Chapter 2
An Efficient and Flexible Approach of Keyword Search in XML

Weidong Yang
Fudan University, China

Hao Zhu
Fudan University, China

ABSTRACT
In this chapter, firstly, the LCA-based approaches for XML keyword search are analyzed and compared with each other. Several fundamental flaws of LCA-based models are explored, of which, the most important one is that the search results are eternally determined nonadjustable. Then, the chapter presents a system of adaptive keyword search in XML, called AdaptiveXKS, which employs a novel and flexible result model for avoiding these defects. Within the new model, a scoring function is presented to judge the quality of each result, and the considered metrics of evaluating results are weighted and can be updated as needed. Through the interface, the system administrator or the users can adjust some parameters according to their search intentions. One of three searching algorithms could also be chosen freely in order to catch specific querying requirements. Section 1 describes the Introduction and motivation. Section 2 defines the result model. In section 3 the scoring function is discussed deeply. Section 4 presents the system implementation and gives the detailed keyword search algorithms. Section 5 presents the experiments. Section 6 is the related work. Section 7 is the conclusion of this chapter.

2.1 INTRODUCTION AND MOTIVATION
XML Keyword Search is a user-friendly information discovery technique, which attracts many interests these years. Different with keyword search over flat documents, the search object is a single XML document/database with structure information inside and the results are supposed to be fragments of it containing keywords. Since it is difficult and sometimes impossible to identify users’ intentions through keywords, it is really a hard task to determine which fragments should be returned. As a matter of fact, this is still an open issue of keyword search on structured and semi-structured data.

DOI: 10.4018/978-1-4666-1975-3.ch002
To define the results of XML keyword search, many valuable models are proposed, and the most popular ones are the Smallest Lowest Common Ancestor (SLCA) model (Xu & Papakonstantinou, 2005) and its variants (Guo, Shao, Botev, & Shanmugasundaram, 2003; Hristidis, Koudas, Papakonstantinou, & Srivastava, 2006; Kong, Gilleron, & Lemay, 2009; Guoliang, Jianhua, Jianyong, & Lizhu, 2007; Yunyao, Cong, & Jagadish, 2004; Ziyang, Jeffrey, & Yi, 2007; Xu & Papakonstantinou, 2008), all of which are called as the LCA-based models in this paper. These researches regard the XML document as a rooted, labeled, unordered tree in which each inner node is an element or an attribute and each leaf is a value which may contain some keywords. In SLCA model a result is defined as a subtree that: (1) the labels of whose nodes contain all the keywords, (2) none of its subtree satisfies the first condition except itself. The root of such a subtree is called a SLCA node. It’s recognized that SLCA model is definitely not a perfect one. After it’s proposed, arguments have been aroused around the meaningfulness and completeness of its search results (Kong et al., 2009; Guoliang et al., 2007; Xu & Papakonstantinou, 2008). Judging from the examples illustrated, they claim that some results of the SLCA method are meaningless and some meaningful ones are missing. These are called the false positive and the false negative problems respectively. Interestingly, the remedy approaches they propose sometimes conflict with each other, and counterexamples can always be found to testify that the two problems still happen.

In order to explain these issues we employ the original examples from former researches (Kong et al., 2009; Guoliang et al., 2007; Xu & Papakonstantinou, 2008), which are illustrated in Figure 1 as three separate XML trees. In these trees keywords are marked in bold and only important nodes are identified by numbers. Besides, we use node_i to denote the node with the number i in any of the trees. Consider keywords {“XML”, “David”} issued on the XML document in Figure 1(a), apparently SLCA method can find two SLCA nodes: node_7 and node_17, and the subtrees rooted in them will be returned as the final results. In many cases the subtrees are not appropriate for users because their sizes are too large and plenty of meaningless information is involved. GDMCT (Hristidis et al., 2006) approach proposes a good way to handle this that it returns the Minimum Connecting Trees (MCTs) instead. As a SLCA node corresponds to a group of keyword nodes (the nodes whose labels contain keywords), a MCT is defined as a subtree which employs the SLCA node as root and the keyword nodes as leaves. Furthermore, a MCT can be expanded to fill more useful information through semantics reference or according to the feedback from users. We can see that in SLCA approach either a subtree or a MCT is determined by a group of keyword nodes, and in fact this rule suits the results of any LCA-based approach. Since the refining of the final results is not a topic in this paper, for convenience a search result is considered as a group of keyword nodes instead of a document fragment by us. In this example, SLCA method finds two results {node_9, node_11} and {node_19, node_21} which are two separate papers in semantics.

MLCA (Yunyao et al., 2004) method and GDMCT (Hristidis et al., 2006) method should agree with the answers because they both define a result model very similar to SLCA. But, ELCA (Guo et al., 2003; Xu & Papakonstantinou, 2008) method will claim that several reasonable results are missed. For instance {node_3, node_25} and {node_16, node_23}, which indicate a conference and one of its sessions respectively. Neither node_1 nor node_14 is a SLCA node due to at least one SLCA node is their descendant node, however seemingly they’re both qualified results. As shown in Figure 1(a), these two new results along with the two SLCA results are all meaningful answers to the query {“XML”, “David”}, besides they are perfectly exclusive to each other.
Related Content

Semantic Query Expansion using Cluster Based Domain Ontologies
[www.irma-international.org/article/semantic-query-expansion-using-cluster/74781/](http://www.irma-international.org/article/semantic-query-expansion-using-cluster/74781/)

Inventory of Leader Sternness (ILS)
[www.irma-international.org/chapter/inventory-leader-sternness-ils/69741/](http://www.irma-international.org/chapter/inventory-leader-sternness-ils/69741/)

Query Recommendation for Improving Search Engine Results
[www.irma-international.org/article/query-recommendation-improving-search-engine/53126/](http://www.irma-international.org/article/query-recommendation-improving-search-engine/53126/)

Using Association Rules for Query Reformulation
[www.irma-international.org/chapter/using-association-rules-query-reformulation/64430/](http://www.irma-international.org/chapter/using-association-rules-query-reformulation/64430/)

Semantic Models in Information Retrieval
[www.irma-international.org/chapter/semantic-models-information-retrieval/64424/](http://www.irma-international.org/chapter/semantic-models-information-retrieval/64424/)