XKFitter: A Keyword Filter on XML Stream

Weidong Yang, Fudan University, China
Fei Fang, Fudan University, China
Nan Li, Fudan University, China
Joan Lu, University of Huddersfield, UK

ABSTRACT

Most existing XML stream processing systems adopt full structured query languages, such as XPath or XQuery, but they are difficult for ordinary users to learn and use. Keyword search is a user-friendly information discovery technique that has been extensively studied for text documents. This paper presents an XML stream filter system called XKFitter, which is the first system for supporting keyword search over XML stream. In XKFitter, the concepts of XLCA (eXclusive Lowest Common Ancestor) and XLCA Connecting Tree (XLCACT) are used to define the search semantic and results of keywords, and present an approach to filter XML stream according to keywords. The prototype XKFitter is implemented in the experiments.

Keywords: DTD, Entity Search Term, Filter, Keyword Search, XML Stream

INTRODUCTION

Stream-based continuous query processing (Babcock, Babu, Datar, Motwani, & Widom, 2002; Madden, Shah, Hellerstein, & Raman, 2002) fits a large class of new applications, such as sensor networks, location tracking, network management, publish-subscribe systems. As eXtensible markup language – XML is a standard for information exchange, the problem of processing streaming XML data is gaining widespread attention from the research community (Babcock et al., 2002; Diao, Altinel, Franklin, Zhang, & Fischer, 2003; Peng & Chawathe, 2005). An XML stream system (XSS) aims to provide fast and on-the-fly matching of XML-encoded data to user’s query, which is different from traditional XML database management systems (Lu & Rahman, 2007). The XSS usually involves handling the XML stream coming online at any moment and any order, and requiring timely response without incurring more memory cost. Therefore, the numbering schemes like Dewey numbers and XML indexing techniques for accelerating query process in XML databases don’t apply to XML data streams processing generally. For XML stream systems, currently, most existing researches adopt full structured query languages such as XPath or XQuery. These query languages can convey complex meaning in the query specifications containing constraints on both structure and content of an XML document, thus, can precisely retrieve the desired results.

DOI: 10.4018/ijirr.2011010101
However, for an ordinary user, especially for a web user, it is difficult to learn the complex query languages, it is also impossible to write a correct query without knowing the exact structure of an XML document.

Keyword search is a user-friendly information retrieval technique that has been extensively studied for text documents. Unlike structured queries on database which adopts exact match approach, the keyword search adopts best match approach which has to "guess" the best search results and provide an appropriate rank model; different from traditional information retrieval systems, keyword search on database, instead of retrieving whole documents, aim at retrieving content components of the whole database, i.e. joined tuples (for relational database) or XML elements (for XML database) of varying granularity that fulfill the user’s query. Recently, many researchers in database field extended this technique into relational database (Liu, Yu, Meng, & Chowdhury, 2006) and XML database (Cohen, Mamou, Kanza, & Sagiv, 2003; Guo, Shao, Botev, & Shanmugasundaram, 2003; Hristidis, Papakonstantinou, & Balmin, 2003; Hristidis, Koudas, Papakonstantinou, & Srivastava, 2006; Liu, Walker, & Yichen, 2007; Xu & Papakonstantinou, 2005) by combining information retrieval techniques and database techniques, and proposing various approaches to define and rank the keyword search results, and developing algorithms to accelerate the execution of keyword search. It is noted that keyword search is also well-suited to some applications under streams data processing environment such as publish-subscribe systems, web monitoring systems. Alexander et al. (Markowitz, Yang, & Papadias, 2007) presented a system called “S-KWS” for keyword search on relational data streams.

XML technology has its reputation in semantic representation of information and knowledge in the subject areas, because of its underpinned theory: ontology, which could define or constrain the unique feature of DTD and schema (Lu, 2005; Lu & Rahman, 2007). The purpose to Integrate Keyword search technology into semantically oriented XML system is to increase the simplicity, efficiency and effectiveness during retrieval process (Lu & Fox, 2007).

In this paper, we focus on keyword search on XML Stream. The main contributions made in the paper are:

1. We present a new software system: XKFilter, to the best of our knowledge, the first system for keyword search on XML Stream. XKFilter can process XML streams with or without schemas.
2. In this system, unlike relational data streams, XML streams are self-described, with its content and multilevel structure which can be recursive mixed together. When XML DTD is not available, we let users register their queries by writing several keywords simply (called pure keyword search), and we define the search results as eXclusive Lowest Common Ancestor Connecting Tree (XLCAT) based on our previous work (Josifovski, Fontoura, & Barta, 2005). Also, we design an algorithm “S-XSF” based on stack to process XML stream and buffering return results efficiently in a single pass.
3. XKFilter provides a user-friendly search interface based on both keyword search and DTD, if the XML DTD is available. This search interface is a simple expansion of pure keyword search syntax which contains a list of search terms. Each search term is composed of a keyword and the label containing keyword directly. By using the search terms of users and XML DTD, the keyword search can be refined to a Keyword Query Graph (KQG). As a result, the XML fragment unrelated to the keywords search will be ignored. It has two benefits: (1) effectively reducing the search space of XML streams; (2) effectively reducing the parsing amount of XML streams. In order to process XML stream in a single pass, base on the KQG, the Keyword Query State Machine (KQSM) is built. The KQSM supports state forward transition and backward transition, and use a run-time stack to process the XML stream and buffering return results effectively.
Experiments and their Assessment
Ibrahim Dweib and Joan Lu (2013). Design, Performance, and Analysis of Innovative Information Retrieval (pp. 249-263).
www.igi-global.com/chapter/experiments-their-assessment/69141?camid=4v1a