ObjectRank: A System for Authority-based Search on Databases

Heasoo Hwang
Computer Science & Engineering
UC San Diego
La Jolla, CA
heasoo@cs.ucsd.edu

Vagelis Hristidis*
School of Computing and Information Sciences
Florida International University
Miami, FL
vagelis@cis.fiu.edu

Yannis Papakonstantinou
Computer Science & Engineering
UC San Diego
La Jolla, CA
yannis@cs.ucsd.edu

1. INTRODUCTION

ObjectRank [1] is a system to perform authority-based keyword search on databases, inspired by PageRank [3]. PageRank is an excellent tool to rank the global importance of the pages of the Web, proven by the success of Google [1]. However, Google uses PageRank as a tool to measure the global importance of the pages, independently of a keyword query. (Google uses traditional IR techniques to estimate the relevance of a page to a keyword query, which is then combined with the PageRank value to calculate the final score of a page.) We appropriately extend and modify PageRank to perform keyword search on databases.

For example, consider the publications database of Figure 1, where edges denote citations (edges start from citing and end at cited paper), and the keyword query "Sorting". Then, using the original variant of ObjectRank [1], the *"Access Path Selection in a Relational Database Management System"* paper would be ranked highest, because it is cited by four papers containing "sorting" (or "sort"). The *"Fundamental Techniques for Order Optimization"* paper would be ranked second, since it is cited by only three "sorting" papers.

We have found through user surveys [1, 4] that the quality of the results of ObjectRank dramatically changes according to various calibration parameters. One of the most interesting parameters is the specificity metric, for which the novel method of Inverse ObjectRank is employed [4]. Ranking solely using ObjectRank, as in the above example, induces the following problem: Objects with general context, like the *"Access Path Selection"* of Figure 1, are ranked higher than more focused (specific) objects, like the *"Fundamental Techniques for Order Optimization"* paper. Intuitively, one might want to rank the *"Fundamental Techniques for Order Optimization"* paper higher because this paper is mostly cited by "sorting" papers, whereas the *"Access Path Selection"* paper is not only cited by "sorting" papers but by many (the three papers on the top right) papers irrelevant to "sorting". We also identified other calibration parameters other than the specificity metric above.

Our system ranks papers as well as authors according to their authority and specificity with respect to the given keywords. We extracted data from the well-known DBLP publications database to demonstrate the power of the "random walk" model for the purposes of discovering authoritative and specific (with respect to the keywords) publications and authors. The ObjectRank demo system is available online at two mirror sites:
http://www.db.ucsd.edu/ObjectRank/, http://dbir.cis.fiu.edu/BibObjectRank/.

2. DATA MODEL

We view a database as a labeled graph, which is a model that captures both relational and XML databases, as well as the web. The *data graph* $D(V, E_D)$ is a labeled directed graph where every node v has a label $\lambda(v)$ and a set of keywords. For example, the node "SIGMOD" of Figure 2 has label "Conference" and the set of keywords {"SIGMOD"}. Each node represents an *object* of the database.

The *authority transfer graph* $G(V, E)$ represents the authority flows between the nodes of the data graph. Given a data graph $D(V, E_D)$, $G(V, E)$ is created as follows. For every edge $e = (u \to v) \in E_D$ we create (potentially) two edges $e^f = (u \to v)$ and $e^b = (v \to u)$. The edges e^f and e^b are annotated with *authority transfer rates* $a(e^f)$ and $a(e^b)$, which denote the maximum portion of authority that can flow between u and v. The authority transfer rates are assigned for every type of semantic connection by domain experts. For the demo, we experimented with various sets of rates and performed user surveys [1] which lead to the following set:

$$a(Paper \xrightarrow{cites} Paper) = 0.7 \quad a(Paper \xrightarrow{cited} Paper) = 0$$
$$a(Paper \to Author) = a(Author \to Paper) = 0.2$$
$$a(Paper \to Conference) = a(Conference \to Paper) = 0.3$$

For example, if the edge is a citation edge, then 0.7 of the authority of the citing paper goes to the cited paper, whereas no authority goes back to the citing paper.

3. DEMO DESCRIPTION

3.1 Overview

Our demo system performs authority-based keyword search on bibliographic databases. It also provides calibration parameters such as the specificity metric and the quality metric. Users can specify various combinations of calibration values to control the behavior of the system.

