Middle-East Journal of Scientific Research 20 (11): 1431-1435, 2014
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2014.20.11.1417

# Classification-Aware Hidden-Web Text Database Selection

*P. Kavitha*

Department of Information Technology,
Bharath University, India

**Abstract:** Many valuable text databases on the web have noncrawlable contents that are "hidden" behind search interfaces. Metasearchers are helpful tools for searching over multiple such "hidden-web" text databases at once through a unified query interface. An important step in the metasearching process is database selection, or determining whch databases are the most relevant for a given user query. Our algorithm is the first to construct In this paper we present algorithms that return the top results for a query, ranked according to an IR-style ranking function, while operating on top of a source with a Boolean query interface with no ranking capabilities (or a ranking capability of no interest to the end user). The algorithms generate a series of conjunctive queries that return only documents that are candidates for being highly ranked according to a relevance metric. Our approach can also be applied to other settings where the ranking is monotonic on a set of factors (query keywords in IR) and the source query interface is a Boolean expression of these factors. Our comprehensive experimental evaluation on the PubMed database and a TREC dataset show that we achieve order of magnitude improvement compared to the current baseline approaches.

**Key words:** Hidden-web databases · Keyword Search · Top-k ranking

## INTRODUCTION

Many online or local data sources provide powerful querying mechanisms but limited ranking capabilities. For instance, PubMed 1. allows users to submit Boolean keyword queries on the biomedical publications database, but ranks the query results by publication date only. Similarly, the US Patent and Trademark Office (USPTO) 2. allows Boolean keyword queries or searching [1] patents but only ranks by patent date. Furthermore, job search databases, such as the job search of LinkedIn, allow users to sort job listings by date or title (alphabetically), but not by IR relevance of the job posting to the submitted query. As a more recent example, the micro-blogging service Twitter 4. offers a highly expressive Boolean search interface but ranks the results by date only [2]. In most cases, these sources do not allow downloading and indexing of data or the size of the underlying database makes any comprehensive download an expensive operation. Often, the user prefers a ranking other than the default sorting (e.g., by date) provided by the source. For instance, a user of the PubMed or USPTO Web sites may prefer a ranking by relevance measured 3 by an Information Retrieval (IR) ranking function, as opposed to a date-based retrieval. Given that traditional IR ranking functions like Okapi and BM25 implicitly assume disjunctive (OR) semantics [3], the naïve approach would be to submit to the database a disjunctive query with all query keywords, retrieve all the returned documents and then rank them according to the relevance metric of choice. However, this would be very expensive due to the large number of results returned by disjunctive queries. For example, consider the query "immunodeficiency virus structure," [4] an example query used to teach information specialists how to search the PubMed database. Executing the corresponding disjunctive query immunodeficiency OR virus OR structure" on PubMed returns 1,451,446 publication results [5].

Downloading and ranking them is infeasible for an interactive query system, even if the source is on the local network. The problem becomes even more critical if we use the public web services provided by PubMed for programmatic (API) access over the web [6]. Given the large overhead incurred when retrieving publications, PubMed imposes quotas on the amount of data an application can retrieve per minute, rendering infeasible

**Corresponding Author:** P. Kavitha, Department of Information Technology, Bharath University, India.

any attempt to download large number of documents [7]. To overcome such problems, in this paper, we present algorithms to compute the top results for an IR ranked query, over a source with a Boolean query interface but without any ranking capabilities (or with a ranking function that is generally uncorrelated to the user's ranking: e.g., by date). A key idea behind our technique is to use a probabilistic modeling approach and estimate the distribution of document scores that are expected to be returned by the database [8]. Hence, we can estimate what are the minimum cutoff scores for including a document in the list of highly ranked documents. To achieve this result over a database that allows only query-based access of documents, we generate a querying strategy that submits a minimal sequence of conjunctive queries to the source. (Note that conjunctive queries are cheaper since they return significantly fewer results than disjunctive ones.) After every submitted conjunctive query we update the estimated probability distributions of the query keywords in the database and decide whether the algorithm should terminate given the user's results con?dence requirement [9] or whether further querying is necessary; in the latter case, our algorithm also decides which is the best query to submit next. For instance, for the above query immunodeficiency virus structure", the algorithm may first execute "immunodeficiency AND virus AND structure", then "immunodeficiency AND structure" and then terminate, after estimating that the returned documents contain all the documents that would be highly ranked under an IR-style ranking mechanism. As we will see, our work fits into the "exploration vs. exploitation" paradigm since we iteratively explore the source by submitting conjunctive queries to learn the probability distributions of the keywords and at the same time we exploit the returned "document samples" to retrieve results for the user query [10].

