Chapter 13
Review on Keyword Search and Ranking Techniques for Semi-Structured Data

Dayananda P.
JSSATEB, India

Sowmyarani C. N.
Rashtreeya Vidyalaya College of Engineering, India

ABSTRACT

The size of semi-structured data is increasing continuously. Handling semi-structured data efficiently is a challenging task. Keyword search is an important task, and required information can be retrieved without having knowledge of data storage hierarchy. There are several challenges in handling XML data. This chapter discusses various challenges in terms of lowest common ancestor (LCA) semantics, processing of queries efficiently, retrieving top-k results for user needed data. The existing approach is defined under many classes based on how the problem and solution are tackled. Analysis of keyword search and ranking techniques for retrieving desired information are discussed in detail.

1. INTRODUCTION

Considering information society and information science development, storage of data in XML format is increasing over internet day by day. Standard way of data representation and exchanging of data over internet by making use of XML format. Effective way of extracting useful information is challenging job for researcher. There are many challenges to handle the data efficiently; many research methods are defined in the field of retrieving information on semi structured data. Basic understanding of how data is stored, semantics of handling data, ranking top k answers for query is very important task for researcher to give optimal solution for existing problem. Thus, taking into consideration of these requirements, in this chapter of the presented work, a number of literatures have been studied and explored for keyword search on semi-structured data.

DOI: 10.4018/978-1-5225-7347-0.ch013

2. TREE-BASED XML KEYWORD SEARCH

When XML documents do not contain IDREF, they can be modeled as trees. Approaches to handle such documents are called tree-based approaches because they are based on tree model. Inspired by the hierarchical structure of the tree model, most of existing tree-based approaches are based on the LCA (Lowest Common Ancestor) semantics, which returns the lowest common ancestors of matching nodes to keyword queries.

There are many subsequent semantics to filter less meaningful answers. Existing works either improve the effectiveness by proposing a new semantics or improve the efficiency by proposing a new method for certain semantics. The widely accepted LCA-based semantics include LCA itself, SLCA, VLCA, MLCA, ELCA, and etc, among which, SLCA and ELCA are the most popular semantics. We classify the existing research works into these semantics and result of our classification is shown in Figure 1. Some research work involves study of more than one semantic such as XRANK (Dayananda, 2016; Guo, 2003), Set-intersection (Bao et al., 2012), and Top-K (Dayananda, 2016).

2.1 LCA Semantics

The LCA semantics for XML keyword search was first proposed in XRANK (Guo et al., 2003). By the LCA semantics, for a set of matching nodes, each of which contains at least one query keyword and each query keyword matches at least one node in this set, the lowest common ancestor (LCA) of this set is a returned node. An answer is a subtree rooted as a returned node (i.e., an LCA) or a path from the returned node to matching nodes. XRANK is extended from Google's Pagerank algorithm for ranking. It takes into account the proximity of the keywords and the references between attributes. XRANK implements a naive approach, and three optimized approaches afterwards to improve the search.

2.2 SLCA Semantics

The SLCA (Smallest LCA) semantics was first proposed in XKSearch(Dayananda, 2016; Papakonstantinou & Xu, 2005). The SLCA semantics defines an SLCA to be an LCA that does not have any other LCAs as its descendants. There are many works on finding the set of SLCAs for a keyword query. XKSearch(Dayananda, 2016 ; Papakonstantinou & Xu, 2005) proposes two efficient algorithms to compute SLCAs, namely Indexed Lookup Eager and Scan Eager. To find all SCLAs, there are two tasks, namely finding all LCAs and remove all ancestors among LCAs to get the SLCAs. It is costly to find all LCAs.

XKSearch optimizes as follows. Firstly, for each matching node u of the keyword which has the least number of matching nodes XKSearch finds its left and right match. The left (right) match v of u refers the matching node of the other queries or the keyword and among all nodes in u's left (right) side, v is the nearest one (by pre-order). Only the LCA of u and v is a candidate SLCA. Thereby, it greatly reduces the number of computation of LCAs. The predominant characteristics of the approach SLCA refers that with provided two queries or keywords k1, k2 with one node u which comprises keyword k1, the one keyword seeks not to explore the complete keyword list so as to find the optimal solutions. In addition, one merely requires finding the left and right match of u in the k2 list, where the right or left match is the node with the minimal or maximum Dewey ID that is greater or smaller as compared to the Dewey ID of the node u. Multiway-SLCA (Dayananda, 2016; Chan et al., 2007) in addition attempted to enhance the performance of XKSearch scheme. The principal impetus behind this paradigm is the