A user inputs (a) a keyword query, (b) a choice for combining semantics (AND or OR), (c) the importance of global quality of the results (i.e., Global ObjectRank), (d) the importance of containing the actual query keywords (translated to a damping factor value d), and (e) a specificity metric (i.e., Inverse ObjectRank). The output

* Partly supported by NSF grant IIS-0534530.

[1] www.Google.com

SIGMOD 2006, June 27–29, 2006, Chicago, Illinois, USA.
Copyright 2006 ACM 1-59593-256-9/06/0006 ...$5.00.

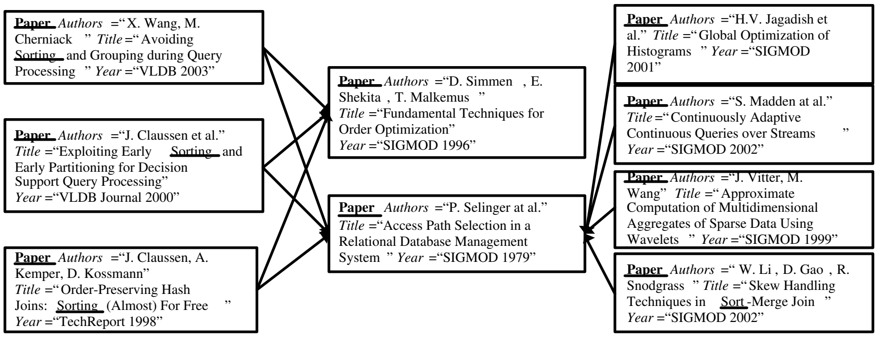
Paper Authors ="X. Wang, M. Cherniack " Title ="Avoiding Sorting and Grouping during Query Processing " Year ="VLDB 2003"
Paper Authors ="J. Claussen et al." Title ="Exploiting Early Sorting and Early Partitioning for Decision Support Query Processing" Year ="VLDB Journal 2000"
Paper Authors ="J. Claussen, A. Kemper, D. Kossmann" Title =" Order-Preserving Hash Joins: Sorting (Almost) For Free " Year ="TechReport 1998"
Paper Authors ="D. Simmen , E. Shekita , T. Malkemus " Title ="Fundamental Techniques for Order Optimization" Year ="SIGMOD 1996"
Paper Authors ="P. Selinger at al." Title ="Access Path Selection in a Relational Database Management System " Year ="SIGMOD 1979"
Paper Authors ="H.V. Jagadish et al." Title ="Global Optimization of Histograms " Year ="SIGMOD 2001"
Paper Authors ="S. Madden at al." Title =" Continuously Adaptive Continuous Queries over Streams " Year ="SIGMOD 2002"
Paper Authors ="J. Vitter, M. Wang" Title =" Approximate Computation of Multidimensional Aggregates of Sparse Data Using Wavelets " Year ="SIGMOD 1999"
Paper Authors =" W. Li , D. Gao , R. Snodgrass " Title =" Skew Handling Techniques in Sort-Merge Join " Year ="SIGMOD 2002"

Figure 1: Instance of a Publications Database

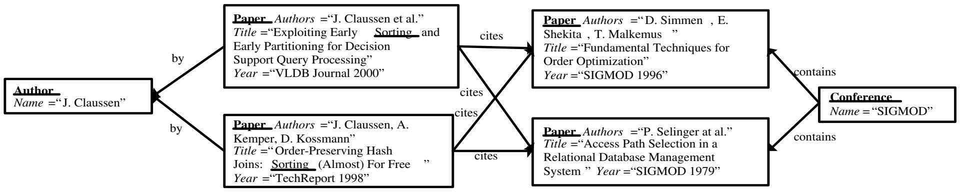
Author Name =" J. Claussen"
by
by
Paper Authors ="J. Claussen et al." Title ="Exploiting Early Sorting and Early Partitioning for Decision Support Query Processing" Year ="VLDB Journal 2000"
Paper Authors ="J. Claussen, A. Kemper, D. Kossmann" Title =" Order-Preserving Hash Joins: Sorting (Almost) For Free " Year ="TechReport 1998"
cites
cites
cites
cites
Paper Authors ="D. Simmen , E. Shekita , T. Malkemus " Title ="Fundamental Techniques for Order Optimization" Year ="SIGMOD 1996"
Paper Authors ="P. Selinger at al." Title ="Access Path Selection in a Relational Database Management System " Year ="SIGMOD 1979"
contains
contains
Conference Name ="SIGMOD"

Figure 2: A Subset of the Instance in Figure 1

of the system is a ranked list of nodes of the database (to be more formal, of the authority transfer graph) according to the input parameters based on the ranking function in [4].

