DBPubs: Multidimensional Exploration of Database Publications

Akanksha Baid[2]
baid@cs.wisc.edu

Andrey Balmin[1]
abalmin@us.ibm.com

Heasoo Hwang[3]
heasoo@cs.ucsd.edu

Erik Nijkamp[4]
erik.nijkamp@de.ibm.com

Jun Rao[1]
junrao@us.ibm.com

Berthold Reinwald[1]
reinwald@us.ibm.com

Alkis Simitsis[1]
asimits@us.ibm.com

Yannis Sismanis[1]
syannis@us.ibm.com

Frank van Ham[5]
fvanham@us.ibm.com

[1]IBM Almaden RC
San Jose, CA, USA

[2]Univ. Wisconsin
Madison, USA

[3]UC San Diego
CA, USA

[4]IBM Germany
Germany

[5]IBM Research
Cambridge, MA, USA

ABSTRACT

DBPubs is a system for effectively analyzing and exploring the content of database publications by combining keyword search with OLAP-style aggregations, navigation, and reporting. DBPubs starts with keyword search over the content of publications. The publications' metadata such as title, authors, venues, year, and so on, provide traditional OLAP static dimensions, which are combined with dynamic dimensions discovered from the content of the publications in the search result, such as frequent phrases, relevant phrases, and topics. We compute publication ranks based on the link structure between documents, i.e., citations, and aggregate them to find seminal papers, discover trends, and rank authors. We deploy an OLAP tool for multidimensional content exploration through traditional OLAP rollup-drilldown operations on the static and dynamic dimensions, solutions for multi-cube analysis, dynamic navigation of the content, and highlighting of interesting dices of the multidimensional content dataspace.

1. MOTIVATION

`DBpubs` is a prototype system that integrates keyword search over a document corpus with OLAP-style aggregations in a multidimensional structure to better analyze and visualize large amounts of data. The multidimensional structure includes the document metadata as traditional OLAP static dimensions that are combined with dynamic dimensions discovered from the analyzed keyword search result as well as measures for document scores based on the link structure between documents.

Assume the following motivating example. Given the corpus of thousands of publications, we are interested in finding the various research topics that have been investigated in the XML domain in the past years. Traditional keyword search is used to retrieve all publications containing the term "XML" in the text or metadata. The query result comprises thousands of papers, and it requires further browsing of each of the resulting publications for content analysis. We dynamically group all the papers about XML into meaningful automatically discovered topics such as 'XQuery', 'Schema Matching', 'XPath Views', and so on. Given the fact that a paper can be characterized by more than one topic, the procedure we follow significantly differs from traditional clustering techniques. Next, we combine the topics with other metadata such as time, location, authors, and we dynamically put them in a multidimensional structure. Such structure provides means for multidimensional content exploration through traditional OLAP rollup-drilldown operations on both, the static and dynamic dimensions.

Several research efforts focused on tools that facilitate search on bibliographic data: DBLife [6] is a portal for the database community, that focuses on high accuracy extraction/integration operators. Similarly, Eventseer [9], Faceted DBLP [8], Publish or Perish [10] are websites that manage metadata for publications, but they do not consider the content –publication text– itself. `DBpubs` provides more than faceted search. It takes a different approach as it combines existing multidimensional OLAP processing with keyword search, link analysis, and dynamic document analysis, to provide a tool to explore content in a multidimensional structure.

2. DATA REPOSITORY

`DBpubs` has a repository that stores the content - unstructured textual data - of database publications along with its respective meta data. Currently, we store over 30,000 database publications from the field of databases and other adjacent areas, including major conferences and journals like SIGMOD, VLDB, ICDE, TODS, VLDB Journal, IEEE TKDE, PODS, STOC, SIGIR, and so on. We incorporate also the associated metadata from the DBLP entries [7], citations from CiteSeer [5], and geographic information from the Mondial database [14]. Figure 1 depicts a high-level view of the data repository along with a flow covering the different phases of our system, from the data integration of the data sources to the querying application existing on top of the repository.

