimal diversity ranking would be a constant ratio approximation algorithm for the independent set problem. Since constant ratio approximation of the independent set is known to be hard (*cf.* Garey and Johnson [55] and Håstad [56]) the corollary follows. To define hard, in his landmark paper Håstad proved that independent set cannot be solved within $n^{1-\epsilon}$ for $\epsilon > 0$ unless all problems in NP are solvable in probabilistic polynomial time, which is widely believed to be not possible.[1]
□

This section shows that the optimal ranking considering mutual influences of parameters is hard. We leave formulating approximation algorithms (not necessarily constant ratio) for future research.

Beyond proving the intractability of mutual influence ranking, we believe that intractability of the simple scenario here explains why all diversity rankings are likely to be intractable. Further, the proof based on the reduction from the well explored independent set problem may help in adapting approximations algorithms from graph theory.

6.6 Simulation Experiments

The analysis in the previous sections shows that the existing ranking strategies are optimal only under more restrictive assumptions on parameters. This

[1] This belief is almost as strong as the belief $P \neq NP$

suggests that the expected relevances for documents (and profits for ads) can be improved by ranking using CE ranking. We perform a number of simulation experiments to quantify the potential increases in expected utilities by CE and its reduced forms.

In our first experiment in Figure 6.3a, we compare the CE ranking with rank by bid amount (Equation 6.11) strategy by Overture and rank by bid × perceived relevance (Equation 6.12) by Google. We assigned the perceived relevance values as a uniform random number between 0 and α $(0 \leq \alpha \leq 1)$ and abandonment probabilities as random between 0 and $1 - \alpha$ (this assures that $\forall_i \ (C(a_i) + \gamma(a_i)) \leq 1$). The bid amounts for ads are assigned uniformly random between 0 and 1. Note that uniform random is the maximum entropy distribution and makes least assumptions about the bid amounts. The number of relevant ads (corresponding to the number of bids on a query) is set to fifty. Simulated users are made to click on ads. The number of ads clicked is set as a random number generated in a zipf distribution with exponent 1.5. A power law is most intuitive for the distribution of the number of clicks.

Simulated users browse down the list. Users click an entity with probability equal to the perceived relevance and abandon search with a probability equal to the abandonment probability. The set of entities to be placed is created at random for each run. For the same set of entities, three runs—one with each ranking strategy—are performed. Simulation is repeated 2×10^5 times for each value of alpha.

In Figure 6.3a CE ranking is optimal for all values of α as expected. Confirming with the discussions in Subsection 6.3.2 above, as the abandonment probability becomes smaller ranking by bid strategy gives better profits and reaches optimal at $\gamma(a) = 0$ (i.e $\alpha = 0$). The expected profit by CE exceeds that by the competing strategy by 40-80% for some values of α. For example, at $\alpha = 0.3$ $bid \times percieved$ (competing strategy) gives an expected profit of \$0.34 while CE gives a profit of \$0.63 (exceeds by 84.4%) and for $\alpha = 0.5$ $bid \times perceived$ gives a profit of \$0.69, as against \$0.97 by CE (exceeds by 40.6%). Further, perceived relevance ranking dominates rank by bid strategy for most values of α.

Another way of interpreting Figure 6.3a is as the comparison of ranking by CE, PRP and perceived relevance ranking (Equation 6.8). As we discussed, PRP and perceived relevance rankings exactly corresponds to ad rankings by bid and bid × perceived relevance respectively, with utility being relevance instead of bid amounts. The simulation graphs will look exactly the same.

In Figure 6.3b we compare CE, PRP and abandonment ranking (Equation 6.9) under the same settings used for Figure 6.3a. CE provides the maximum utility as expected, and abandonment ranking comes in second place. Abandonment ranking provides sub-optimal utility—since the condition $\forall_d R(d) = C(d)$ is not satisfied—but dominates over PRP. Also as abandonment probability becomes zero (i.e $\alpha = 1$) abandonment rankings becomes same as PRP and optimal as we discussed in Subsection 6.3.1.

Figure 6.4a compares the perceived relevance ranking (Equation 6.8), CE, and PRP under the condition for optimality for perceived relevance ranking ($\forall_d \gamma(d) = k - R(d)$). For this, we set $\gamma(d) = \alpha - C(d)$ keeping all other settings same as the previous experiments. The Figure 6.4a shows that the perceived relevance ranking provides optimal utility, exactly overlapping with CE curve as expected. Further, note that utilities by PRP are very low under this condition. The utilities by PRP in fact goes down after $\alpha = 0.2$. The increase in abandonment probability, as well as increased sub-optimality of

PRP for higher abandonment (since PRP does not consider abandonment) probabilities may be causing this reduction.

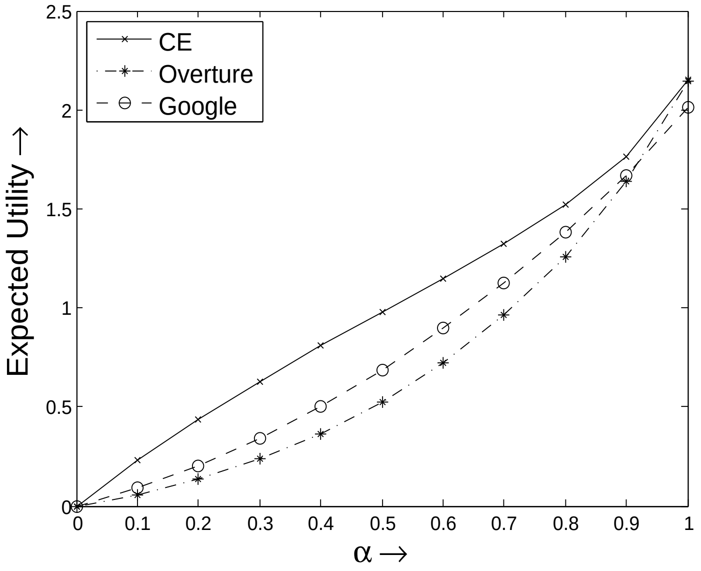
In our next experiment shown in Figure 6.4b, we compare abandonment ranking (Equation 6.9) with PRP and CE under the condition $\forall_d C(d) = R(d)$ (i.e. optimality condition for abandonment ranking). All other settings are the same as those for the experiments in Figure 6.3a and 6.3b. Here we observe that the abandonment ranking is optimal and exactly overlaps with CE as expected. PRP is sub-optimal but closer to optimal than random $C(d)$ used for experiments in Figure 6.3b. The reason may be that $C(d) = R(d)$ is one of the two conditions required for PRP to be optimal for both sets of assumptions we discussed in Subsection 6.3.1. When abandonment probability becomes zero PRP relevance reaches optimum as we have already seen.

Simulation experiments exactly confirm to the predictions by the theoretical analysis above. Although the simulation is no substitute for experiments on real data, we expect that observed significant improvements in expected utilities would motivate future research to evaluate these rankings on click logs.

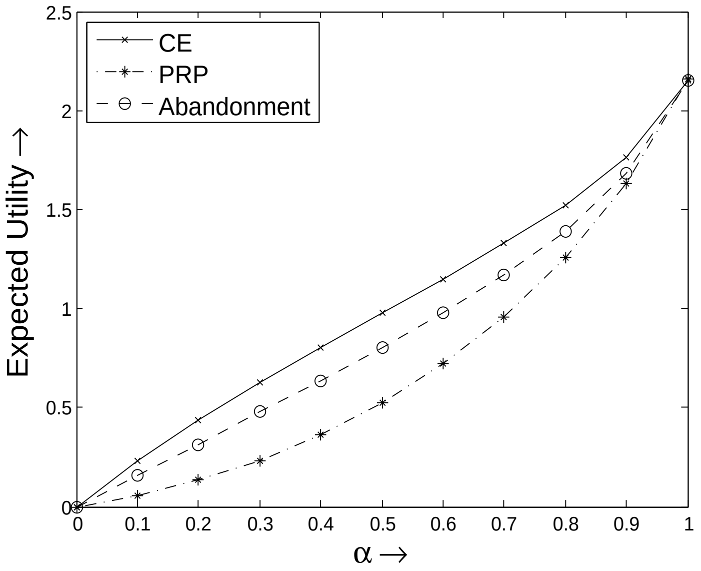
6.7 Chapter Summery

We approach the web ranking as a utility maximization based on user's click model, and derive the optimal ranking—namely CE ranking. The ranking is simple and intuitive; and optimal considering perceived relevance and abandonment probability of user behavior. For specific assumptions on parameters the ranking function reduces to a taxonomy of ranking functions in multiple ranking domains. The enumerated taxonomy will help to decide optimal ranking for a specific user behavior. In addition, the taxonomy shows that the existing document and ad ranking strategies are special cases of the proposed ranking function under specific assumptions.

