Formulating XML-IR Queries

Alan Woodley
Faculty of Information Technology, Queensland University of Technology
PO Box 2434. Brisbane Q 4001, Australia
ap.woodley@student.qut.edu.au

XML information retrieval systems differ from traditional information retrieval systems by returning relevant portions of documents, rather than entire documents. Theoretically, this should better fulfil the information needs of users, especially in situations where their information need is very complex. However, if users are going to exploit this advantage then they need a query formation interface that is both sophisticated and intuitive. This paper outlines four potential query formation interfaces: keywords, formal language, natural language and query by templates. For each interface it: outline the advantages and disadvantages, presents comparative results stemming from experiments and proposes several future research areas involving the four interfaces.

Keywords: XML Information Retrieval, Natural Language Queries, Formal Language Queries, Query by Template

1. INTRODUCTION

Traditional Information Retrieval (IR) systems respond to user queries with ranked lists of relevant documents. Documents formatted in Extensible Markup Language (XML) differ from flat-text documents by explicitly containing both content and structure. By exploiting this difference, XML information retrieval (XML-IR) systems have the potential to present users with highly relevant and highly focused results, thereby, better fulfilling their information needs than traditional (flat text) IR systems. XML-IR research has been accelerated by the Initiative for the Evaluation of XML retrieval (INEX) [1,2]. INEX is an evaluation forum, comparable to TREC, that provides participants with a shared collection to compare retrieval approaches. Historically, much of the research in the field of XML-IR, has been system rather than user oriented. This lack of focus on users has the potential to limit the use of XML-IR systems. This paper presents research that focuses on the needs of the users, specifically, how they formulate XML-IR queries.

It is believed that the information needs of XML-IR users are more complex than traditional IR users since they contain both content and structural needs. Content needs define what information the user is seeking, while structural needs define where in the document that information is likely to located and the level of target element that the user wishes to retrieve. A challenge within the XML-IR community has been locating an interface that is both sophisticated, so that users could express both their content and structural needs, and easy to use, so that users could formulate queries intuitively. This paper outlines four options for XML-IR query formation: keywords; formal language; natural language and query by template. While the focus on the paper is on XML-IR, many of the issue raised are applicable to other information seeking domains such as database access, web queries and the information use community. The paper begins by describing each of the interfaces and outlining their advantages and disadvantages; then, it presents the results from a series of experiments involving each of the interfaces; finally, it provides a discussion on the future directions of XML-IR query formation research.

Keywords: global warming cause and effects

Formal Language: //section[about(., global warming cause and effects)]

Natural Query Language: Find sections about global warming cause and effects.

FIGURE 1: Keywords, Formal Language and Natural language Queries

2. XML-IR QUERY FORMATION INTERFACES

Historically, four interfaces have been used to formulate XML-IR queries: namely: keywords; formal language; natural language and query by template. These interfaces can be grouped into two classes: content only (keyword) and structured (formal language, natural language and query by template); with the difference being that structured interfaces incorporate both users' content and structural information needs. Figure 1 shows examples of keyword, formal language and natural language queries. Figure 2 shows an example of a query by template query. This section outlines each of these interfaces and discusses their advantages and disadvantages.

2.1 Keywords

Keyword interfaces, such as those used in major Internet search engines, have historically been used as the standard form of input for most information retrieval systems. Keyword interfaces are easy to use and suitable for the traditional information retrieval paradigm where documents' content is largely treated like a "bag of words"'. However, since XML documents consist of both content and structure, XML-IR systems should exploit this structure to better fulfil users' information needs; particularly, when the users' information needs are complex. However, keyword interfaces are too unsophisticated to fully capture XML-IR users' complex information needs. In particular, keyword interfaces are unable to capture users' structural requirements, so, users are not able to specify the portion of documents that they want searched or returned.

2.2 Formal Languages

Although not widely used in information retrieval, formal languages are the standard interface for database access. Examples of formal query languages for XML-IR are XPath [3] and NEXI [4]. An advantage of formal languages is that they are very powerful and can capture the users' structural and content needs. However, there are two main liabilities with formal query languages.

First, formal query languages are too difficult for users, both expert and casual, to correctly express their structural and content information needs. For example, at the 2003, INEX workshop XPath was used to specify structured queries; however, 63 per cent of the proposed queries had major semantic or syntactic errors and required 12 rounds of corrections [4]. In 2004, INEX used NEXI [5], a simplified version of XPath, and the error rate dropped to 12 percent, with the number of topic revisions halved [6], although this is still very high for expert users. User-based experiments have confirmed the difficulty that casual users have in formulating their needs with formal query languages [7].