Data Integration. The bottom part of Figure 1 shows the data collection part of the system. Our corpus contains publications gathered from the SIGMOD Anthology Collection [16] as well as downloaded from the Web. We converted the PDF files to text, and created a text index on both the text and on metadata fields such as *title*, *authors*, *venue*, *year*, *location* of the conference, and so on.

PVLDB '08, August 23-28, 2008, Auckland, New Zealand

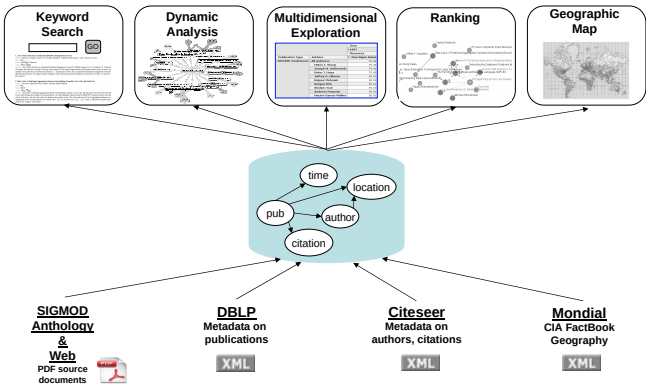

Figure 1: DBPubs Data Repository

The publication metadata is queried from an XML file provided by DBLP. The association between the publications downloaded from the Web and the DBLP entries is established through similarity tests on n-grams of the DBLP title and authors.

In our approach, we use citations to build a reference graph for publications citing each other. Also, we use link analysis over the graph for ranking the publications. The citation information was found by two methods. First, we extracted several citations from the Citeseer dataset. Missing citations were added through searches over the publication text as follows. Instead of extracting citations from the publication text to find out which publication cites another publication, we find out which publication is cited by another publication. Specifically, we create a spanning keyword query using DBLP title, year, and author of a publication. The spanning query consists of several constraints, such as (a) the author name and the title should appear in order and within a fixed number of words of each other, (b) the year can be either a four or two digit word, and can be located after the author name and either before or after the title, and so on. This way, we collected approx 210,000 citations.

Also, we have included geographic information in our repository. We used the Mondial dataset to obtain the geographic information (latitude, longitude) where a conference took place. The association between conference location and the geography information is implemented through approximate string match (q-gram based) using the DBLP proceeding titles. An inverted file contains all the Mondial geography information. The DBLP proceeding titles form a keyword query against the inverted file, and the search gives a ranked list of matches. We pick the top match of the list. Overall the whole data integration process involved very challenging and time-consuming data cleaning processes.

Repository. The data repository in Figure 1 contains *pubs* including the entire publication text and attributes such as *time*, a multivalued attribute *author*, *location*, *title*, a multivalued attribute *citation*, and so forth. *Time* is connected to a time dimension with different granularity for years and decades. *Location* is connected to a location dimension with cities, states/provinces, countries, and continents. *Author* is connected to an author dimension with affiliation information, address, etc. Currently, the affiliation information is provided either by DBLP or Citeseer, although we plan to extract such information from the web (homepages) as well. *Citation* information is used to build the reference graph for ranking.

3. DEMONSTRATION

The upper part of Figure 1 demonstrates the usage of `DBpubs`.

		Time	
		All Times	
		Measures	
Discovered Topic	Publication	Num Papers	Max Importance
All Discovered Topics	All Publications	1,156	69.09
(Other)	All Publications	578	68.44
Data Transformations	All Publications	43	66.53
Filtering of Xml Documents	All Publications	32	58.42
Generation Synthetic	All Publications	23	34.07
Keys Ande Foreign Keys	All Publications	23	22.22
Labeled Dynamic Xml Tree	All Publications	33	26.96
Path Index	All Publications	132	60.90
Query on Compressed Xml	All Publications	18	28.34
Schema Mapping	All Publications	73	45.60
Selectivity Estimation for Xml Twig	All Publications	55	63.33
Validation against DTDs	All Publications	66	69.09
Views Query	All Publications	84	40.25
Web Services	All Publications	53	33.68
Xpath Query on Streaming	All Publications	111	38.31
Xquery on SQL Host	All Publications	71	27.92

Figure 2: Query: abstract:xml, Multidimensional navigation

The user starts with keyword search (1). The query result is dynamically analyzed (2) and enriched with additional dimensional data and measures for ranks (2). Next, it is formed into a multidimensional structure for OLAP (3). Contextual search visualizes the Top-k nodes of the reference graph based on ObjectRank[3], and allows for further document retrieval (4). The GeoMap (5) maps the publications to the location where the paper was presented.

