XML Element Retrieval and Heterogeneous Retrieval: In Pursuit of the Impossible?

Ray R. Larson
School of Information Management and Systems
University of California, Berkeley
Berkeley, California, USA, 94720-4600
ray@sherlock.berkeley.edu

ABSTRACT

This short position paper discusses the issues arising when the expectations of element retrieval are applied to heterogeneous document collections. One assumption of element retrieval strategies is that it is actually possible for searchers to specify the elements to be retrieved. As collections include an ever-increasing number of XML document types with varying schemas or DTDs, this knowledge cannot be expected on the part of searcher (unless one supposes the searcher to be omniscient), and in any case the complexity of queries must also grow monotonically with the number of types, making it increasingly difficult for the searcher to construct an element-oriented query.

Keywords

Information Retrieval, Heterogeneous Search, XML Element Retrieval

1. INTRODUCTION

In the 2004 INEX evaluation a heterogeneous track was introduced that attempted to use a combination of existing INEX topics, and introduced a set of new topics as well. Most of the following discussion in this section is based directly on our Hetergeneous track description on the INEX 2005 web site. Following the description of the track and its tasks we will further discuss the issues arising from the assumptions of element-oriented retrieval in heterogeneous collections.

1.1 Heterogenous Collections Track Motivation

The primary INEX test collection is based on a single DTD. In practical environments, such a restriction will hold in rare cases only. Instead, most XML collections will consist of documents from different sources, and thus with different DTDs or Schemas. In addition, distributed systems (federations or peer-to-peer systems), where each node manages a different type of collection will need to be searched and the results combined. So a heterogeneous collection poses a number of challenges for XML retrieval, including:

1. For content-oriented queries, most current approaches use the DTD for defining elements that would form reasonable answers. In heterogeneous collections, DTD-independent methods need to be developed.
2. For content and structure queries, there is the added problem of mapping structural conditions from one DTD or Schema onto other (possibly unknown) DTDs and Schemas. Methods from federated databases could be applied here, where schema mappings between the different DTDs are defined manually. However, for a larger number of DTDs, automatic methods must be developed, e.g. based on ontologies. The goal of an INEX track on heterogeneous collections is to set up such a test collection, and investigate the new challenges posed by such a setting.

The INEX Heterogeneous track is intended to explore the following research questions:

1. For content-oriented queries, what methods are possible for determining which elements contain reasonable answers? Are pure statistical methods appropriate, or are ontology-based approaches also helpful?
2. What methods can be used to map structural criteria onto other DTDs?
3. Should mappings focus on element names only, or also deal with element content or semantics?
4. What are appropriate evaluation criteria for heterogeneous collections?

Truly heterogeneous collections will be diverse not only in structure, but also in content, themes, sources and motivations. In the 2004 INEX, the heterogeneous track was primarily an exploration of the implementation issues and the questions of this research space. This year we intend to expand both the number and diversity of the collections to be used. The primary focus for 2005 will still be on the construction of an appropriate test collection, and on appropriate tools for evaluation of heterogeneous retrieval. Of equal importance is the exploration of the research questions outlined above.

In INEX 2004, the primary effort in the heterogeneous collection track was focussed on the following tasks:

1. Creation of a heterogeneous test collection.

2. Retrieval experiments with a small number of both CO and CAS queries.

3. Qualitative (rather than quantitative) analysis of the results.

In the following, we discuss each of these in more detail.

1.2 Testbed creation

The INEX 2004 Heterogeneous collection was based on the existing INEX collection and it retained the same topical focus (Computer Science) for additional collections contributed for the track. These collections were:

- The INEX IEEE collection with 12107 fulltext journal articles from IEEE computer science journals.
- The 6 new collections were added that were related to computer science, including:
  - Berkeley (Library catalog entries for CS literature): 12800 items
  - CompuScience (Bibliographic entries from the Computer Science database of FIZ Karlsruhe): 250987 items.
  - bibdbpub (BibTeX converted to XML by the IS group at University of Duisburg-Essen): 3465 items.
  - dblp (Bibliographic entries from the Digital Bibliography & Library Project in Trier): 501101 items.
  - hcibib (Human-Computer Interaction Resources, bibliography from www.hcibib.org): 26402 items.
  - qmuldcsdpub (Publications database of QMUL Department of Computer Science): 2024 items.