3.2 Dataset for the Demo

We use a bibliographic database for our ObjectRank system demo. It was collected using the following method. First, we downloaded all publications and citations from the DBLP database[2]. We noticed that this source is missing many citations, which greatly degrades the quality of link-based analysis. To tackle this deficiency we used Citeseer[3] as an additional citations' source. We built a web crawler to retrieve these citations since we found that the exported files of Citeseer are to a large degree inaccurate.

3.3 ObjectRank

Conceptually, given a query keyword w, the ObjectRank value $r^w(v)$ of an object/node v of the data graph is computed as follows: Myriads of random surfers are initially found at the objects containing the keyword "sorting", which we call base set, and then they traverse the database graph. In particular, at any time step a random surfer is found at a node and either (i) makes a move to an adjacent node by traversing an edge, or (ii) moves back to a "sorting" node. Notice how ObjectRank produces keyword-specific rankings, in contrast to the global ranking of PageRank.

3.4 ObjectRank with Calibration Parameters

Specificity Metric - *Inverse ObjectRank* By analyzing the example in Figure 3, we can observe how the specificity factor affects the top-10 paper list obtained by ObjectRank for the query "Concurrency Control". The difference in the two results is that for Result (a) no specificity metric was used, while for Result (b) we used Inverse ObjectRank. To measure the quality of these results we use the bibliography section of each chapter in a database texbook [5]. We compare the recall of the top 10 papers in Results (a) and (b) with respect to the set P_{CC} of papers in the bibliography sections of the chapters on "Concurrency Control", which are viewed as the ground truth.

For Result (a), six of the papers are found in P_{CC}, meaning that six papers are specific to the given query. However, this result also includes general publications like *"Notes on Data Base Operating System"*, which is cited by many "Concurrency Control" papers, but it is much more general. To avoid such general papers, we incorporate Inverse ObjectRank in the ranking formula (Result (b) where eight papers are found in P_{CC}).

Inverse ObjectRank [4] is a keyword-specific metric of specificity, based on the link-structure of the data graph. In particular, given a keyword w, the Inverse ObjectRank score $p^w(v)$ of node v shows how specific v is with respect to w. In terms of the random surfer model, $p^w(v)$ is the probability that starting from v and following the edges on the opposite direction we are on a node containing w at a specific point in time. As is the case for ObjectRank, the random surfer at any time step may get bored and go back to v.

Quality Metric - *Global ObjectRank* One may be interested in the global importance of papers, which corresponds to the global quality input in Section 3.1. The global (keyword-independent) quality of the results is represented by their Global ObjectRank, which is computed by executing the ObjectRank algorithm with all nodes of the authority flow graph in the base set. Incorporating Global ObjectRank in the ranking function benefits objects with high query-independent authority. In the demo site, Global ObjectRank is incorporated in the ranking formula by setting the value of the 'Global ObjectRank' parameter to 'INCLUDE'. However, we found that this often results in papers of very high global importance being ranked on top even though they are not highly relevant to the given query.

More Calibration Parameters Our demo system provides two more calibration parameters. One is the importance of the results actually containing the query keywords. This parameter determines the importance of a result actually containing the keywords versus being referenced by nodes containing them, which corresponds to the damping factor d in ObjectRank computation [1]. The damping factor determines the portion of ObjectRank that an object transfers to its neighbors as opposed to keeping to itself. It was first introduced in the original PageRank paper [3], where it was used to ensure convergence in the case of PageRank sinks. However, in our work it has a new meaning since by decreasing d, we favor objects

[2] http://www.informatik.uni-trier.de/ ley/db/

[3] http://citeseer.ist.psu.edu/

(a) parameters = (NOT CRUCIAL, IGNORE, NONE)