To apply CE ranking to ad auctions, we incorporate a second price based pricing. The resulting CE mechanism has a Nash Equilibrium in pure strategies which simultaneously optimizes search engine and advertiser revenues. CE mechanism is revenue dominant over VCG for the same bid vectors, and has an equilibrium which is revenue equivalent with the truthful equilibrium of VCG. Finally, we relax the assumption of independence between entities in CE ranking and consider diversity ranking. The ensuing analysis revels that diversity ranking is an inherently hard problem; since even the basic formulations are NP-Hard with unlikely constant ratio approximation algorithms. Our simulation analysis suggests significant improvement in profits by CE ranking over existing ranking strategies.

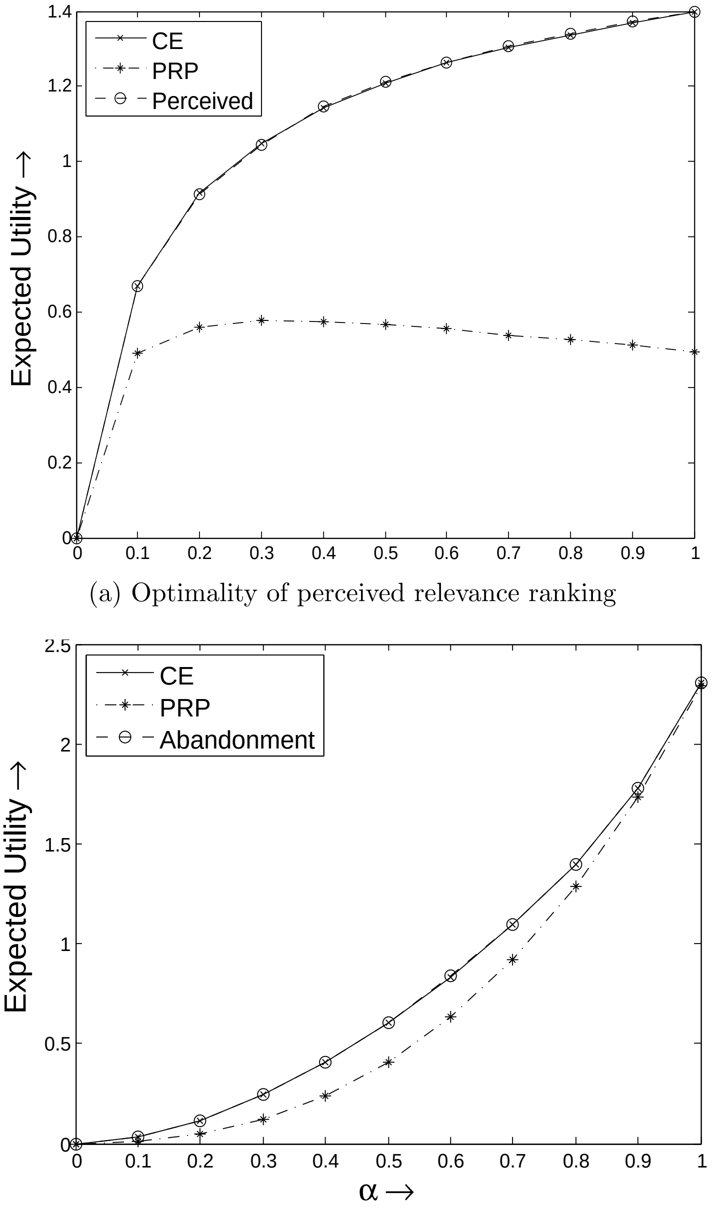


(a) Google and Overture comparison



(b) PRP and abandonment ranking comparison

Figure 6.3: (a) Comparison of Overture, Google and CE rankings. Perceived relevances are uniformly random in $[0, \alpha]$ and abandonment probabilities are uniformly random in $[0, 1 - \alpha]$. CE provides optimal expected profits for all values of α. (b) Comparison of CE, PRP and abandonment ranking (Equation 6.9). Abandonment ranking dominates PRP.



(a) Optimality of perceived relevance ranking

(b) Optimality of abandonment ranking

Figure 6.4: Optimality of reduced forms under specific assumptions (a) fixing $\gamma(d) = k - R(d)$. Perceived relevance ranking is optimal for all values of α. (b) fixing $C(d) = R(d)$. Abandonment ranking is optimal.

Chapter 7
Related Work

There has been a large volume of research in ranking of organic and sponsored results. Even before the prevalence of the World Wide Web, there is early research in library information retrieval and traditional auctions. Among the volume or related research, we describe selected closely related research in deep web and online ad ranking in sections below.

7.1 Deep web Ranking

The related research in deep web may be segregated into three areas:

1. Source selection in deep web integration and in other data integration problems.

2. Trust analysis for open collections including the surface and the deep web.

3. Related problems in searching including result ranking, sampling, schema mapping etc.

We discuss the past research in these three categories in three sections below.

7.1.1 Source Selection

The indispensability and difficulty of source selection for the deep web has been recognized previously [57]. Current relational database selection methods minimize the cost by retrieving maximum number of distinct records from minimum number of sources [4]. Cost based web database selection is formulated as selecting the least number of databases maximizing number of relevant tuples (coverage). The related problem of collecting source statistics [4, 8] has also been researched. These papers do not address the ranking problem but related problems in deep web integration.

Considering research in the text databases selection, Callan *et al.* [6] formulated a method called CORI for query specific selection based on relevance. Cooperative and non-cooperative text database sampling [37, 8] and selection considering coverage and overlap to minimize the cost [45, 7] are addressed by a number of researchers. As we mentioned in the introduction, none of these relational or text databases selection methods consider trust and importance of the databases, which is the main focus of or research.

Centralized warehousing approaches have been tried for integrating parts of the deep web. Google Product Search [28] works on Google Base—an open repository for products—contains data from large number of web databases. In a different surfacing approach of extending the search to web databases, Google crawls and index parts of the data in popular sources as html pages, disregarding the structure [27]. Neither of these papers focuses on ranking problem.

7.1.2 Trust Analysis

A probabilistic framework for trust assessment based on agreement of web pages for question answering has been presented by Yin *et al.* [58], and Yin and Tan [59]. Galland *et. al.* [60] 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 have limiting its usability in the deep web.

Dong *et al.* [61, 42] extend this basic idea of Yin *et al.* [58], and extend the work by computing source dependence and using a different accuracy model.

In this work source copying is detected based on completeness, accuracy and formatting [42]. But deep web collusion is more than having same data (hence data copying), since collusion manifests in data and ranking as discussed in Section 3.6. Further, limited access based on keyword search makes it hard to retrieve the entire data, making extensions of methods by Dong *et al.* to deep web collusion detection hard. 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 answer sets. Our collusion detection approach accounts for these additional difficulties.

Clustered analyzing of trust for multi-group environments has been attempted by Gupta *et al.* [62]. Gupta and Han [63] give a comprehensive survey of network based trust analysis which includes detailed discussions of SourceRank [64, 65].

