A Unifying Framework for Merging and Evaluating XML Information

Ho-Lam Lau and Wilfred Ng

Department of Computer Science,
The Hong Kong University of Science and Technology, Hong Kong
{lauhl, wilfred}@cs.ust.hk

Abstract. With the ever increasing connection between XML information systems over the Web, users are able to obtain integrated sources of XML information in a cooperative manner, such as developing an XML mediator schema or using eXtensible Stylesheet Language Transformation (XSLT). However, it is not trivial to evaluate the quality of such merged XML data, even when we have the knowledge of the involved XML data sources. Herein, we present a unifying framework for merging XML data and study the quality issues of merged XML information. We capture the coverage of the object sources as well as the structural diversity of XML data objects, respectively, by the two metrics of Information Completeness (IC) and Data Complexity (DC) of the merged data.

1 Introduction

Information integration, a long established field in different disciplines of Computer Science such as cooperative systems and mediators, is recognized as an important database subject in a distributed environment [5, 8]. As the networking and mobile technologies advance, the related issues of information integration become even more challenging, since merged data can be easily obtained from a wide spectrum of emerging modern data applications, such as mobile computing, peer-to-peer transmission, mediators, and data warehousing.

As XML data emerges as a de-facto standard of Web information, we find it essential to address the quality issues of integrated XML information. In this paper, we attempt to establish a natural and intuitive framework for assessing the quality of merging XML data objects in a co-operative environment. We assume that there are many XML information sources which return their own relevant XML data objects (or simply XML data trees) as a consequence of searching for a required entity from the users. To gain the maximal possible information from the sources, a user should first query the available sources and then integrate all the returned results. We do not study the techniques used in the search and integration processes of the required XML data objects as discussed in [1, 2, 3]. Instead, we study the problem of how to justify the quality of merged XML information returned from the cooperative sources.

We propose a framework to perform merging and to analyze the merged information modelled as multiple XML data objects returned from a set of XML

L. Zhou, B.C. Ooi, and X. Meng (Eds.): DASFAA 2005, LNCS 3453, pp. 81–94, 2005.

information sources. Essentially, our analysis is to convert an XML data object in an *Merged Normal Form* (MNF) and then analyze the data content of the normalized object based on a *Merged Tree Pattern* (MTP). We develop the notions of *Information Completeness* (IC) and *Data Complexity* (DC). These are the two components related to the measure of the information quality.

Intuitively, IC is defined to compute the following two features related to the completeness of those involved information sources. First, how many XML data objects (or equivalently, XML object trees) can be covered by a data source, and second, how much detail does each XML data object contain. We call the first feature of IC the *merged tree coverage* (or simply the *coverage*) and the second feature of IC the *merged tree density* (or simply the *density*).

The motivation for us to define IC is that, in reality when posing queries upon a set of XML information sources that have little overlaps in some predefined set of core labels $\mathcal{C}$, then the integrated information contains a large number of distinct XML data objects but with few subtrees or data values under the core labels, in this case the integrated information has comparatively high coverage but low density. On the other hand, if the sources have large overlaps in $\mathcal{C}$, the integrated information contains a small number of distinct objects with more subtrees or data elements under the core labels, in this case the integrated information has comparatively low coverage but high density.

The metric DC is defined to compute the following two features related to the complexity of the retrieved data items, resulting from merging data from those involved information sources. First, how diversified the merged elements or the data under a set of core labels are, and second, how specific those merged elements or data are. We call the first feature of DC the merged tree diversity (or simply the *diversity*) and the second feature of DC the merged tree specificity (or simply the *specificity*). In reality, when we merge the data under a label in $\mathcal{C}$ it may lead to a too wide and deep tree structure. For example, if most data of the same object from different sources disagree with each other, then we have to merge a diverse set of subtrees or data elements under the label. Furthermore, the merged tree structure under the label can be very deep, i.e. to give very specific information related to the label.

We assume a global view of data, which allows us to define a set of core labels of an entity that we search over the sources. As a core label may happen anywhere along a path of the tree corresponding to the entity instance, we propose a Merge Normal Form (MNF). Essentially, an XML object in MNF ensures that only the lowest core label along a path in the tree can contain interested subtrees or data elements. Assuming all XML objects are in MNF we aggregate them into a universal template called Merged Tree Pattern (MTP). We perform merging on the subtrees or data values associated with $\mathcal{C}$ from XML tree objects: if the two corresponding core paths (paths having a core label) from different objects are equal, then they can be unanimously merged in MTP. If the two paths are not equal, the conflict is resolved by changing the path to a general descendant path. Finally, if the two core paths do not exist then they are said to be incomplete, the missing node in MTP will be counted when computing IC.