Keyword Search (1). The user first performs a keyword query –we support Lucene's query language– that may include fielded search on the static dimensions. Static dimensions are author, title, venue, year, content, and abstract (including title, authors). For example, the query "abstract:xml" searches for all publications that contain the term "xml" in the abstract field. The keyword query result shows 1156 hits. The results are ranked by relevance. The system displays per hit the title, authors, year, venue, and head of the document. The PDF document can be retrieved from the repository as well.

Dynamic Analysis and Ranking (2). For the interested user to get familiar with "XML", it is not feasible to exhaustively go through 1156 hits and read the papers. Hence, the system dynamically analyzes the document content to discover topics from the documents in the result set. Topic discovery deploys Singular Value Decomposition (SVD) on the relevant phrase-document matrix, which is heuristically pruned for scalability reasons [17]. A document can occur under multiple topics and each topic can in turn contain multiple documents. Documents that cannot be assigned to topics fall under the "Other" topic. The discovered topics are dynamically added as additional attributes to the documents in the hit list. For the keyword query "XML", the system discovers topics such as "Data Transformations", "Path Index", "Schema Mapping", "Validation against DTDs", "Web Services", "Xpath Query on Streaming", and so on.

Furthermore, given the reference graph based on citations, we use the hits in the query result as starting points for the random surfers and run the ObjectRank algorithm [3] on the reference graph to compute a dynamic rank for the publications. The ObjectRank is an adaptation of Personalized PageRank [11] to graphs with typed edges. It has been shown to work well in bibliographic datasets, where different types of relationships, e.g. citation, co-authorship, same-conference, all affect the ranking of the objects albeit to a different degree. We also compute a static page rank [4].

We also apply efficient algorithms to dynamically compute the frequent and relevant phrases, and add them as multi-valued attributes to the documents in the hit list.

Multidimensional Exploration and Reporting (3). The results of the keyword search augmented with dynamic analysis are consumed by an OLAP tool. The OLAP tool allows to explore the

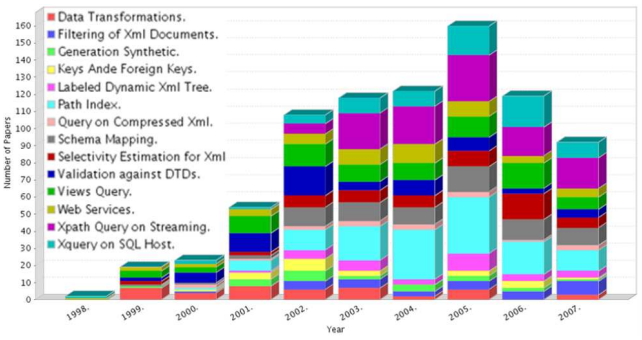

Figure 3: Query: abstract:xml, Discovered topics over years

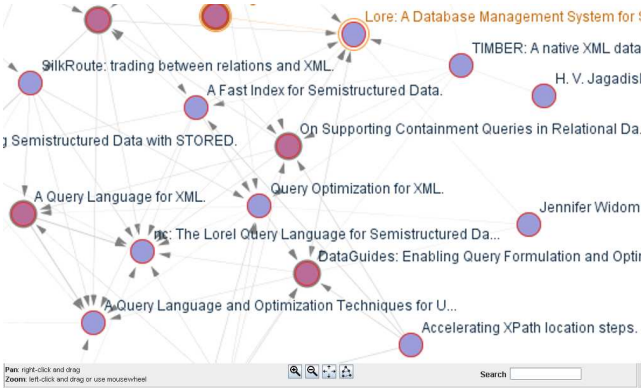

Figure 4: Query: abstract:xml, Context graph view

result from different dimensions, perform OLAP aggregations, roll-up and drill downs, and so on. The OLAP tool uses a Pivot table to display the results. Figure 2 shows on the row axis the topics discovered from keyword query "XML", and publications, and time dimension on the column axis. Measures are the count of publications, and the maximum static page rank. The displayed Pivot table can be explored interactively.