For 2005 we are intending to add more collections from more diverse topical areas, including the specialized databases being used for other INEX tracks such as the Multimedia track. Our goal is to have approximately 20 collections this year. As the above descriptions indicate, the content of the 2004 heterogeneous collections was almost exclusively bibliographic entries, and therefore had a fairly strong common semantics (e.g. Authors, Titles, etc.). An additional goal this year is to provide a wider and more varied set of collections with differing structures and semantics.

1.3 Retrieval experiments

For 2004 the heterogeneous collection was from the same application domain, so we were able to use some of the same topics formulated for the standard INEX tasks. Some preliminary work on new types of CAS queries which were intended to express their structural conditions in a collection-neutral way or as a (sub)collection-specific query (which was then processed on other sub-collections as well).

For 2005 we hope to create queries that take better advantage of the diverse contents of the new collections. This will, of course, be highly dependent on the collections that are made available for this year.

In the first year of the track, no real quantitative evaluation was attempted (in fact attempts to conduct such evaluation revealed other difficulties in dealing with diverse collections and DTDs, such as making the INEX evaluation tool work in a heterogeneous environment). Instead, track participants were asked to analyse their results in a qualitative way and start discussion about possible quantitative evaluation criteria, and tools, for following years.

What we have discovered in the Heterogeneous track is that there are many issues and problems in dealing with such collections and still being able to perform the same kind of element-oriented retrieval that is the mainstay of the main INEX adhoc retrieval evaluation. In the following section we will discuss one attempt to search across collection (Berkeley's heterogeneous track runs) and the issues that arose in attempting to set up a system to search multiple diverse XML structures.

Collection	Author tag	Title tag	Abstract tag
INEX	fm/au	fm/tig/atl	fm/abs
Berkeley	Fld100 Fld700	Fld245	Fld500
Compuscience	author	title	abstract
bibdbpub	author altauthor	title	abstract
dblp	author editor	title booktitle	*none*
hcibib	author	title	abstract
qmulcsdpub	AUTHOR EDITOR	TITLE	ABSTRACT

Table 1: Tags used for particular element types in the Heterogeneous collections.

2. LESSONS AND ISSUES FROM THE 2004 HETEROGENEOUS TRACK

Because the Hetergeneous Track for INEX 2004 was attempting to test the ability to perform searches across multiple XML collections with different structures and contents we employed ideas originally developed for distributed search protocols like Z39.50 and the more recent SRW[1, 3]. The concepts and issues involved in setting up a system for the INEX Heterogeneous Track are remarkably similar to the issues that have been explored for many years in distributed IR experiments (see, for example, [4, 5]). In the latter paper we noted:

> Users of the World Wide Web (WWW) have become familiar with and, in most cases, dependent on the ability to conduct simple searches that rely on information in databases built from billions of web pages harvested from millions of HTTP servers around the world. But this "visible" web harvested by services such as Google and Inktomi is, for many of these servers, only a small fraction of the total information on a particular web site. Behind the myriad "search pages" on many web sites are the underlying databases that support queries on those pages

and the software that constructs pages on demand from their content.

This huge set of databases make up the content of today's digital libraries and has been called collectively the "Deep Web". Estimates of the size of the Deep Web place it at over 7500 Terabytes of information [7]. As increasing numbers of digital libraries around the world make their databases available through protocols such as OAI or Z39.50 the problem arises of determining, for any given query, which of these databases are likely to contain information of interest to a world-wide population of potential users. Certainly one goal must be to aid information seekers in identifying the digital libraries that are pertinent to their needs regardless of whether the desired resources are part of the visible web or the deep web.