7.1.3 Search and Result Ranking

The problem of ranking database tuples for keyword search in databases has been addressed [9, 66]. The focus of these papers are on relevance assessment of tuples for keyword search in a single database, and 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.* [67]. Their paper considers the relevance assessment for search in a single database, and does not consider the trust problem. Further, the paper assumes 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 [68]. Lee [69] 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 3.1 in giving a basis for agreement-based relevance assessment. For the surface web, Gyöngyi *et al.* [70] proposed trust rank, an extension of page rank considering trustworthiness of hyperlinked pages. Kurland and Lee [71] proposed a re-ranking approach based on centrality on a graph induced by language models. Agreement on hidden variables between several learners has been used to achieve tractable learning time for joint learning [72].

Many of the related problems in deep web integration and search have been addressed. Number of methods are used for schema mapping of form interfaces of different web databases [73, 74, 75]. The sampling problem of web databases was explored [76, 77]. Number of methods has been tried for record linkage [36, 78]. Completion and expansion of autonomous web database records at query time was attempted by a few papers [79, 80].

7.2 Ad Ranking

The related research falls into the categories of:

1. The user click models.
2. Document ranking based on a utility maximization approach, and diversity ranking.
3. Recent work on optimizing ad auctions based on click models.

We discuss the research in these three areas in the sections below.

7.2.1 Click Models

User behavior studies in click models validate the ranking function introduced. There are a number of position based and cascade models studied recently [81, 30, 17, 82, 18]. In particular, General Click Model (GCM) by Zhu *et al.*[18] is closely related to the model we used as we mentioned above. Zhu *et al.* [18]

have listed assumptions under which the GCM would reduce to other click models. Optimizing utilities of two dimensional placement of search results has been studied by Chierichetti *et al.* [83]. These models empirically validate the correctness of the click model used in this dissertation.

Along with the current click models, there has been research on evaluating perceived relevance of the search snippets [19] and ad impressions [20]. Research in this direction neatly complements our new ranking function by estimating the parameters required.

7.2.2 Ranking

The existing document ranking based on PRP [14] claims that a retrieval order sorted on relevance leads to the largest number of relevant documents in a result set than any other policy. Gordon and Lenk [15, 84] identified the required assumptions for the optimality of the ranking according to PRP. Our discussion on PRP may be considered as an independent formulation of assumptions under which PRP is optimal for web ranking.

Diversity ranking has received considerable attention recently [25, 26]. The objective functions used to measure diversity by prior works are known to be NP-Hard [24].

7.2.3 Ad Auctions

The impact of click models on ranking has been analyzed in ad-placement. In our workshop paper [85] we proposed the optimal ad ranking considering mutual influences. The ranking uses the same user model, but the paper considers only ad ranking, and does not include generalizations and auctions. Later Aggarwal *et al.* [51] as well as Kempe and Mahdian [86] analyzed placement of ads using a similar Markovian click model. The click model used is less detailed than our model since abandonment is not modeled separately from click probability. These two papers optimize the sum of the revenues of the advertisers. We optimize search engine profits. Nevertheless, the ranking formulation has common components with these two papers, as our previous paper [85] as these three papers formulated ranking based on the similar browsing models independently at almost the same time frame. But, unlike this dissertation, any of the above papers do not have a pricing, auctions, and a generalized taxonomy.

Giotis and Karlin [87] extend markovian model ranking by applying GSP pricing and analyzing the equilibrium. The GSP pricing and ranking lacks the optimality and generality properties we prove in the dissertation. Deng and Yu [88] extend Markovian models by suggesting a ranking and pricing schema for the search engines and prove the existence of a Nash Equilibrium. The ranking is a simpler bid based ranking (not based on CPC as in our case); and mechanism as well as equilibrium do not show optimality properties. This dissertation is different from both the above works by using a more detailed model, by having optimality properties, detailed comparisons with other baseline mechanisms, and in the ability to generalize to a family of rankings.

Kuminov and Tennenholtz [89] proposed a Pay Per Action (PPA) model similar to the click models and compared the equilibrium of GSP mechanism on the model with the VCG. Ad auctions considering influence of the other ads on conversion rates are analyzed by Ghosh and Sayedi [90]. Both these papers address different problems than considered in this dissertation.

The proposed model is a general case of the positional auctions model by Varian [54]. Positional auctions assume static click probabilities for each position independent of other ads. We assume more realistic dynamic click probabilities depending on the ads above. Since we consider these externalities, our model, auction, and analysis are much more complex.

To the best of my knowledge, there is no other work addressing the problems in the deep web ranking and ad auctions addressed in this dissertation.

Chapter 8

Conclusions and Future Work

Improved ranking algorithms are crucial for the accessibility as well as the profitability of the search engines. This dissertation considers the ranking of organic and sponsored results in web search. We present significant advancements in both deep web integration and ad auctions. Considering the importance and dynamism of these emerging areas, there are many related open problems. We discuss conclusions of the dissertation and promising future research directions below.

8.1 Conclusions

We describe the conclusions in deep web integration and ad auctions in the following two sections.

8.1.1 Deep web Integration

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 measure is insensitive to the popularity and trustworthiness of the results. These 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 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. Our comprehensive empirical evaluation shows that SourceRank improves the relevance of the sources selected compared to existing methods and effectively removes corrupted sources. We also demonstrated that combining SourceRank with Google Product search ranking significantly improves the quality of the results. Further our evaluations show that the proposed result ranking effectively improve precision and eliminate 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 demonstrate the added precision by Topical-SourceRank for multi-domain search. We implement the proposed source and result ranking in our Factal deep web search engine prototype Factal (http://factal.eas.asu.edu).

8.1.2 Ad Ranking

The added dimension of profit in addition to relevance incurs interesting problems in ranking sponsored ads. We present a unified approach to ranking of documents and ads as a utility maximization based on user's click model. We derive the ranking function—namely CE ranking—and prove the optimality with respect to the user click model. The ranking is simple and intuitive; and optimal considering perceived relevance and abandonment probability of click models.

For specific assumptions on parameters, the ranking reduces to a taxonomy of ranking functions in multiple ranking domains. The enumerated taxonomy will help to decide optimal ranking for a specific user behavior. In addition, the taxonomy shows that the existing document and ad ranking strategies are special cases of the proposed ranking function under specific assumptions.

To apply CE ranking to ad auctions, we incorporate a second price based pricing. The resulting CE mechanism has a Nash Equilibrium which simultaneously optimizes search engine and advertiser revenues. CE mechanism is revenue dominant over VCG for the same bid vectors, and has an equilibrium which is revenue equivalent with the truthful equilibrium of VCG.

Finally, we relax the assumption of independence between entities in CE ranking and consider diversity ranking. The ensuing analysis revels that diversity ranking is an inherently hard problem; since even the basic formulations are NP-Hard with unlikely constant ratio approximation algorithms.

In addition to proving optimality of the proposed ranking, we perform number of simulation experiments to approximately quantify the improvement in profits. The analysis suggests significant improvement in profits by CE ranking over the existing ranking strategies, and validates the predictions of our earlier theoretical analysis.

8.2 Future Work

The problems in deep web search are at least as large 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. On the other hand need to analyze 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 unique databases, the performance and improvement of these methods may be further explored.

For topic specific sources 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 [91, 92, 93]. Topical sampling queries may be extracted automatically from the databases belonging to a topic after the classification [27].

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.

Another open challenge in ranking results is to decide the significance of source reputation in ranking results, as we pointed out in Section 2.4. A possible approach is to assess the variance in intra-source result quality and change the weightage of linage accordingly. Further, deciding on the tradeoff between the diversity and uniformity in the results is hard. For this, the degree of agreement between the sources may be used as an indication of the appropriate degree of diversity. For example, if the sources provide multiple distinct clusters of results for a query, including a few results from each cluster is likely to satisfy maximum number of users.