**Related Work**

**Top-K Queries:** A significant amount of work has been devoted to the evaluation of top-k queries in databases. Ilyas et al. provide a survey of the research on top-k queries on relational databases [12]. This line of work typically handles the aggregation of attribute values of objects in the case where the attribute values lie in different sources or in a single source For example, Bruno et al. consider the problem of ordering a set of restaurants by distance and price. They present an optimal sequence of random or sequential accesses on the sources (e.g., Zagat for price and [13].

Mapquest for distance) in order to compute the topk restaurants. Since the concept of top-k is typically a heuristic itself for locating the most interesting items in the database, Theobald et al. describe a framework for generating an approximate top-k answer, with some probabilistic guarantees. In our work, we use the same idea; the main and crucial difference is that we only [14-17] have "random access" to the underlying database (i.e., through querying) and no "sorted access." Theobald et al. assumed that at least one source provides "sorted access" to the underlying content.

**Exploration Vs. Exploitation:** The idea of the exploitation/exploration tradeoff (also called the "multi-armed bandit problem") is to determine a strategy of sequential execution of actions, each of which has a stochastic payoff. While executing an action we get back some (uncertain) payoff and at the same time we get some information that allows us to decrease the uncertainty of the payoff of future actions. The problem has been ?rst posed in the 1930's and has been used to model problems in a wide variety of areas, ranging from medicine and economics to ad placement in web pages. In our work, we are trying to maximize the payoff/exploitation of each query (which is the number of new, relevant top-k documents that the query retrieves) while minimizing the expense/exploration (number of queries sent and documents retrieved). Deep Web: Our work bears some similarities to the problem of searching and extracting data from the Deep Web [18] databases. Meng et al.examine the problem of estimating the number of useful documents in the database, assuming that the statistics about the frequency and the tf.idf weights of each word in the database is given. In our work, we estimate such statistics on-the-fly, as part of the explorative sampling process. Ntoulas et al.attempt to download the contents of a Deep Web database by issuing queries through a web form interface. The goal of Ntoulas et al. is to download and index the contents of databases with limited query capabilities, whereas in our case the focus is on achieving on-the-fly ranking of query results, on top of sources with no (or non-useful) ranking capabilities. An alternative approach is to characterize databases by extracting a small sample of documents that is then used to describe the contents of the database. For example, it is possible to use query-based sampling to extract such a document sample, generate estimates for the distribution of each term and then use the estimates to guide the choice of queries that should be submitted to the database. In the experimental section, we compare against this "static sampling" alternative and demonstrate the superiority of the dynamic sampling technique, which dynamically generates estimates tailored to the query at hand.