Several interesting reports can easily be created by the OLAP tool. Figure 3 shows the evolution of discovered topics for the XML keyword query in the course of time. Each topic can in turn be drilled into to reveal the documents that fall under it and their respective characteristics like authors, publication year, conference name, and so forth.

In addition to static reports, we analyze the multidimensional structure and highlight dynamically a small number of dimensions that are deemed most "interesting" to a user. We define interestingness as how surprising an aggregated value of the search result along a dimension is from a certain expectation. The expectation is set based on the distribution of all publications in the repository, and the degree of interestingness is quantified as a probability that the actual value should occur from a random sample of the same size as the search result (the smaller that probability, the more interesting a dimension is.) A corresponding MDX query is generated over the most interesting dimensions for more focused exploration by the user.

Contextual Search (4). Contextual Search provides a graph-oriented view of the dataspace. We build an object graph that contains authors and publications as nodes and relationships such as writes, cites, cited as edges. We dynamically compute the ObjectRank score for authors and publications using the keyword query result as a base set. The displayed graph shows the top-k, e.g., k=30, highest ranked objects, i.e., publications and authors (Figure 4.)

Geo Map (5). The GeoMap is a simple Mashup that takes the publications in the keyword query hit list and displays them on a map using the latitude/longitude (from Mondial) of the conference locations, where a paper was presented. Our future plans include to extend this functionality to authors' affiliation (e.g., show the places on earth that conduct research on 'XML'.)

4. QUERIES AND REPORTS

In this section, we demonstrate several interesting queries and reports that our system supports.

Test of Time Query. Venues such as SIGMOD recognize the best paper from the SIGMOD proceedings 10 years prior based on the most impact. In DBPubs, we rank the publications based on the static PageRank in the reference graph. With the PageRank measure, the keyword query *"+venue:SIGMOD +year:1997"* returns the top papers published in SIGMOD 1997 (see Figure 5.) The paper by Hellerstein, Haas, and Wang on Online Aggregation has the highest rank, and actually received the 2007 SIGMOD Test of Time award. The system also shows, besides many other features, that "Web Browsers", "Document Stores", "OLAP Data Cube", "Data Mining", and "Index and Retrieval" were the top 5 most important topics in 1997.

Most Important Topics. A user may want to find the most important topics in the field of 'XML'. One way to rank the topics is to sum the scores of their publications. The top 10 most important topics is a simple OLAP query that groups by topics and sums the scores of the publications (Figure 6.) Similarly, the top 10 most important authors can be computed by aggregating the scores of the publications per author.

Seminal papers. Figure 7 shows the seminal papers (highest ranked papers) in a sliding window of 2 years.

5. ARCHITECTURE

Figure 8 shows the system architecture of `DBpubs`: The `DBpubs` repository is implemented in DB2 V9.1, heavily exploiting the native XML support particularly in dealing with multi-valued attributes (e.g. authors, topics) as well as implementing the data integration for the XML data sources DBLP, Citeseer, and Mondial. Metadata, PDF documents, and citations are stored in DB2 tables. The open source tool PDFBox[15] is deployed to convert the PDF documents to text. The text is indexed by Lucene. Lucene is integrated into DB2 through User-Defined Functions (UDFs). We implemented UDFs to populate the index and to submit keyword queries.

Static PageRank is implemented as a Java UDF. We use a main-memory sparse matrix representation to represent citation information. The PageRank computation is executed by computing the principal eigenvector of the sparse matrix using the power-law technique. The technique is iterative and requires less than fifteen iterations to converge. For our corpus size the computation takes a few seconds to complete.