Deep web integration systems has to generate wrappers, automatically or semi-automatically [29]. 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 empirically.

Regarding future research in ad ranking, assessing profits by CE mechanism on a large scale search engine click log will quantify improvement in a real data. Learning as well as prediction of abandonment probability from click logs as well as by parametric learning are interesting problems.

The suggested ranking is optimal for other web ranking scenarios with similar click models—like products and friends recommendations—and may be extended to these problems. Further, effective approximation schemes for diversity ranking based on similarity with the independent set problem may be investigated.

Another interesting extension is considering mutual influence between the users—in addition to the mutual influence of ads—in an online social networking context. In social ads, the clicks or shares by a user may influence expected click rates of his friends. This mutual influence may necessitate substantial changes in optimal ranking and pricing strategies.

REFERENCES

[1] A. Wright, “Searching the deep web”, *Commmunications of ACM*, 2008.

[2] J. Madhavan, S. Jeffery, S. Cohen, X. Dong, D. Ko, C. Yu, and A. Halevy, “Web-scale Data Integration: You can only afford to Pay As You Go”, *Proceedings of CIDR*, 2007.

[3] N Fuhr, “A Decision-Theoretic Approach to Database Selection in Networked IR”, *ACM Transactions on Information Systems*, vol. 17, no. 3, pp. 229–249, 1999.

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

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

[6] J.P. Callan, Z. Lu, and W.B. Croft, “Searching distributed collections with inference networks”, in *Proceedings of ACM SIGIR*. ACM, NY, USA, 1995, pp. 21–28.

[7] M. Shokouhi and J. Zobel, “Federated text retrieval from uncooperative overlapped collections”, in *Proceedings of the ACM SIGIR*. ACM, 2007.

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

[9] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan, “Keyword searching and browsing in databases using BANKS”, in *ICDE*, 2002, p. 0431.

[10] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine", *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[11] John Kleinberg, "Authoritative Sources in a Hyperlinked Environment", *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[12] W.W. Cohen, "Integration of heterogeneous databases without common domains using queries based on textual similarity", *ACM SIGMOD Record*, vol. 27, no. 2, pp. 201–212, 1998.

[13] T.H. Haveliwala, "Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search", *IEEE transactions on knowledge and data engineering*, pp. 784–796, 2003.

[14] S E Robertson, "The probability ranking principle in ir", in *Journal of Documentation*, 1977, vol. 33, pp. 294–304.

[15] Michael G Gordon and Peter Lenk, "A utility theory examination of probability ranking principle in information retrieval", in *Journal of American Society of Information Science*, 1991, vol. 41, pp. 703–714.

[16] Mathtew Richardson, Ewa Dominowska, and Robert Ragno, "Predicting clicks: Estimating the click-through rate for new ads", in *Proceedings of WWW*, May 2007.

[17] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.M. Wang, and C. Faloutsos, "Click chain model in web search", in *Proceedings of WWW*. ACM New York, NY, USA, 2009, pp. 11–20.

[18] Z.A. Zhu, W. Chen, T. Minka, C. Zhu, and Z. Chen, "A novel click model and its applications to online advertising", in *In Proceedings of Web search and data mining*. ACM, 2010, pp. 321–330.

[19] Y. Yue, R. Patel, and H. Roehrig, "Beyond Position Bias: Examining Result Attractiveness as a Source of Presentation Bias in Clickthrough Data", in *Proceedings of WWW*, 2010.

[20] C L A Clarke, Eugene Agichtein, Susan Dumais, and Ryen W White, "The influence of caption features on clickthrough patterns in web search", in *Proceedings of SIGIR*. ACM, July 2007, pp. 135–142.

[21] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders", *The Journal of finance*, vol. 16, no. 1, pp. 8–37, 1961.

[22] E.H. Clarke, "Multipart pricing of public goods", *Public choice*, vol. 11, no. 1, pp. 17–33, 1971.

[23] T. Groves, "Incentives in teams", *Econometrica: Journal of the Econometric Society*, pp. 617–631, 1973.

[24] B. Carterette, "An analysis of NP-completeness in novelty and diversity ranking", *Advances in Information Retrieval Theory*, pp. 200–211, 2010.

[25] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results", in *Proceedings of the Second ACM International Conference on Web Search and Data Mining*. ACM, 2009, pp. 5–14.

[26] D. Rafiei, K. Bharat, and A. Shukla, "Diversifying Web Search Results", in *Proceedings of WWW*, 2010.

[27] J. Madhavan, D. Ko, Ł. Kot, V. Ganapathy, A. Rasmussen, and A. Halevy, "Google's deep web crawl", *Proceedings of Very Large Databases Endowment*, vol. 1, no. 2, pp. 1241–1252, 2008.

[28] "Google Products", http://www.google.com/products, 2011.

[29] A. Arasu and H. Garcia-Molina, "Extracting structured data from Web pages", in *Proceedings of SIGMOD*. ACM Press New York, NY, USA, 2003, pp. 337–348.

[30] Nick Craswell, Onno Zoeter, Michael Tayler, and Bill Ramsey, "An experimental comparison of click position bias models", in *Proceedings of WSDM*, February 2008, pp. 87–94.

[31] B. Edelman, M. Ostrovsky, and M. Schwarz, "Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords", 2005.

[32] D. Easley and J. Kleinberg, *Networks, crowds, and markets: Reasoning about a highly connected world*, Cambridge Univ Pr, 2010.

[33] D.F. Gleich, P.G. Constantine, A.D. Flaxman, and A. Gunawardana, "Tracking the random surfer: empirically measured teleportation parameters in PageRank", in *Proceedings of World Wide Web*, 2010.

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

[35] J. Wang, J.R. Wen, F. Lochovsky, and W.Y. Ma, "Instance-based schema matching for web databases by domain-specific query probing", in *In Proceedings of Very Large Databases*. VLDB Endowment, 2004, pp. 408–419.

[36] N. Koudas, S. Sarawagi, and D. Srivastava, "Record linkage: similarity measures and algorithms", in *Proceedings of SIGMOD*. ACM, 2006, p. 803.

[37] J. Callan and M. Connell, "Query-based sampling of text databases", *ACM TOIS*, vol. 19, no. 2, pp. 97–130, 2001.

[38] "New york times books best sellers", http://www.hawes.com/number1s.htm, 2010.

[39] "New york times guide to best 1000 movies", http://www.nytimes.com/ref/movies/1000best.html, 2010.

[40] J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo, "Extracting Semistructured Information from the Web", in *Proceedings of the Workshop on Management of Semistructured Data*. Tucson, Arizona: ACM, 1997, pp. 18–25.

[41] Y. Zhai and B. Liu, "Web data extraction based on partial tree alignment", in *Proceedings of World Wide Web*. ACM, 2005, pp. 76–85.

[42] X.L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava, "Global detection of complex copying relationships between sources", *Proceedings of the Very Large Databases Endowment*, vol. 3, no. 1, 2010.

[43] "IMDB movie database", http://www.imdb.com, 2011.

[44] "UIUC TEL-8 repository", http://metaquerier.cs.uiuc.edu/repository/datasets/tel-8/index.html, 2003.

[45] L. Si and J. Callan, "Relevant document distribution estimation method for resource selection", in *Proceedings of ACM SIGIR*. ACM New York, NY, USA, 2003, pp. 298–305.

[46] "Open directory project movies", http://www.dmoz.org/Arts/Movies/Titles/, 2011.

[47] "Pbase camera list", http://www.pbase.com/cameras, 2011.

[48] "Best selling albums worldwide", http://en.wikipedia.org/wiki/List_of_best-selling_albums_worldwide, 2011.

[49] M. Richardson, E. Dominowska, and R. Ragno, "Predicting clicks: estimating the click-through rate for new ads", in *Proceedings of World Wide Web*. ACM, 2007, pp. 521–530.