However, currently information seekers must rely on the search engines of the visible web to bring them to the search portals of these "Deep Web' databases, where they then must submit a new search that will (it is hoped) obtain results containing information that will satisfy their original need or desire. Today's searcher, therefore, must learn how to search and navigate not only the visible web search engines, but also the differing and often contradictory search mechanisms of the underlying Deep Web databases once those have been identified. The first challenge in exploiting the Deep Web is to decide which of these myriad databases is likely to contain the information that will meet the searcher's needs. Only then can come the challenge of how to mix, match, and combine one or more search engines for diverse digital libraries for any given inquiry, and also how to navigate through the complexities of largely incompatible protocols, metadata, and content structure and representation.

Buckland and Plaunt[2] have pointed out, searching for recorded knowledge in a distributed digital library environment involves three types of selection:

1. Selecting which library (repository) to look in;
2. Selecting which document(s) within a library to look at; and
3. Selecting fragments of data (text, numeric data, images) from within a document.

The databases of the "Deep Web" are being created in XML in many cases (or in some cases they are created in another form, such as a relational database, which is then exported as XML). The issues that arise in searching the "Deep Web" are the same issues raised by the INEX Heterogeneous track. As noted previously, truly heterogeneous collections (like the "Deep Web") will be diverse not only in structure, but also in content, themes, sources and motivations. As Table 1 shows for a few elements, the collections used in the INEX 2004 Heterogeneous track in many cases tended to use the same names for those elements included in the database, with only a few exceptions. Of course, Table 1 doesn't include all of the elements for any of the collections (the numbers of distinct elements ranged from a dozen to hundreds). In most cases each collection had elements that were not shared by any other collection.

One approach to the Heterogeneous Track is to use index mappings for each of the collections focussing on commonalities like the elements shown in Table 1. This index mapping feature was originally developed as part of support in the system for IR protocols like Z39.50. In effect, each collection can be treated as a separate database with its own DTD (either supplied with the collection, or simple "flat" DTDs were generated for those collections lacking them).

One of the issues that arises in this is that of identifying relevant elements from the different collections. The collections in most cases consisted of a single XML "document", (including one of the databases where that single document was 217Mb in size). Obviously, specifying the entire collection is not a reasonable result. This raises another issue of how to identify particular collections or databases in a heterogeneous setting, and whether the identification should be part of the element description. For example, should XPATH element identification be extended to include a URN part to uniquely identify the database/collections as a prefix to the XPATH for the individual element. (That is, should we be using XPointer to specify results, and if so, should we permit identification of elements or section using XPointer ranges?). This also assumes that the collections are maintained in their original forms by the participants, which is probably not the case.

3. DISCUSSION AND CONCLUSION

Many issues arise when the expectations of element retrieval are applied to heterogeneous document collections. A primary assumption of element retrieval strategies is that it is actually possible for searchers to specify the elements to be retrieved. As collections include an ever-increasing number of XML document types with varying schemas or DTDs, this knowledge cannot be expected on the part of searcher (unless one supposes the searcher to be omniscient). Approaches that map the collection structure and elements to some common standard (as described above) is another situation where increasing the number and diversity of collections leads to an intractable complexity of mappings (but where the burden is placed on the designer/developer of the search system instead of the searcher).

In the case of previous IR protocols for distributed search, such as Z39.50, the responsibility for creating the mappings from the canonical index representations to the particular elements of a collection was placed on the database designer (or the search system designer for a given system). Thus, responsibility for knowing how the elements of a particular collection or database correspond to the canonical search elements of the protocol was placed on those most likely to know and understand the particular database. When this responsibility is shifted to the searcher (or the designer of client systems for the searcher) the situation soon becomes intractable as the complexity of queries increases monotonically with the number of database or collections, making it

increasingly difficult for the searcher (or search client system) to construct an element-oriented query.