Second, formal query languages are too tightly bound to the physical structure of documents and require users to have an intimate knowledge of the documents' composition to express their structural need. For example, in order to retrieve information from abstracts, bodies or bibliographies, users need know the actual names of those tags in a collection (for instance: **<abs>**, **<bdy>** and **<bib>**). While this information may be obtained from a document's DTD or Schema there are situations where the proprietor of the collection does not want this information to be publicly accessible. Furthermore, the number of tags in a collection may be too large to be remembered by users. This problem is magnified in heterogeneous collections since multiple tag names could refer to the same structural elements. For instance, one collection may use the tag name (**<p>**) to denote paragraphs while another collection may use the tag name **<para>**.

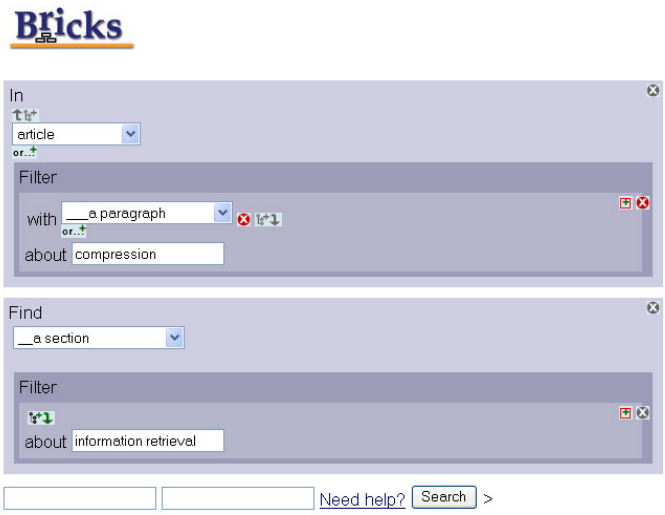

FIGURE 2: Bricks - Query by Template

2.3 Natural Language Interfaces

Natural language interfaces accept queries written in a natural language such as English. Historically, they have been used in both information retrieval and database access systems. Natural language queries have several advantages over other interfaces. In comparison with keyword interfaces, they are able to encapsulate users' content and structural information needs. And, in comparison with formal languages, they are both easier to use, since users should be able to intuitively express their information needs in their natural language, and they allow users to express their structural needs at the conceptual — rather than physical — level; so, while users will need to know that journal articles contain abstracts, bodies, and bibliographies they will not need to know the physical tag names of those elements.

Despite these advantages, natural language interfaces can sometimes have problems interpreting users' queries. In particular, ambiguous queries are known to cause problems for natural language interfaces since they lack a single interpretation. However, non-ambiguous queries can also be incorrectly interpreted by natural query interfaces due to errors in design or implementation. In addition, users can be confused on whether the errors are due to malformed queries or correctly formed queried that are misinterpret by the interface.

Recently several natural language interfaces have been developed for XML-IR systems. One such system is NLPX [8]. NLPX accepts natural language queries (NLQs) and produces formal queries written in the NEXI language. For each NLQ, the translation process involves four steps. First, NLPX tags words either as special connotations (for words of semantic importance) or by their part of speech (for all other words). Second. NLPX divides the query into atomic, non overlapping segments (called chunks) and classifies them into grammatical classes. Third, NLPX matches the query to templates, derived from the inspection of previous structured queries. Finally, NLPX outputs the query in NEXI format.

2.4 Query by Template

Query by template interfaces are a special type of GUI that allows users to create an arbitrary number of forms. Query by template interfaces have been used in database access community and over the other types of interfaces. First, they are more sophisticated than keyword interfaces and are able to capture users' structural and content needs. Secondly, they are easier to use than formal language interfaces and express structure at a conceptual level. Finally, unlike natural language interfaces, they do not allow users to enter ambiguous queries.

Despite these advantages, query by template interfaces have two disadvantages in comparison with natural language interfaces. Firstly, although query by template interfaces are easier to use than formal query interfaces they may not be as intuitive as natural language interfaces. In particular, casual users could find navigating drop down menus and multiple textboxes cumbersome. Secondly, query by template interfaces are less expressive than natural languages and maybe even less expressive than formal query languages.

Recently, a query by template interface, named Bricks [7], was developed for XML information retrieval systems. Bricks allows users to enter their structural needs via drop down menus and their contents needs via textboxes. To aid users, structural needs are indicated via conceptual rather than physical names. Furthermore, Bricks is also able to handle queries that contain multiple information requests by allowing users to develop queries in several steps (``blocks'') starting with their desired unit of retrieval, then by adding any additional information needs. Blocks are added as the user traverses the hierarchy of the documents. Upon completion of input, the data in the Bricks GUI is translated to a formal NEXI expression; however, due to the constraints of the GUI, it is impossible for users to enter malformed expressions.