ObjectRank is implemented as a Java Webservice. It is invoked once a Lucene query result is generated. The list of documents returned by Lucene is used as the ObjectRank's *baseset*, i.e., the

		Measures	
Publication	Author	Num Cited	Max Importance
All Publications	All Authors	1,094	107.91
Online Aggregation.	All Authors	118	107.91
	Helen J. Wang	118	107.91
	Joseph M. Hellerstein	118	107.91
	Peter J. Haas	118	107.91
An Overview of Data Warehous	All Authors	81	100.42
Lore: A Database Management	All Authors	58	73.58

Figure 5: Query: "+venue:SIGMOD +year:1997", Test of time

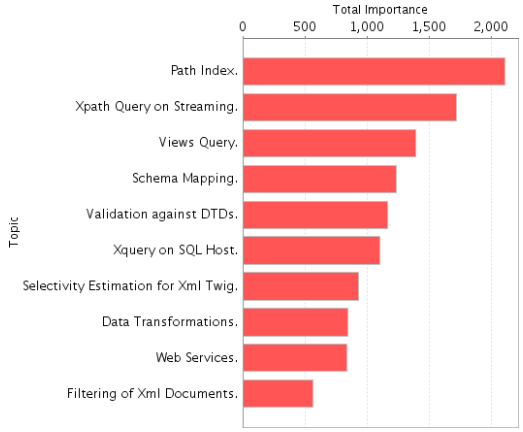

Figure 6: Query: abstract:xml, Most important topics

starting points of random walks. ObjectRank keeps in memory the static graph that contains publication and author nodes as well as the relationship edges.

Frequent phrases are phrases consisting of up to five words that occur across more than five documents. We experimented with different thresholds without any major difference in the results. Interesting (a.k.a relevant) phrases with respect to a keyword query are frequent phrases scaled by the corresponding TFIDF scores. For example, for the keyword query OLAP, phrases like "data cube" are scaled up a lot, since they appear often in the result set and relatively infrequent in the full corpus. Topic discovery builds on relevant phrases using a relevant phrase-document matrix applying an approach inspired by [12].

Frequent and relevant phrases as well as topics are added as multi-valued attributes to a dynamic fact table. Multi-valued attributes are implemented through XML.

We use Mondrian([2]) as an OLAP server. Mondrian is implemented in Java and hosted in Tomcat. The data cube is described to Mondrian in an XML configuration file. This file describes all the constituents of the multidimensional schema, such as fact and dimension tables, levels, measures, and multi-cubes connecting different fact tables through appropriate join dimensions or bridge tables. The configuration file is generated dynamically once the keyword query is executed and the dynamic dimensions filled in. This functionality allows us to construct on-the-fly the suitable multidimensional schema w.r.t. the query issued.

Once the multidimensional schema has been created, the user can query it explicitly through the MDX language [13]. Mondrian

YEA	TITLE	AUTHORS
2004	Retrieval evaluation with incomplete information.	Chris Buckley; Ellen M. Voorhees
2002	Models and Issues in Data Stream Systems.	Brian Babcock; Shivnath Babu; Mayur Datar; Rajeev
1998	The Anatomy of a Large-Scale Hypertextual Wel	Sergey Brin; Lawrence Page
1996	Data Cube: A Relational Aggregation Operator G	Jim Gray; Adam Bosworth; Andrew Layman; Hamid P
1993	Mining Association Rules between Sets of Items	Rakesh Agrawal; Tomasz Imielinski; Arun N. Swami
1992	An Interval Classifier for Database Mining Applic	Rakesh Agrawal; Sakti P. Ghosh; Tomasz Imielinski;
1990	The R*-Tree: An Efficient and Robust Access M	Norbert Beckmann; Hans-Peter Kriegel; Ralf Schneic
1987	The R+-Tree: A Dynamic Index for Multi-Dimensi	Timos K. Sellis; Nick Roussopoulos; Christos Faloutsı
1984	R-Trees: A Dynamic Index Structure for Spatial S	Antonin Guttman
1983	Benchmarking Database Systems A Systematic /	Dina Bitton; David J. DeWitt; Carolyn Turbyfill
1981	The Functional Data Model and the Data Langu	David W. Shipman
1979	Access Path Selection in a Relational Database	Patricia G. Selinger; Morton M. Astrahan; Donald D. I
1976	System R: Relational Approach to Database Ma	Morton M. Astrahan; Mike W. Blasgen; Donald D. Ch
1974	SEQUEL: A Structured English Query Language	Donald D. Chamberlin; Raymond F. Boyce
1973	Optimum Data Base Reorganization Points.	Ben Shneiderman
1972	Theoretical Improvements in Algorithmic Efficien	Jack Edmonds; Richard M. Karp
1970	A Relational Model of Data for Large Shared Dal	E. F. Codd

