Data & Knowledge Engineering 99 (2015) 105–125

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Exploiting semantics for XML keyword search

Thuy Ngoc Le [a,*], Zhifeng Bao [b], Tok Wang Ling [a]

[a] National University of Singapore, Singapore
[b] RMIT University, Australia

ARTICLE INFO

Article history:
Received 23 February 2015
Accepted 9 June 2015
Available online 18 June 2015

Keywords:
XML
Keyword search
Object
LCA
Independence
Semantics

ABSTRACT

XML keyword search has attracted a lot of interests with typical search based on lowest common ancestor (LCA). However, in this paper, we show several problems of the LCA-based approaches, including meaningless answers, incomplete answers, duplicated answers, missing answers, and schema-dependent answers. To handle these problems, we exploit the semantics of object, object identifier, relationship, and attribute (referred to as the ORA-semantics). Based on the ORA-semantics, we introduce new ways of labeling and matching. More importantly, we propose a new semantics, called CR (*C*ommon *R*elative) for XML keyword search, which can return answers independent from schema designs. To find answers based on the CR semantics, we discover properties of common relative and propose an efficient algorithms. Experimental results show the seriousness of the problems of the LCA-based approaches. They also show that the CR semantics possesses the properties of completeness, soundness and independence while the response time of our approach is faster than the LCA-based approaches thanks to our techniques.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Since XML has become a standard for information exchange over the Internet, more and more data are represented as XML. Therefore, XML has wide applications such as electronic business,[1] science,[2] text databases,[3] digital libraries,[4] healthcare,[5] finance,[6] and even in the cloud [3]. As a result, XML has attracted a huge of interests in both research and industry with a wide range of topics such as XML storage, twig pattern query processing, query optimization, XML view, and XML keyword search. There have been several XML database systems such as Timber [10], Oracle XML DB,[7] MarkLogic Server,[8] and the Toronto XML Engine.[9] XML keyword search has also been studied extensively based on lowest common ancestors such as SLCA [24], VLCA [16], MLCA [19] and ELCA [26].

Keyword search is a user-friendly way so that users can issue keyword queries without or with little knowledge about the schema of the underlying data. However, they often know what the data is about. Therefore, when they issue a query, they often have some expectations about the answers in mind. Since they may not know which schema is being used, their expectations are independent from schema designs. If they already got some answers for this schema, it could be surprised if different answers are returned

* Corresponding author at: 13 Computing Drive, Singapore 117417, Republic of Singapore.
E-mail addresses: ltngoc@u.nus.edu (T.N. Le), zhifeng.bao@rmit.edu.au (Z. Bao), lingtw@comp.nus.edu.sg (T.W. Ling).

[1] http://www.ebxml.org.
[2] http://www.biodas.org/documents/spec-1.53.html.
[3] http://www-connex.lip6.fr/~denoyer/wikipediaXML/.
[4] http://www.loc.gov/standards/mods/presentations/mets-mods-morgan-ala07/.
[5] http://www.ncbi.nlm.nih.gov/pubmed/11066651.
[6] http://schemas.liquid-technologies.com/Category/Financial.
[7] http://www.oracle.com/technetwork/database-features/xmldb/overview/index.html.
[8] http://www.marklogic.com/.
[9] http://www.cs.toronto.edu/tox/.

http://dx.doi.org/10.1016/j.datak.2015.06.003
0169-023X/© 2015 Elsevier B.V. All rights reserved.

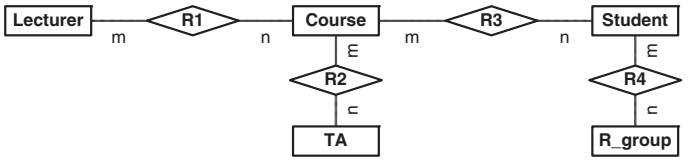

Fig. 1. ER diagram of a database.

when they try another schema which represents the same data content. Thus, different schemas of the same data content should provide them the same answers. However, this is not the case for the existing LCA-based approaches as shown in Example 1.

Consider the database with the ER diagram in Fig. 1 as our running database. There are many ways to represent this database in XML. Fig. 2 shows five possible XML schema designs for this database. For simplicity, we do not show attributes and values in these schemas. Each edge in the schemas corresponds to a *many-to-many* relationship types between the two object classes.

Example 1. Schema dependence

Users may know a university database about courses, lecturers, teaching assistants (TAs), students, and research groups (R_group),[10] *but they do not know what the schema looks like, i.e., which of the five schema designs in Fig. 2 is used. When they ask for two students (e.g., Q = {StudentA, StudentB}), beside information about the two students, they may want to know some of the below*:

- *Ans1: the common courses that they both take,*
- *Ans2: the common research groups (R_groups) that they both belong to,*
- *Ans3: the common lecturers who teach both of them,*
- *Ans4: the common teaching assistants (TAs) who teach and mark both of them.*

They are common ancestors in some schema(s): Ans1 in Schema 1, Schema 2 and Schema 3; Ans2 in Schema 5; Ans3 in Schema 2; and Ans4 in Schema 3. Therefore, they are all meaningful answers (probably with different ranking scores). Different users may have different expectations. However, expectations of a user should be independent from schema designs because he does not know which schema is used. However, all five different schema designs provide five different sets of answers by the LCA semantics. Particularly:

- *for Schema 1: only Ans1 could be returned;*
- *for Schema 2: Ans1 and Ans3 could be returned;*
- *for Schema 3: Ans1 and Ans4 could be returned;*
- *for Schema 4: no answer;*
- *for Schema 5: only Ans2 could be returned.*

The above example provides a strong evidence for our two following arguments:

Firstly, meaningful answers can be found beyond common ancestors because all kinds of answers Ans1, Ans2, Ans3 and Ans4 are meaningful. However, if relying only on the common ancestor techniques, none of the five schemas can provide all the above meaningful answers. For some schema, answers from common ancestors may be better than the others, but returning more meaningful answers would be better than missing meaningful ones.

A final answer obtained by LCA-based approaches includes two parts: a returned node (LCA node) and a presentation of the answer, e.g., a subtree or paths. Arguably, the presentation of an answer as a subtree may contain other answers. For instance, for Schema 1, the subtree rooted at the common courses (Ans1) that both students take may contain other kinds of answers (Ans2, Ans3, Ans4). However, the LCA-based approaches do not explicitly identify them and it may be hard for users to identify them because this presentation contains a great deal of irrelevant information. Thus, it is necessary to identify and separate them clearly.

Secondly, answers of XML keyword search should be independent from the schema designs, e.g., Ans1, Ans2, Ans3 and Ans4 should be returned regardless which schema is used to capture data. However, as can be seen, the LCA-based approaches return different answer sets for different schema designs in Fig. 2.

In practice, many real XML datasets have different schema designs such as IMDb[11] and NBA.[12] In IMDb, there are many ways to capture relationships among actors, actresses, movies, and companies. In NBA, relationships among coaches, teams, and players can also be captured in different ways. Moreover, due to the flexibility and exchangeability of XML, many relational datasets can be transformed to XML [12], and each relational database can correspond to several XML schemas by picking up different entities as the root for the resulting XML document.

Therefore, it necessitates to consider the above two arguments when processing XML keyword search. However, to the best of our knowledge, no current system satisfies the above two arguments, including keyword search over XML graph.

[10] R_group can be an object class with attributes: name, topics, leader, etc.
[11] http://www.imdb.com/interfaces.
[12] http://www.nba.com.

Download English Version:

https://daneshyari.com/en/article/378727

Download Persian Version:

https://daneshyari.com/article/378727

Daneshyari.com