3. EXPERIMENTS

Several experiments have been executed comparing both the retrieval performance and usability of the various interfaces. The experiments can be classified into two groups: system based experiments, in the tradition of Cranfield [11], and user based experiments, conducted under Borland's simulated work task environment [12].

3.1 System Testing

INEX's Ad-hoc task is evaluated using the Cranfield methodology, albeit slightly modified to handle the particularities of XML retrieval. In the annual INEX proceedings several participants have reported the retrieval performance of their systems when keywords and NEXI queries are used as input [9]. While some systems report superior retrieval performance using when NEXI queries are used as input the increase in performance is small.

The first system wide comparison between keyword and NEXI queries was conducted by Trotman and Lalmas [10]. In their study, Trotman and Lalmas compared the retrieval of all INEX 2005 Ad-hoc participants when NEXI and keywords were used as input. They concluded that while the retrieval performance of some systems improved when NEXI was used as input, the improvement was not significant. Trotman and Lalmas proposed two reasons why this occurred: first, the originators of the topics (that is INEX participants) are not able to write NEXI queries that successfully use structure and second, that the INEX collection (which consisted of a set of IEEE journal articles) did not contain a high number of semantically significant tags.

Comparisons have also between systems using NEXI and natural language queries as input. Most of these comparisons have been made by participants in INEX's Natural Language Processing track. Participants in the NLP track developed interfaces that translate natural language queries into formal language (NEXI) queries. The translated queries are then executed on a single backend retrieval. Also executed is an equivalent set of manually constructed formal language queries. Historically, the retrieval performance of the backend systems using the translated queries has being about 80% of its performance using the manually constructed formal language queries.

3.1 User Testing

The field of interactive information retrieval evaluation was established to collect quantitative and qualitative feedback from users regarding their use of IR systems. The INEX interactive track was established to focus on the needs of the user, however, most interactive XML-IR experiments have focused on results presentation rather than other areas of interaction. Despite this, a small number of researchers have explored the area of XML-IR query formulation.

Zwol et al. [7] investigated how users formulate queries using keywords, formal language and query by template interfaces. Their analysis showed that, in terms of retrieval performance, the structured interfaces (formal languages and query by template) were superior to the keywords only interface and that, in terms of usability, Bricks was superior to the other interfaces. The results also showed that as the complexity of the query increased so did the efficiency of the query by template interface.

Woodley et al. [13] investigated how users formulate queries using keywords, query by template and natural language interfaces. Both in terms of retrieval performance and usability, the three interfaces performed comparably; however, since the number of participants was small any quantitative scores derived are not statistically significant. Interviews following the experiment discovered that participants felt that while the natural language interface was easier to use, they were not sure if it was correctly interpreting their structural requirements.

4. FUTURE OUTLOOK

Despite the progress made in the field of XML-IR, and specifically XML-IR query formation some gaps of knowledge remain. This section highlights some remaining questions and discusses future research that could be performed in this area. Interestingly, the applicability of both the questions and potential research extends beyond XML-IR into database access, web queries and general information use.

1.1 Are Structural Requirements Useful?

The first unanswered question for XML-IR query formation is: are the addition of users' structural requirements useful? The work of Trotman and Lalmas [10] suggests that they are not and that superior retrieval performance can be achieved using keywords, although alternative research suggests otherwise [7]. Trotman and Lalmas provides two possible reasons why structure did not aid retrieval: first, that the collection used was unsuitable and two, that users where unable to produce useful structured queries.

In addition to the points raised by Trotman and Lalmas, there are two other possible reasons why the addition of structural requirements did not benefit retrieval in their study. First, is that the general information seeking task, where the information need is vague and the number of relevant items large, may not be suitable for XML-IR; instead, maybe XML-IR should be used for more specific or dedicated tasks. Second, is that current XML-IR systems are not responsive to the users' structural constraints. It is important to note that these four reasons are not independent of each other. In particular, it should be straightforward to create useful XML-IR queries once a appropriate task is defined and a suitable collection is discovered; and, once these three problems have been solved, responsive XML-IR systems should be a natural by-product. Below outlines two possible tasks that may be more suitable to XML-IR.

The first possible task is specific search task. In this scenario the user's information need would be very specific and therefore, only a small number of results would be relevant. Optimally the user would have a high domain knowledge of the collection, which would have a high number of semantic tags. An example collection for this task could be a travel guide, such as the Lonely Planet, since its domain is well known and semantically significant. An example information need could be "*Find all the Hotels in New York that are rated 3 or more stars*". Naturally, this information need is very specific, therefore, there would only be a small number of items would be relevant. Furthermore, the information need lends itself to a semantically tagged collection, since only *hotel* elements would be relevant and not other elements, such as *restaurants*.

