A flexible XML query language for NON dummies

Stefania Marrara[1], Emanuele Panzeri[2], and Gabriella Pasi[2]

[1] Università degli Studi di Milano - DTI
via Bramante 65, 26013 Crema (CR), Italy, stefania.marrara@unimi.it
[2] Università degli Studi di Milano Bicocca - DISCo
viale Sarca 336, 20126 Milano (MI), Italy, {panzeri,pasi}@disco.unimib.it

Abstract. This paper introduces and motivates our proposal of an XPath extension which allows the definition of queries with flexible constraints on both content and structure. The proposed language allow expert users to benefit of the recall improvements of flexible languages while using their collection knowledge to improve the retrieval precision.

1 Introduction and Motivation

XML (Extensible Markup Language), a World Wide Web Consortium (W3C) standard for the World Wide Web, since its first introduction in 1998 has proved its ability to be the basis for the data interchange on the Internet. Differently from the most popular HyperText Markup Language (HTML), XML allows document designers to set up their own tag vocabulary, and to describe the structure of documents by defining tag nesting. It is widely recognized that, in order to exploit the power of XML, a query language for XML documents should allow constraint formulation on both document content and structure. In other words, it should represent tag nesting as well as their content. The birth of huge collections of XML documents has implied the definition of appropriate query languages, whose main representatives are today the W3C standards XPath [20] and XQuery [21]. The main issue with these XML query languages is that they assume the user be fully aware of the target document structure, and allow only an exact specification of the target documents, due to the boolean nature of their query-document matching systems. This assumption is debatable since most XML documents have no pre-set structure (DTD or XML Schema); even worse, it requires the user to write a different query for each variation of the document structure itself. In order to tackle this problem, in the last few years both Information Retrieval (IR) and Database (DB) communities proposed flexible XML query languages with search paths that provide a loose example of the information the user is interested in. In Information Retrieval, the proposals are broadly classified as *content-only* search (CO) and *content and structure* search (CAS). CO approaches usually allow keyword based queries without any possibility to specify constraints on the expected document structure. Most of the existing keyword search systems return query results based on the notion of *Lowest Common Ancestor (LCA)* [15], and its variants [16, 14]. On the opposite

side, CAS approaches focus on approximate matching of limited XPath predicates (usually the *child* axis is transformed into a *descendant* axis during the query evaluation), and on designing indexes to score document fragments. These approaches are based on the notion of *structural hint* which considers the query structure as a mere template of the required information. All fragments similar to the template are retrieved. In the DB community, the most important contributes are XPath Full Text (XPathFT), and XQuery Full-Text (XQueryFT) [1], which include full-text capabilities in the traditional query languages. Differently from CAS approaches, XPathFT and XQuery FT do not include structural hints but query structures are evaluated as in the traditional standards. All mentioned approaches focus on the notion of *dummy* user: the main idea is that users are often unable or too lazy in order to formalize complex queries and therefore keyword based queries or queries without too strict structural constraints are preferred. In this work we start from a different perspective. In the last years many important collections such as DBPL, Wikipedia, the Cancer Gene Disease and Gene Compound or the Protein Data Bank (PDBML) have been created by the scientific and research communities. Users belonging to these communities are often scientists, high-educated people who, due to their continue use, develop a good, even if partial (due to its dimensions), knowledge of the collection document structure, and therefore they are both able to and interested in formalizing complex queries if this can improve the retrieval precision. With this idea in mind, in [11] a flexible XPath extension has been proposed (briefly sketched in Section 2), which allows users to define the extent and type of desired constraint relaxation in the query both on content and structure. In this approach the query is not considered as a mere template of the required information, as the user can include exact and flexible constraints in the same query on both content and structure. In this way the user can benefit of the recall improvements typical of flexible languages while exploiting his/her knowledge of the considered collection to improve the retrieval precision.

1.1 Related Work

Recent research in both Information Retrieval and Database communities has led to several approaches aimed at introducing some degrees of flexibility in XML retrieval [9, 12, 18]. In the Information Retrieval research context, CO approaches address the issue of querying XML documents by using a keyword based approach [2, 7, 8], while CAS models consider both document content and structure in the retrieval process: just as the keywords are hints, so are the structural constraints. Examples of CAS approaches are offered by, for instance, XIRQL [13], NEXI [19], TeXQuery [3], and FleXPath [5], and the recent standard XQuery and XPath Full Text 1.0 [1]. Other approaches proposed in the literature define some flexibility on the evaluation of the query structure in a more explicit way. For instance, in [13], [17], and [4] the authors define some relaxations such as the introduction of generalized data types, the adoption of edit distances on paths, and some operations to modify the structure like *delete* a node, *insert* intermediate nodes or *rename* a node. FleXPath [5] is the first approach proposing a formalization of relaxations on the evaluation of the structure

of XML queries, as well as the first algebraic framework for spanning relaxations. In addition, it proposes new ranking functions with properties that relaxations must satisfy, and it develops efficient evaluation algorithms.