1 **The Notions of Consistency and Predicate Locks in a Database System.** *Commun. ACM* 1976. Kapali P. Eswaran, Jim Gray, Raymond A. Lorie, Irving L. Traiger
2 **Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment.** *J. ACM* 1973. James W. Layland, C. L. Liu
3 **Concurrency Control and Recovery in Database Systems.** *book* 1987. Philip A. Bernstein, Nathan Goodman, Vassos Hadzilacos
4 **On Optimistic Methods for Concurrency Control.** *ACM Trans. Database Syst.* 1981. H. T. Kung, John T. Robinson
5 **Notes on Data Base Operating Systems.** *Advanced Course: Operating Systems* 1978. Jim Gray
6 **A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases.** *ACM Trans. Database Syst.* 1979. Robert H. Thomas
7 **Concurrency Control in Distributed Database Systems.** *ACM Comput. Surv.* 1981. Philip A. Bernstein, Nathan Goodman
8 **Towards a Unified Theory of Concurrency Control and Recovery.** *PODS* 1993. Hans-Jörg Schek, Gerhard Weikum, Haiyan Ye
9 **Concurrency Control in Advanced Database Applications.** *ACM Comput. Surv.* 1991. Naser S. Barghouti, Gail E. Kaiser
10 **The serializability of concurrent database updates.** *J. ACM* 1979. Christos H. Papadimitriou

(b) parameters = (NOT CRUCIAL, IGNORE, sqrt(Inverse ObjectRank))

1 **Concurrency Control and Recovery in Database Systems.** *book* 1987. Philip A. Bernstein, Nathan Goodman, Vassos Hadzilacos
2 **On Optimistic Methods for Concurrency Control.** *ACM Trans. Database Syst.* 1981. H. T. Kung, John T. Robinson
3 **Concurrency Control in Distributed Database Systems.** *ACM Comput. Surv.* 1981. Philip A. Bernstein, Nathan Goodman
4 **A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases.** *ACM Trans. Database Syst.* 1979. Robert H. Thomas
5 **The Notions of Consistency and Predicate Locks in a Database System.** *Commun. ACM* 1976. Kapali P. Eswaran, Jim Gray, Raymond A. Lorie, Irving L. Traiger
6 **Towards a Unified Theory of Concurrency Control and Recovery.** *PODS* 1993. Hans-Jörg Schek, Gerhard Weikum, Haiyan Ye
7 **Concurrency Control Performance Modeling: Alternatives and Implications.** *ACM Trans. Database Syst.* 1987. Rakesh Agrawal, Michael J. Carey, Miron Livny
8 **System Level Concurrency Control for Distributed Database Systems.** *ACM Trans. Database Syst.* 1978. Philip M. Lewis II, Daniel J. Rosenkrantz, Richard Edwin Stearns
9 **Concurrency Control in Advanced Database Applications.** *ACM Comput. Surv.* 1991. Naser S. Barghouti, Gail E. Kaiser
10 **Experimental Evaluation of Real-Time Optimistic Concurrency Control Schemes.** *VLDB* 1991. Jiandong Huang, Krithi Ramamritham, John A. Stankovic, Donald F. Towsley

Figure 3: Top 10 paper lists on "Concurrency Control" with calibration parameters (*Containment of actual keywords, Global ObjectRank, Specificity metric*)

(a)

47.31	11.44	An XML Indexing Structure with Relative Region Coordinate. Dao Dinh Kha, ICDE 2001
41.02	3.08	DataGuides: Enabling Query ... Optimization in Semistructured... Roy Goldman, VLDB 1997
7.44	28.43	Access Path Selection in a RDBMS. Patricia G. Selinger, SIGMOD 1979
31.44	3.24	Querying Object-Oriented Databases. Michael Kifer, SIGMOD 1992
26.73	3.09	A Query … Optimization Techniques for Unstructured Data. Peter Buneman, SIGMOD 1996

(b)

47.31	11.44	An XML Indexing Structure with Relative Region Coordinate. Dao Dinh Kha, ICDE 2001
7.44	28.43	Access Path Selection in a RDBMS. Patricia G. Selinger, SIGMOD 1979
2.04	102.1	R-Trees: A Dynamic Index Structure for Spatial Searching. Antonin Guttman, SIGMOD 1984
1.73	112.7	The K-D-B-Tree: A Search Structure For Large … Indexes. John T. Robinson, SIGMOD 1981
41.02	3.08	DataGuides: Enabling Query … Optimization in Semistructured... Roy Goldman, VLDB 1997

Figure 4: Top 5 papers on "XML Index", with and without emphasis on "XML"

that contain the actual query keywords (i.e., objects in the base set). Typical values for d are 0.85 for normal behavior and 0.3 to favor objects that actually contain the keywords. In the demo, setting this parameter to 'Not Crucial' translates to $d = 0.85$ whereas 'Crucial' to $d = 0.3$.