The second possible task involves exploring user needs. This task is based upon the fact that, while, structure has not being found to benefit users, it is known that the information needs of users change over the information seeking session [14]. However, the system oriented evaluation conducted by organisation such as INEX lacks is unable to encapsulate how the users' needs evolve. But, an interactive user experiment would be able to capture how users' evolve, and how

this evolving user needs affects the use of structure in the users' queries. Therefore, it one could assume that at the beginning of the search session when the information needs are vague and poorly defined, that keywords satisfy their user need; however, as the information seeking session progresses and the user needs become better defined, then adding structure to their queries would better fulfil their information needs.

4.2 Which Structured Interface is Best?

If it is proven that the addition of structures aid retrieval, then the next question is: which structured interface best fulfils the information needs of users. This will of course require more user based studies in the style of Zwol et al. [7]. However, it would be suitable for future research to include a more diverse pool of participants than just computer science studies to gain a closer indication on how typical users would formulate XML-IR queries.

An alternative approach would be to explore how users interact with an operational XML-IR system. This study would take place over a longer time span, for example 12 – 18 months. This would provide many observations on the behaviour of users with respect to query formation. For instance, the users' interface preference could depend on: their individual information need; their progress during the information seeking session or their experience using each interface over the longer time span.

5. CONCLUSION

Query formation is an important aspect of XML information retrieval. This paper outlined four query formation interfaces. For each of the interfaces it: first, discussed in detail their strengths and weaknesses; second, presented comparative results from experiments and third proposed future research. Overall, the paper highlighted the necessity for an effective query formation interface if the use of XML-IR systems is to become more widespread.

REFERENCES

[1] Fuhr N., Lalmas M., Malik S. and Kazai G. (eds) (2006) Advances in XML Information Retrieval and Evaluation, 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005, Dagstuhl Castle, Germany, November 28 – 30, 2005, Revised Selected Papers, Volume 3977 of Lecture Notes in Computer Science, Springer.

[2] Fuhr N., Lalmas M., Malik S. and Szlávik Z. (eds) (2005) Advances in XML Information Retrieval, Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004, Dagstuhl Castle, Germany, December 6 – 8, 2004, Revised Selected Papers, Volume 3493 of Lecture Notes in Computer Science, Springer.

[3] Clark C. and DeRose S. (1999) XML Path Language (XPath) 1.0, W3C Recommendation.

[4] O'Keefe R. and Trotman A. (2003) The Simplest Query Language That Could Possibly Work. In Fuhr N., Lalmas M., Malik S. (eds), *INEX 2003 Workshop Proceedings, December 15 – 17, Schloss Dagstuhl, International Conference and Research Centre for Computer Science*.

[5] Trotman A. and Sigurbjörnsson B. (2005) Narrowed Extended XPath I (NEXI).In Fuhr et al. [2],16 – 40.

[6] Trotman A. and Sigurbjörnsson B. (2005) NEXI: Now and NEXT. In Fuhr et al. [2], 41 – 53.

[7] Zowl, R, Bass, J., van Oostendorp, H., Wiering, F (2005) Query Formulation for XML Retrieval with Bricks. In Fuhr, N., Lamas, M., Trotman, A. (eds), *Proceedings of INEX 2005 Workshop on Element Retrieval Methodology vol 2,* Glasgow, Scotland, 75 – 83.

[8] Woodley A. and Geva S. (2005) NLPX at INEX 2005. In Fuhr et al. [2]. 358 – 372.

[9] Geva S. GPX - Gardens Point XML-IR at INEX 2005. In Fuhr et al. [2]. 240 – 253.

[10] Trotman A. and Lalmas M. (2006) Why Structural Hints in Queries do not Help XML-Retrieval. In SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development Information retrieval, New York, NY, USA, 2006. ACM Press. 711 – 712.

[11] Cleverdon, C. (1967) The Cranfield tests on English language devices. In *Aslib Proceedings 19*(6).

[12] Borlund, P. and Ingwersen, P. The Development of a Method for the Evaluation of Interactive Information Retrieval Systems. In *Journal of Documentation*, (53)3, 1997, 225-250.

[13] Woodley, A., Geva S. Edwards S. L.. (2006) Comparing XML-IR Query Formation Interfaces In Australian Journal of Intelligent Information Processing Systems, Vol 9, No. 2 64-71.

[14] Spink A. Multiple Search Sessions Model of End-User Behaviour: An Exploratory Study. In *JASIS 47*(8): 603-609.