[50] R. Balakrishnan and S. Kambhampati, "Factal: Integrating Deep Web Based on Trust and Relevance", in *Proceedings of World Wide Web*. ACM, 2011.

[51] G. Aggarwal, J. Feldman, S. Muthukrishnan, and M. Pál, "Sponsored search auctions with markovian users", *Internet and Network Economics*, pp. 621–628, 2008.

[52] Matthew Richardson, Amit Prakash, and Eric Brill., "Beyond pagerank: Machine learning for static ranking", in *WWW Proceedings*. ACM, May 2006, pp. 707–714.

[53] G. Aggarwal, A. Goel, and R. Motwani, "Truthful auctions for pricing search keywords", in *Proceedings of the 7th ACM conference on Electronic commerce*. ACM, 2006, pp. 1–7.

[54] H.R. Varian, "Position auctions", *International Journal of Industrial Organization*, vol. 25, no. 6, pp. 1163–1178, 2007.

[55] Michael R Garey and David R Johnson, "The complexity of near-optimal graph coloring", in *Journal of ACM*, 1976.

[56] J. Håstad, "Clique is hard to approximate within n", in *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, 1996, pp. 627–636.

[57] J. Madhavan, A. Halevy, S. Cohen, X.L. Dong, S.R. Jeffery, D. Ko, and C. Yu, "Structured Data Meets the Web: A Few Observations", *Data Engineering*, vol. 31, no. 4, 2006.

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

[59] X. Yin and W. Tan, "Semi-supervised truth discovery", in *Proceedings of World Wide Web*. ACM, 2011, pp. 217–226.

[60] Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart, "Corroborating information from disagreeing views", in *Proceedings of Web search and data mining*, 2010, WSDM '10, pp. 131–140.

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

[62] M. Gupta, Y. Sun, and J. Han, "Trust analysis with clustering", in *Proceedings of World Wide Web*. ACM, 2011, pp. 53–54.

[63] M. Gupta and J. Han, "Heterogeneous network-based trust analysis: A survey", 2011.

[64] R. Balakrishnan and S. Kambhampati, "Sourcerank: Relevance and trust assessment for deep web sources based on inter-source agreement", in *Proceedings of World Wide Web*. ACM, 2011, pp. 227–236.

[65] R. Balakrishnan and S. Kambhampati, "SourceRank: relevance and trust assessment for deep web sources based on inter-source agreement", in *Proceedings of World Wide Web*. ACM, 2010, pp. 1055–1056.

[66] S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum, "Probabilistic ranking of database query results", in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 888–899.

[67] S. Agrawal, K. Chakrabarti, S. Chaudhuri, V. Ganti, A.C. Konig, and D. Xin, "Exploiting web search engines to search structured databases", in *Proceedings of World Wide Web*. ACM, 2009, pp. 501–510.

[68] W.B. Croft, “Combining approaches to information retrieval”, *Advances in information retrieval*, vol. 7, pp. 1–36, 2000.

[69] J.H. Lee, “Analyses of multiple evidence combination”, in *ACM SIGIR Forum*. ACM, 1997, vol. 31, p. 276.

[70] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen, “Combating web spam with trustrank”, in *Proceedings of Very Large Databases*, 2004.

[71] O. Kurland and L. Lee, “Pagerank without hyperlinks: structural reranking using links induced by language models”, in *Proceedings of ACM SIGIR*. ACM, 2005, pp. 306–313.

[72] P. Liang, D. Klein, and M.I. Jordan, “Agreement-based learning”, *Advances in Neural Information Processing Systems*, vol. 20, pp. 913–920, 2008.

[73] J. Madhavan, PA Bernstein, A. Doan, and A. Halevy, “Corpus-based schema matching”, in *Data Engineering, 2005. ICDE 2005. Proceedings.*, 2005, pp. 57–68.

[74] J. Wang, J.R. Wen, F. Lochovsky, and W.Y. Ma, “Instance-based schema matching for web databases by domain-specific query probing”, in *Proceedings of Very Large Databases*. VLDB Endowment, 2004, pp. 408–419.

[75] B. He and K.C.C. Chang, “Statistical schema matching across web query interfaces”, in *Proceedings of SIGMOD*. ACM, 2003, pp. 217–228.

[76] A. Dasgupta, G. Das, and H. Mannila, “A random walk approach to sampling hidden databases”, in *Proceedings of SIGMOD*. ACM Press New York, NY, USA, 2007, pp. 629–640.

[77] J. Wang and F.H. Lochovsky, “Data extraction and label assignment for web databases”, in *Proceedings of World Wide Web*. ACM, 2003, pp. 187–196.

[78] I.P. Fellegi and A.B. Sunter, “A theory for record linkage”, *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, 1969.

[79] R. Gummadi, A. Khulbe, A. Kalavagattu, S. Salvi, and S. Kambhampati, “Smartint: using mined attribute dependencies to integrate fragmented

web databases", in *Proceedings of World Wide Web*. ACM, 2011, pp. 51–52.

[80] G. Wolf, A. Kalavagattu, H. Khatri, R. Balakrishnan, B. Chokshi, J. Fan, Y. Chen, and S. Kambhampati, "Query processing over incomplete autonomous databases: query rewriting using learned data dependencies", *The Very Large Databases Journal*, vol. 18, no. 5, pp. 1167–1190, 2009.

[81] G.E. Dupret and B. Piwowarski, "A user browsing model to predict search engine click data from past observations.", in *Proceedings of SIGIR*. ACM, 2008, pp. 331–338.

[82] O. Chapelle and Y. Zhang, "A dynamic bayesian network click model for web search ranking", in *Proceedings of WWW*. ACM, 2009, pp. 1–10.

[83] F. Chierichetti, R. Kumar, and P. Raghavan, "Optimizing two-dimensional search results presentation", in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 257–266.

[84] Michael G Gordon and Peter Lenk, "When is probability ranking principle suboptimal?", in *Journal of American Society of Information Science*, 1992, vol. 42.

[85] R. Balakrishnan and S. Kambhampati, "Optimal ad ranking for profit maximization", in *Proceedings of the 11th International Workshop on the Web and Databases*, 2008.

[86] D. Kempe and M. Mahdian, "A cascade model for externalities in sponsored search", *Internet and Network Economics*, pp. 585–596, 2008.

[87] I. Giotis and A. Karlin, "On the equilibria and efficiency of the GSP mechanism in keyword auctions with externalities", *Internet and Network Economics*, pp. 629–638, 2008.

[88] X. Deng and J. Yu, "A New Ranking Scheme of the GSP Mechanism with Markovian Users", *Internet and Network Economics*, pp. 583–590, 2009.

[89] D. Kuminov and M. Tennenholtz, "User modeling in position auctions: re-considering the gsp and vcg mechanisms", in *Proceedings of The 8th In-*

ternational Conference on Autonomous Agents and Multiagent Systems-Volume 1, 2009, pp. 273–280.

[90] A. Ghosh and A. Sayedi, “Expressive auctions for externalities in online advertising”, in *Proceedings of the 19th international conference on World wide web.* ACM, 2010, pp. 371–380.

[91] L. Gravano, P.G. Ipeirotis, and M. Sahami, “QProber: A system for automatic classification of hidden-Web databases”, *ACM Transactions on Information Systems*, vol. 21, no. 1, pp. 1–41, 2003.

[92] B. He, T. Tao, and K.C.C. Chang, “Organizing structured web sources by query schemas: a clustering approach”, in *Proceedings of the thirteenth ACM international conference on Information and knowledge management.* ACM, 2004, pp. 22–31.

[93] L. Barbosa, J. Freire, and A. Silva, “Organizing hidden-web databases by clustering visible web documents”, in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on.* IEEE, 2007, pp. 326–335.

