Demonstrating Effective Ranked XML Keyword Search with Meaningful Result Display

Zhifeng Bao[1], Bo Chen[1], Tok Wang Ling[1], and Jiaheng Lu[2]

[1] School of Computing, National University of Singapore
{baozhife, chenbo, lingtw}@comp.nus.edu.sg
[2] School of Information and DEKE, MOE, Renmin University of China,
jiahenglu@gmail.com

Abstract. In this paper, we demonstrate an effective ranked XML keyword search with meaningful result display. Our system, named ICRA, recognizes a set of object classes in XML data for result display, defines the matching semantics that meet user's search needs more precisely, captures the ID references in XML data to find more relevant results, and adopts novel ranking schemes. ICRA achieves both high result quality and high query flexibility in search and browsing. An online demo for DBLP data is available at **http://xmldb.ddns.comp.nus.edu.sg/**.

1 Introduction

The goal of XML keyword search is to find only the meaningful and relevant data fragments corresponding to interested objects that users really concern on. Most previous efforts are built on either the tree model or the digraph model. In tree model, Smallest Lowest Common Ancestor (SLCA) [5] is an effective semantics. However, it cannot capture the ID references in XML data which reflect the relevance among objects, while digraph model can. In digraph model, a widely adopted semantics is to find the *reduced subtrees* (i.e. the smallest subtrees in a graph containing all keywords). However, enumerating results by increasing the sizes of reduced subtrees is a NP-hard problem, leading to intrinsically expensive solutions. Moreover, it neither distinguishes the containment and reference edge in XML data, nor utilizes the database schema in defining matching semantics.

Besides, we observe that existing approaches in both models have two common problems. *First*, regarding to the design of matching semantics, they fail to effectively identify an appropriate information unit for result display to users. Neither SLCA (and its variants) nor reduced subtree is an appropriate choice, as neither of them is able to capture user's search target, as shown in Example 1. *Second*, the existing ranking strategies in both models are built at XML node level, which cannot meet user's search concern more precisely at object level.

Example 1. Query 1: "Suciu" is issued on the XML data in Fig. 1, intending to find papers written by "Suciu". Both SLCA and reduced subtree return the author nodes with value "Suciu", which is not informative enough to user.
Query 2: "Suciu XML" is issued on Fig. 1 to find XML papers written by Suciu. As there is no Suciu's paper containing "XML", the SLCA result is the whole subtree under the root node, which contains too much irrelevant information.□

Generally speaking, the technical challenges of this demo lie in as below:
(1) The results need to be semantically meaningful to the user to precisely meet user's search needs, and meanwhile avoid overwhelming the user with a huge number of trivial results. However, methods on graph model suffer from producing large number of trees containing the same pieces of information many times.

(2) How to define appropriate matching semantics to find more relevant results by capturing ID references in XML data while optimizing the search efficiency.
(3) How to design a general-purpose and effective ranking scheme.

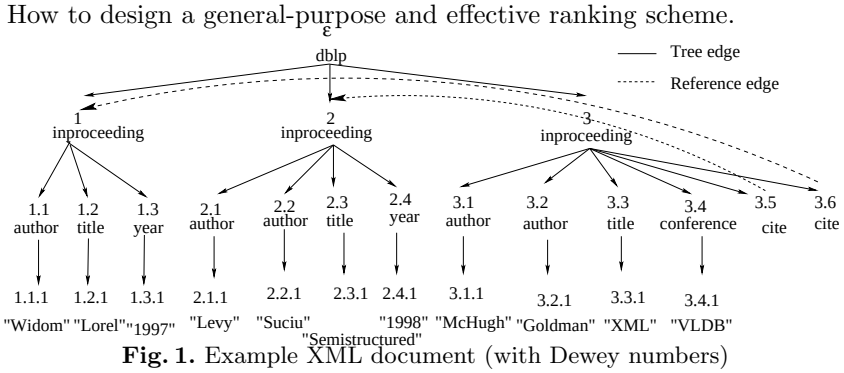

Fig. 1. Example XML document (with Dewey numbers)

