OOXSearch: A Search Engine for Answering Loosely Structured XML Queries Using OO Programming

Kamal Taha and Ramez Elmasri

Department of Computer Science and Engineering
The University of Texas at Arlington, USA
kamal.taha@cse.uta.edu, elmasri@cse.uta.edu

Abstract. There has been extensive research in XML keyword-based and loosely structured querying. Some frameworks work well for certain types of XML data models and fail in others. The reason is that the proposed techniques are based on finding relationships between solely individual nodes while overlooking the context of these nodes. The context of a leaf node is determined by its parent node, because it specifies one of the characteristics of its parent node. Building relationships between individual leaf nodes without consideration of their parents may result in relationships that are semantically disconnected. Since leaf nodes are nothing but characteristics of their parents, we observe that we could treat each parent-children set of nodes as one unified entity. We then find semantic relationships between the different unified entities. Based on those observations, we propose an XML semantic search engine called OOXSearch, which answers loosely structured queries. The recall and precision of the engine were evaluated experimentally and compared with two recent proposed systems [1, 2] and the results showed marked improvement.

Keywords: Canonical Tree, Ontology Label, Relevant Canonical Tree, Search Term Context.

1 Introduction

The spectrum of users who interact with XML and their levels of skill have significantly widened due to the popularity and widespread use of XML. Since that spectrum includes naïve users, extensive research in XML keyword-based querying has been done. Even sophisticated users who are not aware of the XML document's schema may find keyword-based and loosely structured querying helpful. The studies that have been done could be categorized into four groups. The first expands structured query languages [15, 16]. The second uses keyword-based search techniques for ranking results based on importance and relevance [9,10, 24]. The key drawback of those ranking techniques is that they do not consider search semantics. The third employs search techniques based on semantic relationships between individual nodes [1, 2, 3]. The fourth proposes modeling the XML document as a graph and processing the graph based on driven schema

R. Cooper and J. Kennedy (Eds.): BNCOD 2007, LNCS 4587, pp. 82–100, 2007.

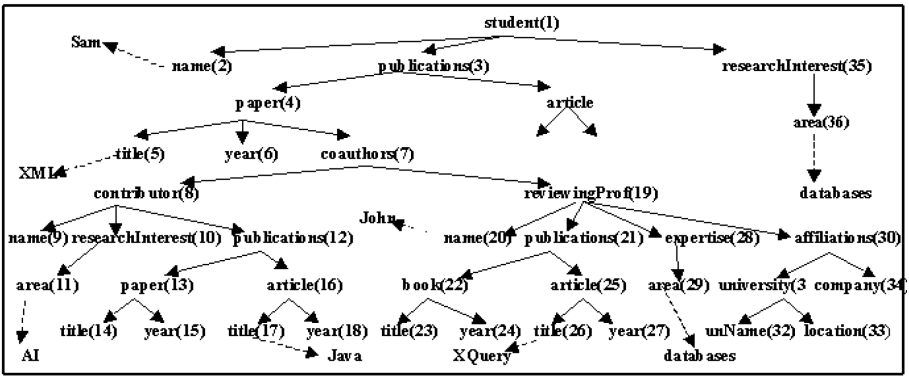

Fig. 1. A graduate school's authors and coauthors bibliography XML tree with sample data instances (student.xml doc)

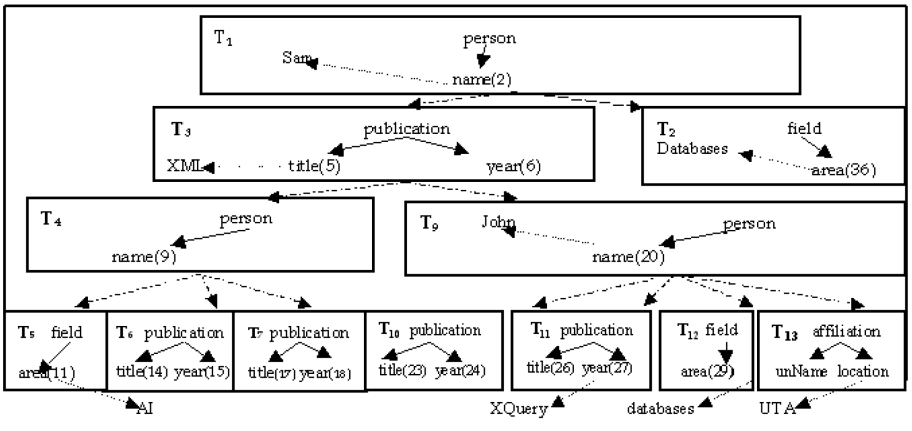

Fig. 2. Canonical Trees graph of the XML tree presented in Figure 1

[22, 23, 26]. While each of these proposed techniques has advantages, it also has drawbacks, such as returning redundant and/or wrong answers. The reason is that they propose frameworks based on finding relationships between individual nodes while overlooking the "context" of those nodes. The context of a leaf node is determined by its parent node because a leaf node is a characteristic of its parent. If for example a node is labeled "title", we cannot determine whether the node refers to a book title or a job title without referring to the parent of the node. The techniques proposed by [1] and [2] for example may return wrong answers as a result of not considering leaf nodes contexts. In [1], the authors propose that if the relationship tree that connects nodes a and b does not include two or more nodes that have the same label, then nodes a and b are related. In Figure 1, for example, the relationship tree of nodes 2 and 4 contains nodes 2,