APPENDIX

PROOFS

A-1 Proof of Theorem 1

Theorem. *The expected utility in Equation 6.3 is maximum if the entities are placed in the descending order of the value of the ranking function CE,*

$$CE(e_i) = \frac{U(e_i)C(e_i)}{C(e_i) + \gamma(e_i)}$$

Proof. Consider results e_i and e_{i+1} in positions i and $i+1$ respectively. Let $\mu_i = \gamma(e_i) + C(e_i)$ for notational convenience. The total expected utility from e_i and e_{i+1} when e_i is placed above e_{i+1} is

$$\prod_{j=1}^{i-1} (1 - \mu_j) \left[U(e_i)C(e_i) + (1 - \mu_i)U(e_{i+1})C(e_{i+1})\right]$$

If the order of e_i and e_{i+1} are inverted by placing e_i above e_{i+1}, the expected utility from these entities will be,

$$\prod_{j=1}^{i-1}(1 - \mu_j) \left[U(e_{i+1})C(e_{i+1}) + (1 - \mu_{i+1})U(e_i)C(e_i))\right]$$

Since utilities from all other results in the list will remain the same, the expected utility of placing e_i above e_{i+1} is greater than inverse placement *iff*

$$\begin{array}{rcl} U(e_i)C(e_i) + (1 - \mu_i)U(e_{i+1})C(e_{i+1}) & \geq & U(e_{i+1})C(e_{i+1}) + (1 - \mu_{i+1})U(e_i)C(e_i) \\ & \Updownarrow & \\ \dfrac{U(e_i)C(e_i)}{\mu_i} & \geq & \dfrac{U(e_{i+1})C(e_{i+1})}{\mu_{i+1}} \end{array}$$

This means if entities are ranked in the descending order of $\frac{U(e)C(e)}{C(e)+\gamma(e)}$ any inversions will reduce the profit. Otherwise ranking by $\frac{U(e)C(e)}{C(e)+\gamma(e)}$ is optimal. □

A-2 Proof of Theorem 2

Theorem. *The order by w_ib_i is the same as the order by CE_i for the auction i.e.*

$$w_ib_i \geq w_jb_j \Longleftrightarrow CE_i \geq CE_j$$

Proof. Without loss of generality, we assume that a_i refers to ad in the position i in the descending order of $w_i b_i$.

$$\begin{aligned} CE_i &= \frac{p_i c_i}{\mu_i} \\ &= \frac{b_{i+1} c_{i+1} \mu_i}{\mu_{i+1} c_i} \frac{c_i}{\mu_i} \\ &= \frac{b_{i+1} c_{i+1}}{\mu_{i+1}} \\ &= w_{i+1} b_{i+1} \\ &\geq w_{i+2} b_{i+2} = CE_{i+1} \end{aligned}$$

□

A-3 Proof of Theorem 3

Theorem (Nash Equilibrium)**.** *Without the loss of generality assume that the advertisers are ordered in the decreasing order of $\frac{c_i v_i}{\mu_i}$ where v_i is the private value of the i^{th} advertiser. The advertisers are in a pure strategy Nash Equilibrium if*

$$b_i = \frac{\mu_i}{c_i} \left[v_i c_i + (1 - \mu_i) \frac{b_{i+1} c_{i+1}}{\mu_{i+1}} \right]$$

This equilibrium is socially optimal as well as optimal for search engines for the given CPC's.

Let there are n advertisers. Without loss of generality, let us assume that advertisers are indexed in the descending order of $\frac{v_i c_i}{\mu_i}$. We prove equilibrium in two steps.

Step 1: Prove that

$$w_i b_i \geq w_{i+1} b_{i+1} \tag{A-1}$$

Proof.

$$w_i b_i = \frac{b_i c_i}{\mu_i}$$

Expanding b_i by Equation 6.19,

$$\begin{aligned} w_i b_i &= v_i c_i + (1-\mu_i)\frac{b_{i+1}c_{i+1}}{\mu_{i+1}} \\ &= v_i c_i + (1-\mu_i) w_{i+1} b_{i+1} \\ &= \frac{v_i c_i}{\mu_i}\mu_i + (1-\mu_i) w_{i+1} b_{i+1} \end{aligned}$$

Notice that $w_i b_i$ is a convex linear combination of $w_{i+1}b_{i+1}$ and $\frac{v_i c_i}{\mu_i}$. This means that the value of $w_i b_i$ is in between (or equal to) the values of $w_{i+1}b_{i+1}$ and $\frac{v_i c_i}{\mu_i}$. Hence to prove that $w_i b_i \geq w_{i+1}b_{i+1}$ all we need to prove is that $\frac{v_i c_i}{\mu_i} \geq w_{i+1}b_{i+1}$. This inductive proof is given below.

Induction hypothesis: Assume that

$$\forall_{i \geq j} \frac{v_i c_i}{\mu_i} \geq w_{i+1} b_{i+1}$$

Base case: Prove for $i = N$ i.e. for the bottommost ad.

$$\frac{v_{N-1}c_{N-1}}{\mu_{N-1}} \geq w_N b_N$$

Assuming $\forall_{i>N} b_i = 0$

$w_N b_N = v_N c_N \leq \dfrac{v_N c_N}{\mu_N}$ (as $\mu_N \leq 1$) $\leq \dfrac{v_{N-1}c_{N-1}}{\mu_{N-1}}$ (by the assumed order i.e. by $\frac{v_i c_i}{\mu_i}$)

Induction: Expanding $w_j b_j$ by Equation 6.19,

$$w_j b_j = \frac{v_j c_j}{\mu_j}\mu_j + (1-\mu_j) w_{j+1} b_{j+1}$$

$w_j b_j$ is the convex linear combination, i.e $\frac{v_j c_j}{\mu_j} \geq w_j b_j \geq w_{j+1} b_{j+1}$, as we know that $\frac{v_j c_j}{\mu_j} \geq w_{j+1} b_{j+1}$ by induction hypothesis. Consequently,

$$w_j b_j \leq \frac{v_j c_j}{\mu_j} \leq \frac{v_{j-1} c_{j-1}}{\mu_{j-1}} \text{ (by the assumed order)}$$

This completes the induction. □

Since advertisers are ordered by $w_i b_i$ for pricing, the above proof says that the pricing order is the same as the assumed order in this proof (i.e. ordering by $\frac{v_i c_i}{\mu_i}$). Consequently,

$$p_i = \frac{b_{i+1} c_{i+1} \mu_i}{\mu_{i+1} c_i}$$

As corollary of Theorem 2 we know that $CE_i \geq CE_{i+1}$.

In the second step we prove the equilibrium using results in Step 1.

Step 2: No advertiser can increase his profit by changing his bids unilaterally

Proof of lack of incentive to underbid advertisers below. In the first step let us prove that ad a_i can not increase his profit by decreasing his bid to move to a position $j \geq i$ below.

Inductive hypothesis: Assume true for $i \leq j \leq m$.

Base Case: Trivially true for $j = i$.

Induction: Prove that the expected profit of a_i at $m + 1$ is less or equal to the expected profit of a_i at i.