To address these challenges, we present an XML keyword search system ICRA [1]. In particular, by modeling XML data as a set of *interconnected object-trees*, ICRA first automatically recognizes a set of objects of interest and the connections between them. Meanwhile, object trees as query results contain enough but non-overwhelming information to represent a real world entity, so the problem of proper result display is solved. To capture user's search concern on a single object, *ICA* is proposed; to capture user's search concern on multiple objects, *IRA pair (group)* is proposed to find a pair (group) of object trees that are related via direct or indirect citation/reference relationships and together contain all keywords. i.e. IRA helps find more relevant results. For Query 1 in Example 1, ICA returns inproceeding:2 rather than its *author* subelement, which is both informative and relevant. For Query 2, ICA cannot find any qualified single inproceeding, while IRA finds a pair of inproceedings (inproceeding:2, inproceeding:3), where inproceeding:2 written by "Suciu" is cited by inproceeding:3 containing "XML".

Compared with prior search systems, ICRA has significant features.
(1) The interconnected object-trees model guarantees meaningful result display. Compared to tree model, it can capture ID references to find more relevant results; compared to digraph model, it achieves more efficient query evaluation.
(2) It takes advantage of the schema knowledge to define the matching semantics which is of same granularity as user's search needs, and facilitate the result display and performance optimization in terms of result quality and efficiency.
(3) It designs a novel relevance oriented ranking scheme at object-tree level, which takes both the internal structure and content of the results into account.

2 Ranking Techniques

As ICA and IRA correspond to different user search needs, different ranking schemes are designed. To rank the ICA results, traditional TF*IDF similarity is extended to measure the similarity of an object tree's content w.r.t. the query. Besides considering the content, the structural information within the results are also considered: (1)**Weight of matching elements in object tree**. (2)**Pattern of keyword co-occurrence**. Intuitively, an object tree o is ranked higher if a nested element in o directly contains all query keywords, as they co-occur closely. (3)**Specificity of matching element**. Intuitively, an object

tree o is ranked higher if an element nested in o exactly contains (all or some of) the keywords in Q, as o fully specifies Q. E.g. for query "Won Kim", Won Kim's publications should be ranked before Dae-Won Kim's publications. To rank the IRA results, we combine the self similarity of an IRA object o and the bonus score contributed from its IRA counterparts. Please refer to [1] for more details.

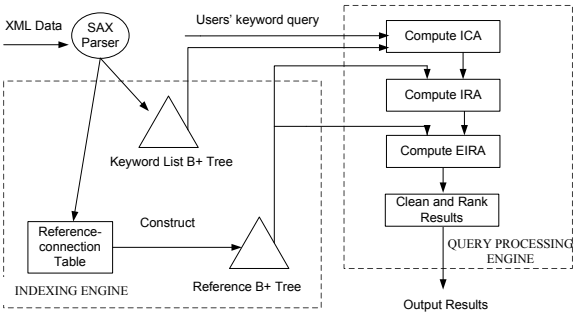

Fig. 2. System architecture

3 System Architecture

System architecture is shown in Fig. 2. During data preprocessing, the *Indexing Engine* parses the XML data, identifies the object trees, and builds the keyword inverted list storing for each keyword k a list of object trees containing k; it also captures ID references in XML data and stores them into *reference connection table*. A B+ tree is built on top of these two indices respectively. During the query processing stage, it retrieves the object trees containing the specified keywords to compute ICA results; then it computes IRA and Extended IRA (EIRA) results with the help of reference connection table. Lastly, it cleans and ranks the results.

4 Overview of Online Demo Features

ICRA provides a concise interface, user can explicitly specify their search concern - *publications* (default) or *authors*. ICRA offers various query flexibility: users can issue pure keyword queries that can be any combinations of words in full or partial specification of author names, topics, conference/journal and/or year.