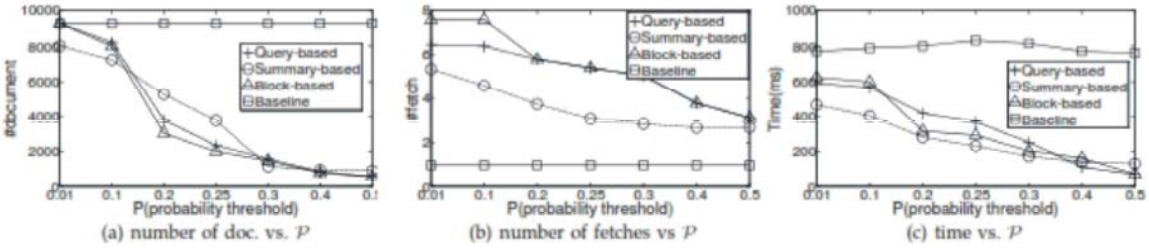


Fig. 2: LocalPubMed:Varying

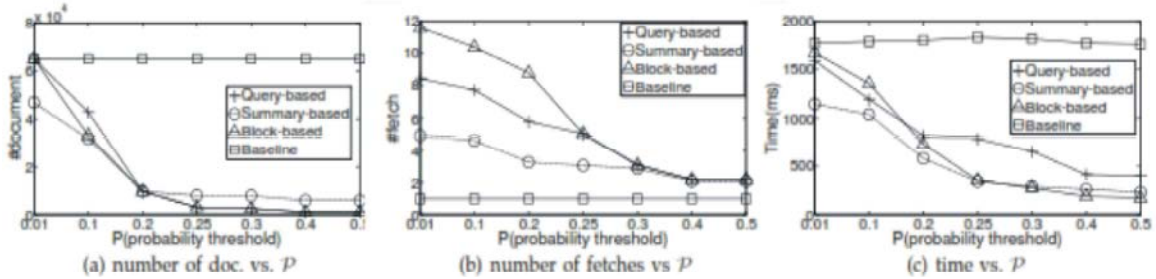


Fig. 3: LocalTREC:Varying

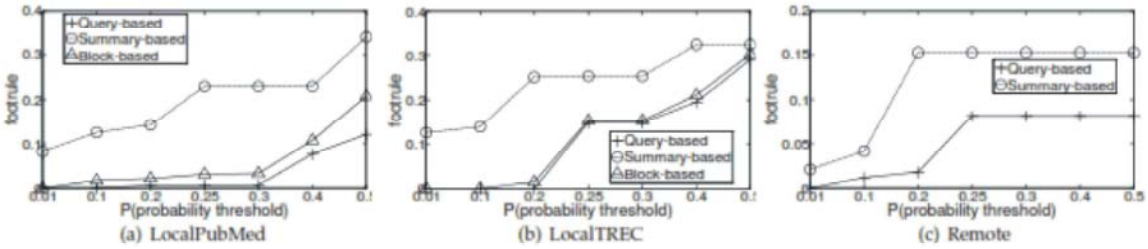


Fig. 4: Footrule VS p

**Experiments:** We experimentally evaluate the performance and quality of the retrieval algorithms. We compare the Query-based probability estimation strategy described in Section 5.4 to the Summary-based estimation strategy of Section 5.2 and also consider the Total vs. Block variants of the top-k [15] querying algorithm of Section 5.5. For that, we compare the following algorithm variants:

**Baseline:** This algorithm submits the disjunction of all query keywords to the database and retrieves all matching results. Documents that do not match this disjunctive query and hence are not returned, are guaranteed to have zero tf.idf score. Then this algorithm computes the IR score for each document and returns the true top-k to the user. Therefore, this algorithm is guaranteed to generate a perfect ranking, at the expense of a significant cost of downloading all documents before ranking them.

**Experiments on Local Datasets**

**Varying P:** First, we examine the effect of P in the performance of our algorithms. P is the parameter that defines the confidence that the returned results are close to the optimal. Smaller values of P mean that the algorithms tries harder to approximate the optimal list, while large [16] values of P mean that the algorithm can stop earlier, returning more rough approximations of the optimal list. In Figures 2 and 3 we set the number of keywords to 3 and fix k = 50. For Block-based algorithm, we set the Block size to 2000. We vary P from 0.01 to 0.5. Figures 2(a) and 3(a) show that Summary-based, Query-based and Block-based fetch fewer documents as P grows. We observe that Block-based retrieves slightly fewer documents but submits more conjunctive [17-21] queries compared with Query-based (called fetches in Figures 2(b) and 3(b)). As expected, Summary-based retrieves the least documents in most cases. (As discussed in Section 5.2, the summary-based algorithm retrieves 300 documents for the initial document summary to generate the estimates but we do not include this onetime cost in the reported results.) Moveover, in Fig 2(b) we see that for P _ 0:2, Query-based and Block-based coincide, because the number of the documents Blockbased fetches is less than Block size B. The same phenomenon also happens in Fig. 3(b) for P _ 0:25 fetches which incur an overhead.

Summary-based is the fastest because it performs the fewest fetches (queries) and also the lambda estimation is performed off-line. Although the Summary-based algorithm is the most ef?cient, we observed that the speed comes at the expense of the quality of the results. In terms of quality, Figures 4(a) and 4(b) show that both Query-based and Block-based achieve excellent Footrule values for P up to 0.3 (for LocalPubMed) or 0.2 (for LocalTREC) while Summary-based is the worst in all cases as expected: this is the result of the rough probability estimates. In the rest of this section, due to space constraints, we only report the results for LocalPubMed, given that the results of LocalTREC follow similar trends.

## CONCLUSIONS

We presented a framework and efficient algorithms to build a ranking wrapper on top of a documents data source that only serves Boolean keyword queries. This setting is common in various major databases today, including PubMed and USPTO. Our algorithm submits a minimal sequence of conjunctive queries instead of a very expensive disjunctive one. The query score distributions of the candidate conjunctive queries are learned as documents are retrieved from the source. Our comprehensive experimental evaluation on the PubMed database shows that we achieve order of magnitude improvement compared to the baseline approach. We found that applying tf probabilistic estimation techniques and processing a whole conjunctive query ata time (without splitting it to blocks) lead to better performance.