The other calibration parameter is the weight of each query keyword. If the ObjectRank values of all query keywords are given equal weight, the more popular keywords are favored. The reason is that the distribution of ObjectRank values is more skewed when the size of the base set increases, because the top objects tend to receive more references. For example, consider two results for the query "XML Index" shown in Figure 4. Result (b) corresponds to the situation described above. It noticeably favors the "Index" keyword over the "XML" one. The first paper is the only one in the database that contains both keywords in the title. However, the next three results are all classic works on indexing and do not apply directly to XML. Intuitively, "XML" as a more specific keyword is more important to the user. We conducted a survey [2] to confirm this intuition. Notice that we currently disallow changing this parameter in the demo since assigning equal weight almost never improves the user experience.

3.5 Enhancing Results using Ontology Graph

In order to enable users to exploit the domain knowledge related to a given query, we integrate a domain ontology to the ObjectRank system.

We first build the *ontology graph* $G_O(V_O, E_O)$, a labeled directed graph that captures a domain knowledge for terms. A *term* consists of one or more keywords and generally it represents a subject in a specific domain such as 'Concurrency Control' in database literature. We create a node v for every term identified. An edge $e = (v \rightarrow u)$ is added if there is a semantic relationship between terms v and u. The edge is annotated with the type of the relationship and a weight w $(0 < w \leq 1)$ which denotes the strength of the relationship. So far, we only consider the relationship type 'is-a'. To provide the ontology graph of subjects in computer science area, we use a subset of the ACM Computing Classification System[4].

First, we compute related terms by running the ObjectRank algorithm on the ontology graph in the same way that we used the ObjectRank algorithm on the publications data graph to compute relevance values between a query and publications. Then, we calculate a new rank value of a publication p on a term t by combining the ObjectRank values of p on terms related to t. For example, when we run the ObjectRank algorithm on the ontology graph with *"Transaction Management"* node as a base set, terms such as *"Concurrency Control"* and *"Crash Recovery"* would get very high authority values. Using the new ranking function, which combines rank values of terms relevant to *"Transaction Management"*, publications relevant to *"Concurrency Control"* or *"Crash Recovery"* are favored even though their ObjectRank values on the given query are not high. In this way, the system can enhance search results automatically under the guidance of the ontology graph.

As another example consider the query *"Transaction Management Locking"*. If the system infers from the ontology graph that *"Locking"* is more closely related to *"Concurrency Control"* than to *"Transaction Management"*, and *"Transaction Management"* is highly relevant to *"Concurrency Control"*, the system will generate results that are similar to the results obtained by the ObjectRank algorithm with the query, *"Concurrency Control"* and *"Locking"*, which is desirable.

4. CONCLUSION

We presented the ObjectRank system that performs authority-based keyword search on bibliographic databases. We used Inverse ObjectRank as a keyword-specific specificity metric and other calibration parameters such as Global ObjectRank. Finally, we proposed a methodology that enables us to enhance the query results using an ontology graph.

5. REFERENCES

[1] A. Balmin, V. Hristidis, and Y. Papakonstantinou. ObjectRank: Authority-Based Keyword Search in Databases. *VLDB*, 2004.

[2] A. Balmin, V. Hristidis, and Y. Papakonstantinou. ObjectRank: Authority-Based Keyword Search in Databases (extended version). *UCSD Technical Report*, 2004.

[3] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *WWW Conference*, 1998.

[4] V. Hristidis, H. Hwang, and Y. Papakonstantinou. Authority-Based Keyword Search in Databases. *under preparation for journal submission*, 2006.

[5] R. Ramakrishnan and J. Gehrke. *Database Management Systems. Third Edition*. McGraw-Hill Book Co, 2003.

[4] http://www.acm.org/class/