Let ρ_k denotes the amount paid by a_i when he is at the position k. By inductive hypothesis, the expected profit at m is less or equal to the expected profit at i. So we just need to prove that the expected profit at $m + 1$ is less or equal to the expected profit at m. i.e.

$$\frac{(v_i - \rho_m)}{(1 - \mu_i)} \prod_{l=1}^{m} (1 - \mu_l) \geq \frac{(v_i - \rho_{m+1})}{(1 - \mu_i)} \prod_{l=1}^{m+1} (1 - \mu_l)$$

Canceling the common terms,

$$v_i - \rho_m \geq (v_i - \rho_{m+1})(1 - \mu_{m+1}) \tag{A-2}$$

ρ_m—the price charged to a_i at position m—is based on the Equations 6.16 and 6.19. Since the a_i is moving downward, a_i will occupy position m by shifting ad a_m upwards. Hence the ad just below a_i is a_{m+1}. Consequently, the price charged to a_i when it is at the m^{th} position is,

$$\rho_m = \frac{b_{m+1}c_{m+1}\mu_i}{\mu_{m+1}c_i} = \frac{\mu_i}{c_i}\left[v_{m+1}c_{m+1} + (1-\mu_{m+1})\frac{b_{m+2}c_{m+2}}{\mu_{m+2}}\right]$$

Substituting for ρ_m and ρ_{m+1} in Equation A-2,

$$v_i - \frac{\mu_i}{c_i}\left[v_{m+1}c_{m+1} + (1-\mu_{m+1})\frac{b_{m+2}c_{m+2}}{\mu_{m+2}}\right] \geq \left(v_i - \frac{\mu_i}{c_i}\left[v_{m+2}c_{m+2} + (1-\mu_{m+2})\frac{b_{m+3}c_{m+3}}{\mu_{m+3}}\right]\right)(1-\mu_{m+1})$$

Simplifying, and multiplying both sides by -1

$$\frac{\mu_i}{c_i}\left[v_{m+1}c_{m+1} + (1-\mu_{m+1})\frac{b_{m+2}c_{m+2}}{\mu_{m+2}}\right] \leq v_i\mu_{m+1} + \frac{\mu_i}{c_i}(1-\mu_{m+1})\left[v_{m+2}c_{m+2} + (1-\mu_{m+2})\frac{b_{m+3}c_{m+3}}{\mu_{m+3}}\right]$$

Substituting by b_{m+2} from Equation 6.19 on RHS.

$$\frac{\mu_i}{c_i}\left[v_{m+1}c_{m+1} + (1-\mu_{m+1})\frac{b_{m+2}c_{m+2}}{\mu_{m+2}}\right] \leq v_i\mu_{m+1} + \frac{\mu_i}{c_i}(1-\mu_{m+1})\frac{b_{m+2}c_{m+2}}{\mu_{m+2}}$$

Canceling out the common terms on both sides,

$$\frac{\mu_i}{c_i}v_{m+1}c_{m+1} \leq v_i\mu_{m+1}$$

$$\Updownarrow$$

$$\frac{v_{m+1}c_{m+1}}{\mu_{m+1}} \leq \frac{v_ic_i}{\mu_i}$$

Which is true by the assumed order as $m \geq i$ □

Inductive proof for $m \leq i$ is somewhat similar and enumerated below.

Inductive hypothesis: Assume true for $j \leq m$.

Base Case: Trivially true for $j = i$.

Proof of lack of incentive to overbid ad one above . The case in which a_i increases his bid to move one position up i.e. to $i-1$ is a special case and need to be proved separately. In this case, by moving a single slot up, the index of the ad below a_i will change from $i+1$ to $i-1$ (a difference of two). For all other movements of a_i to a position one above or one below, the index of the advertisers below will change only by one. Since the amount paid by a_i depends on the ad below a_i, this case warrants a slightly different proof,

$$\begin{aligned} (v_i - \rho_i)\prod_{l=1}^{i-1}(1-\mu_l) &\geq (v_i - \rho_{m-1})\prod_{l=1}^{i-2}(1-\mu_l) \\ &\Updownarrow \\ (v_i - \rho_i)(1-\mu_{i-1}) &\geq v_i - \rho_{i-1} \end{aligned}$$

Expanding ρ_i is straight forward.To expand ρ_{i-1}, note that when a_i has moved upwards to $i-1$, the ad just below a_i is a_{i-1}. Since a_{i-1} has not changed its bids, the ρ_{i-1} can be expanded as $\frac{\mu_i}{c_i}\left[v_{i-1}c_{i-1} + (1-\mu_{i-1})\frac{b_i c_i}{\mu_i}\right]$. Substituting for ρ_i and ρ_{i-1},

$$\begin{aligned} \Bigg(v_i - \frac{\mu_i}{c_i}\Bigg[v_{i+1}c_{i+1} + \quad &\geq \quad v_i - \frac{\mu_i}{c_i}\Bigg[v_{i-1}c_{i-1} + \\ (1-\mu_{i+1})\frac{b_{i+2}c_{i+2}}{\mu_{i+2}}\Bigg]\Bigg)(1-\mu_{i-1}) \quad & \quad (1-\mu_{i-1})\frac{b_i c_i}{\mu_i}\Bigg] \end{aligned}$$

Simplifying and multiplying by -1

$$\begin{aligned} v_i\mu_{i-1} + \frac{\mu_i}{c_i}\Bigg[v_{i+1}c_{i+1} + \quad &\leq \quad \frac{\mu_i}{c_i}\left[v_{i-1}c_{i-1} + (1-\mu_{i-1})\frac{b_i c_i}{\mu_i}\right] \\ (1-\mu_{i+1})\frac{b_{i+2}c_{i+2}}{\mu_{i+2}}\Bigg](1-\mu_{i-1}) & \end{aligned}$$

Substituting b_{i+1} from Equation 6.19

$$\begin{aligned} v_i\mu_{i-1} + \frac{\mu_i}{c_i}\frac{b_{i+1}c_{i+1}}{\mu_{i+1}}(1-\mu_{i-1}) &\leq \frac{\mu_i}{c_i}\left[v_{i-1}c_{i-1} + (1-\mu_{i-1})\frac{b_i c_i}{\mu_i}\right] \\ &\Updownarrow \\ v_i\mu_{i-1} + \frac{\mu_i}{c_i}(1-\mu_{i-1})\frac{b_{i+1}c_{i+1}}{\mu_{i+1}} &\leq \frac{\mu_i v_{i-1}c_{i-1}}{c_i} + \frac{\mu_i}{c_i}(1-\mu_{i-1})\frac{b_i c_i}{\mu_i} \end{aligned}$$

We now prove that both the terms in RHS are greater or equal to the corresponding terms in LHS separately.

$$\begin{aligned} v_i\mu_{i-1} &\leq \frac{\mu_i v_{i-1}c_{i-1}}{c_i} \\ &\Updownarrow \\ \frac{v_i c_i}{\mu_i} &\leq \frac{v_{i-1}c_{i-1}}{\mu_{i-1}} \end{aligned}$$

Which is true by our assumed order.

Similarly,

$$\begin{aligned} \frac{\mu_i}{c_i}(1-\mu_{i-1})\frac{b_{i+1}c_{i+1}}{\mu_{i+1}} &\leq \frac{\mu_i}{c_i}(1-\mu_{i-1})\frac{b_i c_i}{\mu_i} \\ &\Updownarrow \\ \frac{b_{i+1}c_{i+1}}{\mu_{i+1}} &\leq \frac{b_i c_i}{\mu_i} \end{aligned}$$

Which is true by Equation A-1 above. This completes the proof for this case. □

Induction: Prove that the expected profit at $m-1$ is less or equal to the expected profit at m. The proof is similar to the induction for the case $m > i$.