## REFERENCES


1. Madhavan, J., D. Ko, L. Kot, V. Ganapathy, A. Rasmussen and A.Y. Halevy, 2008. Google's Deep Web Crawl, Proc, VLDB, 1(2): 1241-1252.
2. Ntoulas, A., P. Zerfos and J. Cho, 2005. Downloading Textual Hidden Web Content by Keyword Queries,″ Proc. Fifth ACM and IEEE Joint Conf. Digital Libraries (JCDL '05).
3. Herskovic, J.R. and E.V. Bernstam, 2005. Using Incomplete Citation Data for Medline Results Ranking, Proc. AMIA Ann. Symp, pp: 316-20.
4. Lu, Z., W. Kim and W.J. Wilbur, 2009. Evaluating Relevance Ranking Strategies for Medline Retrieval, J. Am. Medical Informatics Assoc., 16(1): 32-36.
5. Salton, G. and M.J. McGill, 1986. Introduction to Modern Information Retrieval. McGraw-Hill, Inc.,
6. Singhal, A., Modern Information Retrieval, A Brief Overview, 2001. Bull. IEEE CS Technical Committee on Data Eng., 24(4): 35-42. http://singhal. infoieee2001.pdf.
7. Robertson, S.E., S. Walker, S. Jones, M. Hancock-Beaulieu and M. Gatford, 1994. Okapi at Trec-3, Proc. Text Retrieval Conf. (TREC).
8. Geer, R.C., *et al*., 2007. Ncbi Advanced Workshop for Bioinformatics Information Specialists: Sample User Questions and Answers,"http://www.ncbi.nlm. nih.gov/Class/NAWBIS index.html.
9. Berry, D.A. and B. Fristedt, 1985. Bandit Problems: Sequential Allocation of Experiments. Springer.
10. Lee, J., J. Lee and H. Lee, 2003. Exploration and Exploitation in the Presence of Network Externalities, Management Science, 49(4): 553-570.
11. Macready, W.G. and D.H. Wolpert, Apr. 1998. Bandit Problems and the Exploration/Exploitation Tradeoff," IEEE Trans. Evolutionary Computation, 2(1): 2-22.
12. Hristidis, V., Y. Hu and P.G. Ipeirotis, 2010. Ranked Queries Over Sources with Boolean Query Interfaces without Ranking Support, Proc. 26th IEEE Int'l Conf. Data Eng. (ICDE '10).
13. Ilyas, I.F., G. Beskales and M.A. Soliman, 2008. A Survey of Top-K Query Processing Techniques in Relational Database Systems," ACM Computing Survey, 40(4): 1-58.
14. Udayakumar, R., V. Khanna, T. Saravanan and G. Saritha, 2013. Retinal Image Analysis Using Curvelet Transform and Multistructure Elements Morphology by Reconstruction, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 16(12): 1798-1800.
15. Udayakumar, R., V. Khanna, T. Saravanan and G. saritha, 2013, Cross Layer Optimization For Wireless Network (Wimax), Middle-East Journal of Scientific Res., ISSN:1990-9233, 16(12): 1786-1789.
16. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. A frame work for modelling task coordination in Multi-agent system, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1851-1856.
17. Thooyamani, K.P., V.R. Khanaa and Udayakumar, 2013. An Integrated Agent System for E-mail Coordination using Jade, Indian Journal of Science and Technology, ISSN: 0974-6846, 6(6): 4758-4761.
18. Tatyana Aleksandrovna Skalozubova and Valentina Olegovna Reshetova, 2013. Leaves of Common Nettle (*Urtica dioica* L.) As a Source of Ascorbic Acid (Vitamin C),World Applied Sciences Journal, 28(2): 250-253.

19. Rassoulinejad-Mousavi, S.M., M. Jamil and M. Layeghi, 2013. Experimental Study of a Combined Three Bucket H-Rotor with Savonius Wind Turbine, World Applied Sciences Journal, 28(2): 205-211.
20. Vladimir, G. Andronov, 2013. Approximation of Physical Models of Space Scanner Systems World Applied Sciences Journal, 28(4): 528-531.
21. Naseer Ahmed, 2013. Ultrasonically Assisted Turning: Effects on Surface Roughness World Applied Sciences Journal, 27 (2): 201-206.