Chapter 6

Keyword Search in XML Streams

Weidong Yang
Fudan University, China

Hao Zhu
Fudan University, China

ABSTRACT

Most existing XML stream processing techniques adopt full structured query languages such as XPath or XQuery, which are difficult for ordinary users to learn and use. This chapter presents an XML stream filter system called XKFilter, which uses keyword to filter XML streams. In XKFilter, we use the concepts of XLCA (eXclusive Lowest Common Ancestor) and XLCA Connecting Tree (XLCACT) to define the search semantic and results of keywords, and present an approach to filter XML stream according to keywords. In section 1, the background of keyword search in XML streams is introduced. Section 2 explains the searching results. In section 3, a stack-based keyword searching algorithm for XML stream filtering without schemas is presented in-depth. Section 4 presents a keyword search over XML streams by using schema information. The system architecture of XKFilter is described in section 5. Section 6 is the experiments to show the performance. Section 7 discusses the related work. Section 8 is the summaries of this chapter.

6.1 INTRODUCTION

Stream-based continuous query processing (Babcock, Babu, Datar, Motwani & Widom, 2002) fits a large class of new applications, such as sensor networks, 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 (Diao, Altinel, Franklin, Zhang & Fischer, 2003). 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. 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 XSS, currently, most existing researches adopt full structured query languages such as XPath (Berglund, Boag, Chamberlin, Fernandez, Kay, Robie & Simon, 2002) or XQuery Chamberlin, D. (2003). 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. 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 (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. Alexander et al (Markowetz, Yang & Papadias, 2007) presented a system called “S-KWS” for keyword search on relational data streams.

In this chapter, we focus on keyword search on XML Stream, including:

- A prototype system called XKFilter is introduced, which can process XML streams with or without schemas.
- When XML DTD is not available, XKFilter lets 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 (XLCACT) based on our previous work (Xu, Lu, Wang & Shi, 2006). Also, we design an algorithm “S-XSF” based on stack to process XML stream and buffering return results efficiently in a single pass.
- 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: (i) effectively reducing the search space of XML streams; (ii) 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.
Related Content

On Knowledge-Enhanced Document Clustering
[www.irma-international.org/article/knowledge-enhanced-document-clustering/78315/](www.irma-international.org/article/knowledge-enhanced-document-clustering/78315/)

Personalized Content-Based Image Retrieval
[www.irma-international.org/chapter/personalized-content-based-image-retrieval/28074/](www.irma-international.org/chapter/personalized-content-based-image-retrieval/28074/)

Chronological Ordering Based on Context Overlap Detection
Mohamed H. Haggag and Bassma M. Othman (2012). *International Journal of Information Retrieval Research* (pp. 31-44).
[www.irma-international.org/article/chronological-ordering-based-on-context-overlap-detection/90440/](www.irma-international.org/article/chronological-ordering-based-on-context-overlap-detection/90440/)

Advanced Branching and Synchronization Patterns Description Using Pi-Calculus
[www.irma-international.org/chapter/advanced-branching-synchronization-patterns-description/69151/](www.irma-international.org/chapter/advanced-branching-synchronization-patterns-description/69151/)

Business Process Modeling: Analysis and Evaluation
[www.irma-international.org/chapter/business-process-modeling/69150/](www.irma-international.org/chapter/business-process-modeling/69150/)