4.1 Search for publications

Users can search for publications with various types of queries as below. Readers are encouraged to try more query types at ***http://xmldb.ddns.comp.nus.edu.sg/***.

Author name. - E.g. we can query "Jiawei Han" for his publications. ICRA will rank Jiawei Han's papers higher than papers co-authored by Jiawei and Han.

Multiple author names. - to search for co-authored papers.

Topic. - E.g. we can query "xml query processing".

Topic by an author. - E.g. we can query "Jim Gray transaction" for his publications related to transaction. Jim Gray's papers containing "transaction" are ranked before his papers citing or cited by "transaction" papers.

Topic of a year. - E.g. we can query "keyword search 2006".

Conference and author. - E.g. we can query "VLDB Raghu Ramakrishnan".

4.2 Search for authors

Users can also search for authors with various types of queries as below.

Author name. - By typing an author name, ICRA returns this author followed by a ranked list of all his/her co-authors (e.g. try "Tova Milo").

Topic. - We can search for authors who have the most contributions to a research topic (e.g. try "XML keyword search").
Conference/Journal name. - We can find active authors in a particular conference or journal (e.g. try "DASFAA").
Author name and topic/year/conference/journal. - Besides the author himself/herself, we can also search for his/her co-authors in a particular topic or year or conference/journal (e.g. we can search for Ling Tok Wang and his co-authors in DASFAA 2006 with a query "Ling Tok Wang DASFAA 2006").

4.3 Browsing
Besides searching, ICRA also supports browsing from search results to improve its practical usability. E.g. users can click an author (or conference/journal) name in a result to see all publications of this author (or the proceeding/journal). Link is provided to find the references and citations of a paper. When searching for authors, we output both the number of publications containing all keywords and the number of publications that may be relevant via the reference connections.

4.4 Result Display
ICRA displays the result for ICA and IRA semantics *separately* in "AND" and "OR" part. In addition, since a same paper may appear in more than one IRA pair/group, it will annoy the user in result consumption if such paper appears many times. Therefore, ICRA only outputs one IRA object o for each IRA pair/group, and provides links to the objects that form IRA pair/group with o.

5 Effectiveness of ICRA
In the demo, we will compare the result quality of ICRA with typical academic demo systems for DBLP, such as BANKS [4], ObjectRank [3] and FacetedDBLP [2]. We will also compare ICRA with commercial systems such as Microsoft Libra and Google Scholar[†]. ICRA has a good overall performance in terms of both result quality and query response time, as evidenced by experiments in [1].

6 Feature Comparisons
A comparison of the features in existing demos is given from a user's perspective. BANKS produces results in form of reduced trees which is difficult for novice users to consume. Query types supported by ObjectRank and FacetedDBLP are not as flexible as ICRA. E.g. they cannot handle searching papers of co-authors, or a topic by author etc. ObjectRank doesn't support browsing. DBLP CompleteSearch doesn't employ any relevance oriented ranking functions.

References

1. Z. Bao, B. Chen, and T. W. Ling. Effective ranked XML keyword search with meaningful result display. http://xmldb.ddns.comp.nus.edu.sg/Demo.pdf.
2. J. Diederich, W.-T. Balke, and U. Thaden. Demonstrating the semantic growbag: automatically creating topic facets for faceteddblp. In *JCDL*, 2007.
3. H. Hwang, V. Hristidis, and Y. Papakonstantinou. Objectrank: a system for authority-based search on databases. In *VLDB*, 2006.
4. V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, and R. Desai. Bidirectional expansion for keyword search on graph databases. In *VLDB*, 2005.
5. Y. Xu and Y. Papakonstantinou. Efficient keyword search for smallest LCAs in XML databases. In *SIGMOD*, 2005.

[†] Libra: http://libra.msra.cn/ Google Scholar: http://scholar.google.com/