Chapter 3

Ontology-Driven Keyword Search for Heterogeneous XML Data Sources

Weidong Yang
Fudan University, China

Hao Zhu
Fudan University, China

ABSTRACT

Massive heterogeneous XML data sources emerge on the Internet nowadays. These data sources are generally autonomous and provide search interfaces of XML query language such as XPath or XQuery. Accordingly, users need to learn complex syntaxes and know the schemas. Keyword Search is a user-friendly information discovery technique, which can assist users in obtaining useful information conveniently without knowing the schemas, and is very helpful to search heterogeneous XML data. In this chapter, the authors present a system called SKeyword which provides a common keyword search interface for heterogeneous XML data sources, and employs OWL ontology to represent the global model of various data sources. Section 1 introduces the context of keyword search for heterogeneous XML data source. In Section 2, the preliminary knowledge is given, and the semantics of keyword search result in ontology is defined. In section 3, the system architecture is described. Section 4 presents the approaches of ontology integration and index building used by SKeyword. Section 5 presents the generation algorithm of searching results and discusses how to rewrite the keyword search of global conceptual model to into the XQuery sentences for local XML sources. Section 6 discussed how to organize and rank the results. Section 7 shows the experiments. Section 8 is the related work. Section 9 is the conclusion of this chapter.

3.1 INTRODUCTION

More and more autonomous XML databases and systems emerge. Consider some data sources from a specific domain, which provide structural query interfaces and have heterogeneous schemas. The schemas mostly represent similar or even the same semantics. A reasonable requirement is requesting an identical query over them. However, distinct query interfaces and semantic conflicts bring difficulties. Moreover, the query results are heterogeneous and consequently cannot be sorted easily. In this chapter, we present a system called SKeyword which provide a common interface and
can meet the requirement. SKkeyword employs keyword search as the query model and builds a global ontology upon the multiple data sources to solve semantic conflicts, and finally uniform, sorts and returns the results to users.

**Example 1:** University A and university B both have a library management system with XML databases. Figure 1 illustrates the schemas of their XML databases. A has all data stored in one database, and B has two XML databases and hence has two corresponding schemas. In Figure 1 (b), the left schema is from the database which stores the information of books and employees, and the other one is from the database stores the information of students (assume only students borrow books). One thing should be noticed is: the dashed lines with an arc between them denote a “choice” relation. For example, in Figure 1 (b) each book could be written either by some authors or by an organization.

Suppose we have a common keyword search interface for two systems of A and B, and keywords “Yang Maths Differential Geometry” are submitted, then what kind of information should be returned?

Some reasonable search results could be deduced instantly, such as: “Is there a borrower whose name is Yang and now holds a Maths book about Differential Geometry?”, or “Is there a Maths book about Differential Geometry, and one of its authors is Yang?” etc. For former XML keyword search systems cannot find integrated information on multiple XML databases, users can never know which book is now in which person’s hand in university B (suppose Yang is a student of university B and he borrowed a Maths book about Differential Geometry). However even only one single database is considered (suppose the searching target is A), in former systems problems still exist: (1) they don’t know “Maths” means “Mathematics”, so none match of this term would be found; (2) although “a borrower Yang has borrowed a Maths book about Differential Geometry” is a very reasonable query, considering most researches are based on structural distance of nodes on tree or graph, such semantics would be the last thing be considered because the information of book’s name and borrower’s name are so far from each other structurally in the schema (no matter their LCA or distance is considered); (3) they return XML document segments or variations as the results, which mostly cannot express meaningful semantics (in some cases, only a value node is returned).

SKkeyword provides a solution to these problems. Suppose heterogeneous databases belong to multiple autonomous systems, and each system provides a search interface of XQuery. Firstly SKkeyword employs semiautomatic ontology integration to integrate heterogeneous schemas into a global ontology. Implicit semantics in a single schema or between schemas can be revealed and described in the global ontology (e.g. the relation between borrower and book in Figure 1). Then, an inverted index of all terms along with a term-concept mapping table are built and stored in the SKkeyword server (as illustrated in Figure 2). When keywords are submitted, with the term-concept mapping table SKkeyword classifies keywords into two kinds: concepts and values. A concept indicates a thing users are interested in, which is practically a class in the global ontology (e.g. library, book and borrower). After analysis of keywords, some concepts are figured out as targets. Then, some ontology segments called SCNs are generated according to the distance between concepts have keywords. Afterwards, these segments are translated to XQuery sentences which then are sent to autonomous systems. After receives all the query results, SKkeyword organizes them into ontologies. At last they are sorted and sent to users as final results.