Figure 7: Query: Seminal Papers

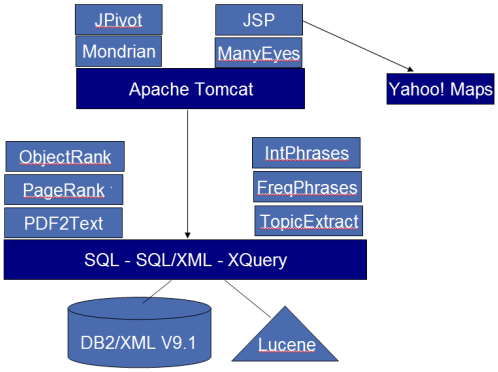

Figure 8: `DBpubs` System Architecture

parses MDX into SQL for DB2. Alternatively, the user can use the GUI to perform typical OLAP navigations like slicing, dicing, drill downs, and so on, without acquiring any specific knowledge about MDX or any other programming language. Such navigational operations are translated internally into MDX by JPivot. JPivot[1] is a JSP tag library that renders Pivot tables and uses XMLA to communicate with Mondrian.

Our graph visualization deploys a component from ManyEyes [18]. The GeoMap uses Yahoo! Map.

6. CONCLUSIONS

The framework of our approach is fully described in a paper accepted in VLDB'08 [17]. Currently, `DBpubs` is a working system available to the CS group at IBM's Almaden Research Center. During its operation, we have gathered several fruitful comments and feedback from our colleagues and due to them our system has been improved a lot.

7. REFERENCES

[1] JPivot. http://jpivot.sourceforge.net/.
[2] Mondrian, http://mondrian.pentaho.org.
[3] A. Balmin, V. Hristidis, and Y. Papakonstantinou. Objectrank: Authority-based keyword search in databases. In *VLDB*, pages 564–575, 2004.
[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.
[5] CiteSeer. http://citeseer.ist.psu.edu.
[6] DBLife. http://dblife.cs.wisc.edu.
[7] DBLP. http://www.informatik.uni-trier.de/ ley/db.
[8] J. Diederich. Faceted DBLP, http://dblp.l3s.de.
[9] Eventseer. http://eventseer.net.
[10] Harzing. Publish or Perish, http://www.harzing.com/pop.htm.
[11] G. Jeh and J. Widom. Scaling personalized web search. In *WWW*, 2003.
[12] S. Law, O. Jerzy, and S. Dawid. Lingo: Search results clustering algorithm based on singular value decomposition, 2004.
[13] MDX. http://en.wikipedia.org/wiki/multidimensional_expressions.
[14] Mondial. http://www.dbis.informatik.uni-goettingen.de/mondial.
[15] PDFBox.org. PDFBox, http://www.pdfbox.org/.
[16] SIGMOD. SIGMOD Anthology, http://www.sigmod.org/sigmod/anthology/index.htm.
[17] A. Simitsis, A. Baid, Y. Sismanis, and B. Reinwald. Multidimensional content exploration. In *VLDB*, 2008.
[18] F. B. Viégas, M. Wattenberg, F. van Ham, J. Kriss, and M. M. McKeon. Manyeyes: a site for visualization at internet scale. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1121–1128, 2007.