Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology,

Held in Seattle, Washington, USA,

10 August 2006.

Edited by
Andrew Trotman
and
Shlomo Geva

Proceedings of the
SIGIR 2006 Workshop on XML Element Retrieval Methodology,
Held in Seattle, Washington, USA,
10 August 2006.

Published by:
Department of Computer Science,
University of Otago,
PO Box 56,
Dunedin,
New Zealand.

Editors:
Andrew Trotman
and
Shlomo Geva

ISBN 0-473-11227-2

http://www.cs.otago.ac.nz/sigirmw/

Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology, Held in Seattle, Washington, USA, 10 August 2006.

Preface

These proceedings contain the papers of the SIGIR 2006 Workshop on XML Element Retrieval Methodology held in Seattle, Washington, USA on 10th August 2006. Seven papers were selected by the program committee from eleven submissions (64% acceptance rate). Each paper was reviewed by two members of the program committee.

When reading this volume it is necessary to keep in mind that these papers represent the opinions of the authors (who are trying to stimulate debate). It is the combination of these papers and the debate that is will make the workshop a success.

We would like to thank the ACM and SIGIR for hosing the workshop. Thanks also go to the program committee, the paper authors, and the participants, for without these people there would be no workshop.

Andrew Trotman
Shlomo Geva

Organizers

Andrew Trotman	University of Otago (New Zealand)
Shlomo Geva	Queensland University of Technology (Australia)

Program Committee

Shlomo Geva	Queensland University of Technology (Australia)
Jaap Kamps	University of Amsterdam (The Netherlands)
Mounia Lalmas	Queen Mary University of London (UK)
Birger Larsen	Royal School of Library and Information Science (Denmark)
Jovan Pehcevski	RMIT University (Australia)
Benjamin Piwowarski	University of Chile (Chile)
Mark Sanderson	University of Sheffield (UK)
Andrew Trotman	University of Otago (New Zealand)
Ellen Voorhees	NIST (USA)
Arjen de Vries	Centrum voor Wiskunde en Informatica (The Netherlands)
Ross Wilkinson	CSIRO (Australia)

Contents

Element Retrieval in Digital Libraries: Reality Check 1
Philipp Dopichaj
University of Kaiserslautern

XOR – XML Oriented Retrieval Language 5
Shlomo Geva[1], Marcus Hassler[2], Xavier Tannier[3]
[1]Queensland University of Technology, [2]Universität Klagenfurt, [3]Ecole Nationale Supérieure des Mines

Understanding Differences between Search Requests in XML Element Retrieval 13
Jaap Kamps[1], Birger Larsen[2]
[1]University of Amsterdam, [2]Royal School of Library and Information Science

What does Shakespeare have to do with INEX? 20
Gabriella Kazai[1], Elham Ashoori[2]
[1]Microsoft Research Cambridge, [2]Queen Mary, University of London

Designing User Studies for XML Retrieval 28
Miro Lehtonen
University of Helsinki

Relevance in XML retrieval: the user perspective 35
Jovan Pehcevski
RMIT University

Passage Retrieval and other XML-Retrieval Tasks 43
Andrew Trotman[1], Shlomo Geva[2]
[1]University of Otago, [2]Queensland University of Technology

Element Retrieval in Digital Libraries: Reality Check

Philipp Dopichaj
dopichaj@informatik.uni-kl.de
University of Kaiserslautern, AG DBIS
Gottlieb-Daimler-Str.
67663 Kaiserslautern

ABSTRACT

Although research on XML element retrieval is steadily gaining popularity, it is not clear if and in what form element retrieval can be useful in real-world scenarios. In this paper, we compare the XML element retrieval models used in the INEX workshop with the search interfaces of two online digital library services. We demonstrate that element retrieval is indeed useful for digital libraries and that there is a lot of room for improvements in this field.

1. INTRODUCTION

The Initiative for the Evaluation of XML Retrieval (INEX) provides the infrastructure for conducting (XML) element retrieval experiments [2, 1]. So far, there has been no consensus about what a real-world application of element retrieval might look like, which was identified as a major obstacle to realistic experiments in this area [3]. In this paper, we try to address this by looking at two commercial online digital library systems that provide search functions similar to those used at the INEX workshops.