All the above mentioned approaches, however, do not allow users to define the extent and the type of the desired flexible constraints in the query: the query is considered as a mere template of the required information. All fragments similar to the template are retrieved. By these approaches the user has no possibility to distinguish between portions of the query that must be considered as exact and those that allow a certain flexibility in the retrieval process. In order to allow users to explicitly specify both content-based and structure-based flexible constraints in a query aimed at retrieving XML fragments, in [11, 6] a flexible language has been proposed and defined as an extension of the XPath query language. The flexible constraints have been syntactically defined in compliance with the XPath language; the query evaluation produces a score, expressing the degree of compatibility of the document's content/structure with respect to the user requirement. In Section 2 this language is briefly presented.

2 A flexible extension of XPath

In [6, 10, 11] a new approach aimed at introducing flexibility in the XPath query language has been defined. This new language extends the syntax of XPath to allow the definition of some flexible constraints on both the content (the defined constraints are named *cw* and *around*), and the structure of the XML document (the defined constraints are named *near*, *below*, and *approximately*). Informally, *cw* (*contain words*) is applied to nodes that have a textual content, and is introduced to specify keyword-based constraints, as in usual IR query languages. The constraint *cw* is followed by the list of search terms to be retrieved in the textual element.

The constraint *around* is applied to numeric or date values (within specific numeric or data content nodes), and it requires their approximate evaluation. The structure-based constraints *below* and *near*, inserted as a flexible axis of a path expression, allow to extract XML fragments (i.e. elements, attributes or text) that are, respectively, direct descendants or connected through any path to the current node (also allowing users to fix a maximum path length). For each retrieved fragment a retrieval status value is computed that is inversely proportional to the distance (measured in number of arcs) between the *ideal* path structure and the retrieved one. Finally, the constraint *approximately* allows to select the elements with a given name that have a number of direct descendants close to the one indicated in the query.

3 Conclusions

The aim of this paper was to introduce and motivate the proposal of an XPath extension that allows the specification of flexible constraints on both content and structure of XML documents. The proposed language, briefly sketched in Section 2, allows expert users to benefit of the recall improvements typical of flexible languages. In future work the full syntax and semantics of this XPath

extension will be presented, as well as the definition of an appropriate data structure; moreover evaluations will be performed.

References

1. XQuery and XPath Full Text 1.0. http://www.w3.org/TR/xpath-full-text-10/.
2. M. S. Ali, M. P. Consens, G. Kazai, and M. Lalmas. Structural relevance: a common basis for the evaluation of structured document retrieval. In *CIKM'08*, pages 1153–1162, 2008.
3. S. Amer-Yahia, C. Botev, and J. Shanmugasundaram. Texquery: a full-text search extension to xquery. In *WWW'04*, pages 583–594, 2004.
4. S. Amer-Yahia, S. Cho, and D. Srivastava. Tree pattern relaxation. In *EDBT'02*, pages 496–513, 2002.
5. S. Amer-Yahia, L. V. S. Lakshmanan, and S. Pandit. Flexpath: flexible structure and full-text querying for xml. In *SIGMOD'04*, pages 83–94, 2004.
6. A. Campi, E. Damiani, S. Guinea, S. Marrara, G. Pasi, and P. Spoletini. A fuzzy extension of the xpath query language. *J. Intell. Inf. Syst.*, 33(3):285–305, 2009.
7. D. Carmel, Y. S. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer. Searching xml documents via xml fragments. In *SIGIR'03*, pages 151–158, 2003.
8. C. L. A. Clarke. Controlling overlap in content-oriented xml retrieval. In *SIGIR'05*, pages 314–321, 2005.
9. S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv. Xsearch: a semantic search engine for xml. In *VLDB'03*, pages 45–56, 2003.
10. E. Damiani, S. Marrara, and G. Pasi. Fuzzyxpath: Using fuzzy logic an ir features to approximately query xml documents. In *IFSA'07*, pages 199–208, 2007.
11. E. Damiani, S. Marrara, and G. Pasi. A flexible extension of xpath to improve xml querying. In *SIGIR'08*, pages 849–850, 2008.
12. D. Florescu, D. Kossmann, and I. Manolescu. Integrating keyword search into xml query processing. *Comput. Netw.*, 33(1-6):119–135, 2000.
13. N. Fuhr and K. Grobjohann. Xirql: a query language for information retrieval in xml documents. In *SIGIR'01*, pages 172–180, 2001.
14. J. W. G. Li, J. Feng and L. Zhou. Effective keyword search for valuable lcas over xml documents. In *CIKM*, pages 31–40, 2007.
15. C. B. L. Guo, F. Shao and J. Shanmugasundaram. Xrank: Ranked keyword search over xml documents. In *SIGMOD*, pages 16–27, 2003.
16. Y. K. S. Cohen, J. Mamou and Y. Sagiv. Xsearch: A semantic search engine for xml. In *VLDB*, pages 45–56, 2003.
17. T. Schlieder. Similarity search in xml data using cost-based query transformations. In *WebDB'01*, 2001.
18. A. Theobald and G. Weikum. Adding relevance to xml. In *WebDB'00*, pages 105–124, 2001.
19. A. Trotman and B. Sigurbjörnsson. Narrowed extended xpath i (nexi). In *INEX'04*, pages 16–40, 2004.
20. W3C. Xml path language (xpath) 2.0. http://www.w3.org/TR/xpath20/, November 2007.
21. W3C. Xquery 1.0: An xml query language. http://www.w3.org/TR/xquery/, November 2007.