Flexible Information Retrieval on XML Documents

Torsten Grabs and Hans-Jörg Schek

6.1 Introduction

XML – short for the eXtended Markup Language defined by the World Wide Web Consortium in 1998 – is very successful as a format for data interchange. The reason is the high flexibility of the semistructured data model underlying XML [3]. Therefore, XML documents are well suited for a broad range of applications covering both rigidly structured data such as relations as well as less rigorously structured data such as text documents. So far, research on database systems has spent much effort on data-centric processing of rigidly structured XML documents. However, the importance of document-centric processing increases the more XML extends to application domains requiring less rigorously structured data representation.

In the context of document-centric XML processing, this chapter focuses on the problem of flexible ranked and weighted retrieval on XML documents [139]. Like information retrieval on text, XML retrieval aims to effectively and efficiently cover the information needs of users searching for relevant content in a collection of XML documents. However, due to the flexibility inherent to XML, conventional text retrieval techniques are not directly applicable to ranked and weighted retrieval on XML documents.

First, the notion of a document collection must be refined in the context of XML. This is because often a single large XML document comprises all the content. Figure 6.1 illustrates such an XML document representing a collection of books. A document collection in the context of conventional information retrieval in contrast usually comprises many documents. For instance, global collection-wide IR statistics such as document frequencies, i.e., the number of documents a word occurs in, build on the conventional notion where a collection comprises many documents.

Second, different parts of a single XML document may have content from different domains. Figure 6.1 illustrates this with the different branches of the bookstore – one for medicine books and one for computer science books. Intuitively, the term 'computer' is more significant for books in the medicine
branch than in the computer science branch. Conventional information retrieval however assumes that the documents of the collection are from the same domain. Again, global IR statistics such as document frequencies depend on this assumption that seems not longer appropriate in the context of XML. Note that this is indeed an issue when users refer to the different domains when posing their queries. We distinguish two different cases in this respect, namely single-category retrieval and multi-category retrieval.

Single-category retrieval stands for queries that refer to a single domain or category in isolation, i.e., only content from one category is subject to the query. The difficulty with this type of queries is that global IR statistics such as document frequencies may not be meaningful depending on the scope of the query. Think again of a collection with only a single large XML document: conventional document frequencies cannot be larger than 1 in this case. Document frequencies of terms occurring in the collection always equal 1 since there is only a single large document. Document frequencies of the other terms are 0, as usual. Hence, a more fine-grained resolution of global IR statistics is needed in the context of XML retrieval, as the following XML illustrates.

Example 1 (Single-Category Retrieval). Consider a user searching for relevant books in the computer science branch of the example document in Fig. 6.1. Obviously, he restricts his queries to books from this particular category. Thus, it is not appropriate to process this query with term weights derived from both the categories medicine and computer science in combination. This is because the document frequencies in medicine may skew the overall term weights such that a ranking with term weights for computer science in isolation would increase retrieval quality.

Now think of another user who wants to process a query on several categories, or he does not care at all to which category the retrieved content