Proof. Base case is trivially true.

$$(v_i - \rho_m)\prod_{l=1}^{m-1}(1-\mu_l) \geq (v_i - \rho_{m-1})\prod_{l=1}^{m-2}(1-\mu_l)$$

Canceling common terms,

$$(v_i - \rho_m)(1-\mu_{m-1}) \geq v_i - \rho_{m-1}$$

In this case, note that a_i is moving upwards. This means that a_i will occupy position m by pushing the ad originally at m one position downwards. Hence the original ad at m is the one just below a_i now. i.e.

$$\rho_m = \frac{b_m c_m \mu_i}{\mu_m c_i} = \frac{\mu_i}{c_i}\left[v_m c_m + (1-\mu_m)\frac{b_{m+1}c_{m+1}}{\mu_{m+1}}\right]$$

Substituting for ρ_m and ρ_{m-1}

$$\left(v_i - \frac{\mu_i}{c_i}\left[v_m c_m + (1-\mu_m)\frac{b_{m+1}c_{m+1}}{\mu_{m+1}}\right]\right)(1-\mu_{m-1}) \geq v_i - \frac{\mu_i}{c_i}\left[v_{m-1}c_{m-1} + (1-\mu_{m-1})\frac{b_m c_m}{\mu_m}\right]$$

Simplifying and multiplying by -1

$$v_i\mu_{m-1} + \frac{\mu_i}{c_i}\left[v_m c_m + (1-\mu_m)\frac{b_{m+1}c_{m+1}}{\mu_{m+1}}\right](1-\mu_{m-1}) \leq \frac{\mu_i}{c_i}\left[v_{m-1}c_{m-1} + (1-\mu_{m-1})\frac{b_m c_m}{\mu_m}\right]$$

Substituting by b_m from Equation 6.19

$$v_i\mu_{m-1} + \frac{\mu_i}{c_i}\frac{b_m c_m}{\mu_m}(1-\mu_{m-1}) \leq \frac{\mu_i}{c_i}\left[v_{m-1}c_{m-1} + (1-\mu_{m-1})\frac{b_m c_m}{\mu_m}\right]$$

Canceling common terms,

$$\begin{aligned} v_i\mu_{m-1} &\leq \frac{\mu_i}{c_i}v_{m-1}c_{m-1} \\ &\Updownarrow \\ \frac{v_i c_i}{\mu_i} &\leq \frac{v_{m-1}c_{m-1}}{\mu_{m-1}} \end{aligned}$$

Which is true by the assumed order as $m < i$. □

A-4 Proof of Theorem 4

Theorem (Search Engine Revenue Dominance)**.** *For the same bid values for all the advertisers, the revenue of search engine by CE mechanism is greater or equal to the revenue by VCG.*

Proof. VCG payment of the ad at position i (i.e. a_i) is equal to the reduction in utility of the ads below due to the presence of a_i. For each user viewing the list of ads (i.e. for unit view probability), the total expected loss of ads below

a_i due to a_i is,

$$\begin{aligned}
p_i^{V_u} &= \frac{1}{1-\mu_i}\sum_{j=i+1}^{n} b_j c_j \prod_{k=1}^{j-1}(1-\mu_k) - \sum_{j=i+1}^{n} b_j c_j \prod_{k=1}^{j-1}(1-\mu_k) \\
&= \frac{\mu_i}{1-\mu_i}\sum_{j=i+1}^{n} b_j c_j \prod_{k=1}^{j-1}(1-\mu_k) \\
&= \frac{\mu_i}{1-\mu_i}\prod_{k=1}^{i}(1-\mu_k)\sum_{j=i+1}^{n} b_j c_j \prod_{k=i+1}^{j-1}(1-\mu_k) \\
&= \mu_i \prod_{k=1}^{i-1}(1-\mu_k)\sum_{j=i+1}^{n} b_j c_j \prod_{k=i+1}^{j-1}(1-\mu_k)
\end{aligned}$$

This is the expected lose per user browsing the ad list. Pay per click should be equal to the lose per click. To calculate the pay per click, we divide by the click probability of a_i. i.e.

$$\begin{aligned}
p_i^V &= \frac{\mu_i \prod_{k=1}^{i-1}(1-\mu_k)\sum_{j=i+1}^{n} b_j c_j \prod_{k=i+1}^{j-1}(1-\mu_k)}{c_i \prod_{k=1}^{i-1}(1-\mu_k)} \\
&= \frac{\mu_i}{c_i}\sum_{j=i+1}^{n} b_j c_j \prod_{k=i+1}^{j-1}(1-\mu_k)
\end{aligned}$$

Converting to recursive form,

$$\begin{aligned}
p_i^V &= \frac{b_{i+1}\mu_i}{c_i} c_{i+1} + (1-\mu_{i+1})\frac{\mu_i c_{i+1}}{c_i \mu_{i+1}} p_{i+1}^V \\
&= \frac{b_{i+1}\mu_i c_{i+1}}{c_i \mu_{i+1}}\mu_{i+1} + (1-\mu_{i+1})\frac{\mu_i c_{i+1}}{c_i \mu_{i+1}} p_{i+1}^V \qquad \text{(A-3)}
\end{aligned}$$

For the CE mechanism payment from Equation 6.16 is,

$$p_i^{CE} = \frac{b_{i+1} c_{i+1} \mu_i}{\mu_{i+1} c_i}$$

Note that p_i^V is convex combination of P_i^{CE} and $\frac{\mu_i c_{i+1}}{c_i \mu_{i+1}} p_{i+1}^V$, and hence is between these two values. To prove that $p_i^{CE} \geq p_i^V$ all we need to prove is that $P_i^{CE} \geq \frac{\mu_i c_{i+1}}{c_i \mu_{i+1}} p_{i+1}^V \Leftrightarrow b_i \geq p_i^V$. This directly follows from individual rationality property of VCG. Alternatively, a simple recursion with base case as $p_N^V = 0$ (bottommost ad) will prove the same. Note that we consider only the ranking (not selection), and hence the VCG pricing of the bottommost ad in the ranking is zero. □

A-5 Proof of Theorem 5

Theorem (Equilibrium Revenue Equivalence)**.** *At the equilibrium in Theorem 3, the revenue of search engine is equal to the revenue of the truthful dominant strategy equilibrium of VCG.*

Proof. Rearranging Equation A-3 and substituting true values for bid amounts,

$$p_i^V = \frac{\mu_i}{c_i}\left[v_{i+1}c_{i+1} + \frac{(1-\mu_{i+1})c_{i+1}}{\mu_{i+1}}p_{i+1}^V\right]$$

For the CE mechanism, substituting equilibrium bids from Equation 6.19 in payment (Equation 6.16),

$$p_i^{CE} = \frac{b_{i+1}c_{i+1}\mu_i}{\mu_{i+1}c_i} = \frac{\mu_i}{c_i}\left[v_{i+1}c_{i+1} + (1-\mu_{i+1})\frac{b_{i+2}c_{i+2}}{\mu_{i+2}}\right]$$

Rewriting b_{i+2} in terms of p_{i+1},

$$\begin{aligned} p_i^{CE} &= \frac{\mu_i}{c_i}\left[v_{i+1}c_{i+1} + \frac{(1-\mu_{i+1})c_{i+1}}{\mu_{i+1}}p_{i+1}^{CE}\right] \\ &= p_i^V \quad (\textit{iff } p_{i+1}^V = p_{i+1}^{CE}) \end{aligned}$$

Ad at the bottommost position pays same amount zero, a simple recursion will prove that the payment for all positions for both VCG and the proposed equilibrium is the same. □

A-6 Proof of Theorem 6

Theorem. *Diversity ranking optimizing expected utility in Equation 6.22 is NP-Hard.*

Proof. Independent set problem can be formulated as a ranking problem considering similarities. Consider an unweighed graph G of n vertices $\{e_1, e_2, ..e_n\}$ represented as an adjacency matrix. This conversion is clearly polynomial time. Now, consider the values in the adjacency matrix as the similarity values between the entities to be ranked. Let the entities have the same utilities, perceive relevances and abandonment probabilities. In this set of n entities from $\{e_1, e_2, .., e_n\}$, clearly the optimal ranking will have k pairwise independent entities as the top k entities for a maximum possible value of k. But the set of k independent entities corresponds to the maximum independent set in graph G. □

BIOGRAPHICAL SKETCH

Raju Balakrishnan received Bachelor's Degree in Computer Science and Engineering from Cochin University of Science and Technology, Kerala, India in 2001. After working in IBM for five years, he joined Arizona State University for his PhD in 2006.

His research interests are large scale data integration and search, ranking for the deep web and ads, auction mechanisms of online Ads and real time auctions. He is particularly interested in deriving deep insights from large data collections by combined analysis of multiple data sources and types.

He will be joining Groupon inc. at Palo Alto, California as a Data Scientist in August 2012.