Is there a simple solution to these problems? One *possible* approach is to follow the example of previous IR protocols and establish a canonical set of generic "meta-elements" and make it the responsibility of the database provider to define the mapping between the meta-elements and the actual elements of the particular collection. This kind of solution must focus on the semantics of the particular type of documents that are, or might be, included in a searchable collection. In the case of most IR research there is an assumption that the items to be retrieved are usually "document-like objects" that are electronic analogues of printed documents, thus when the diversity of different possible "documents" is considered (ranging from short documents like bills of sale to books or collections of other documents) the scale of the problem becomes apparent. Metadata systems like the Dublin Core were designed to accommodate a wide variety of "document-like objects" starting with a simple set of 15 basic metadata elements that are the most common in description of documents. The elements (form of the names is from the OAI-MHP XML Schema for Dublin core) are:

1. title: The title or name of the object.
2. creator: The person or organization responsible for creation of the object.
3. subject: A topical description of the object.
4. description: A more detailed description of the object.
5. contributor: Additional persons or organizations involved in the creation or production of the object.
6. publisher: Person or organization that is making the object available.
7. date: Date that the object was created (or published).
8. type: Genre or type of object.
9. format: Physical or electronic format (could potentially be a reference for the object Schema or DTD).
10. identifier: URN or URL for the object.
11. source: If the object is derived from another object (such as a translation of another object) this element is a reference for the original.
12. language: The language(s) of the object.
13. relation: Relationships between the object and other objects.
14. coverage: Time ranges and/or geographic extents of the object.
15. rights: Rights information (copyright, etc.)

All of the Dublic Core elements can be repeated any number of times, and all are optional. Heterogeneous query specifications potentially could be framed in the context of Dublin Core, and then individual mappings for each collection from the DC elements to the actual elements could be generated. However, this is obviously not an automatic process, and it requires that the database designer knows how the DC elements are expressed in the particular database. It is, however, a less complex problem than that of constructing queries to access each unique DTD or Schema in a heterogeneous collection. As one reviewer pointed out, there can still be problems with more complex DTDs where the relationships between elements may need more complex relational mapping ("For example one DTD can have <book> and all <author>'s as children while another DTD can have an <author> and all her <book>'s as children. This requires not only name mapping but also relation mapping.")

In summation we might suggest another "Postulate of Impotence" like those suggested by Swanson[6]:

> PI10: You can either have heterogeneous retrieval, or precise element specifications in queries, but you cannot have both simultaneously.

NOTE: this paper is intended to start discussion of the issues and problems faced by heterogeneous retrieval when combined with element retrieval. I hope to provoke thought on the topic, and the statements above are aimed at such provocation.

4. REFERENCES

[1] ANSI/NISO. *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification (ANSI/NISO Z39.50-2003).* NISO, Bethesda, MD, 2003.

[2] M. K. Buckland and C. Plaunt. Selecting libraries, selecting documents, selecting data. In *Proceedings of the International Symposium on Research, Development & Practice in Digital Libraries 1997, ISDL 97, Nov. 18-21, 1997, Tsukuba, Japan*, pages 85–91, Japan, 1997. University of Library and Information Science.

[3] R. Denenberg and R. Sanderson. SRW - Search/Retrieve Web Service. Library of Congress: Available as *http://www.loc.gov/srw/*, 2004.

[4] R. R. Larson. A logistic regression approach to distributed IR. In *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002, Tampere, Finland*, pages 399–400. ACM, 2002.

[5] R. R. Larson. Distributed IR for digital libraries. In *Research and Advanced Technology for Digital Libraries (ECDL 2003)*, pages 487–498. Springer (LNCS #2769), 2003.

[6] D. R. Swanson. Information retrieval and the future of an illusion. *Journal of the American Society for Information Science*, 39(2):92–98, 1988.

[7] H. Varian and P. Lyman. How much information? Available as *http://sims.berkeley.edu/research/projects/how-much-info/.*, 2002.