We focus on two online library services that offer full-text search for their online books, Books24x7[1] (launched in 1999) and Safari[2] (launched in 2001).[3] Each of them offers access to several thousand technical books through a web interface. The sheer amount of information necessitates good search interfaces so that the users can find relevant books or sections without problems.

In Section 2, we look at the ways in which these library services support different search models and contrast them to what is done at INEX. Section 3 addresses some aspects that are of relevance to INEX user models.

2. REAL-WORLD SEARCH INTERFACES

Both of the library services provide search interfaces that resemble the retrieval tasks at INEX at least to some extent. We first look at the result views that group by documents ("Relevant in Context" or "Fetch and Browse" in INEX terminology) or by sections (similar to "Thorough" retrieval in INEX) and finally examine to what extent structural queries are supported.

2.1 Results Grouped by Document

In traditional (flat) information retrieval, results are typically presented as a list of matching documents. For books, this alone is not a viable option: The user also needs to know *where* in the books he can find the relevant text, so there should be further information about relevant sections. INEX offers the "Relevant in Context" task (formerly "Fetch and Browse"), where the relevant elements are first sorted according to the book's score and then (inside each book) according to the element's score, and the "Best in Context" task, where the best entry point to each document is sought.

Both Books24x7 (see Figure 1) and Safari (see Figure 2) support displaying results in this fashion: They present a list of relevant books, and for each book a list of the titles of the most relevant sections or chapters from that book. In contrast to the "Relevant in Context" task, where the number of elements from a given document is unlimited, only three sections are displayed for each book, so this is probably more comparable to the "Best in Context" task. The user then has the option of navigating to a certain book, or directly to a section from that book. In both services, the in-book results can overlap, that is, it is possible that both a chapter and a section from that chapter appears in the results (note that this would not be allowed in the INEX task).

Comparing the results from Safari's book-based result display to their flat results (described in the following section), the books appear to be ranked by the score of the most relevant section.

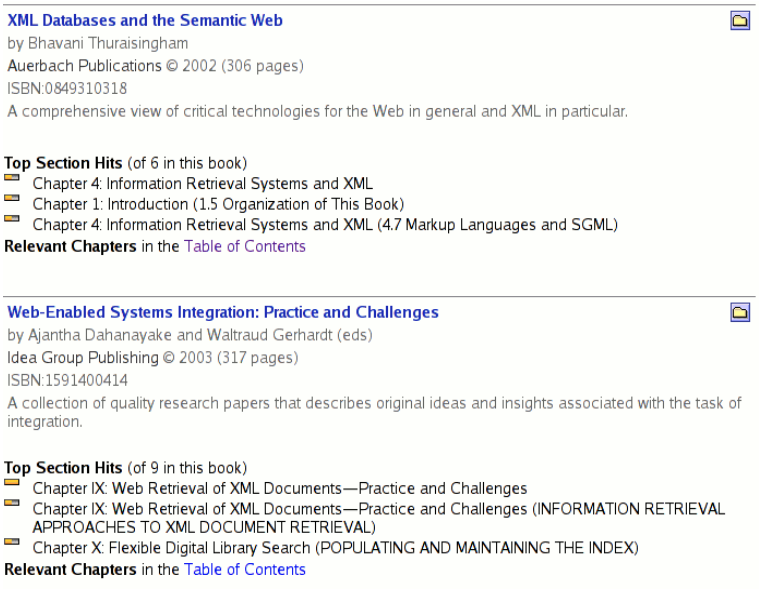
2.2 Flat Results

Another way of displaying results is presenting a flat list of relevant fragments (or snippets with pointers to the fragments). This type of result list has the advantage of being familiar to most searchers, as it is the format that most web search engines use. In contrast to web search engines, however, the results can overlap, so it is possible to have both a chapter and a section from this chapter in the results.

INEX offers two approaches addressing the issue of overlapping results: The "Thorough" task ignores the issue and

[1] see http://www.books24x7.com/
[2] see http://safari.oreilly.com/
[3] Although it is not certain that they use XML for the storage of their documents, they definitely use a semi-structured format.

SIGIR 2006 Workshop on XML Element Retrieval Methodology
August 10, 2006, Seattle, Washington, USA.



(a) Results page

(b) Table of contents of the first result with markers indicating the relavance of the chapters

Figure 1: Books24x7 search

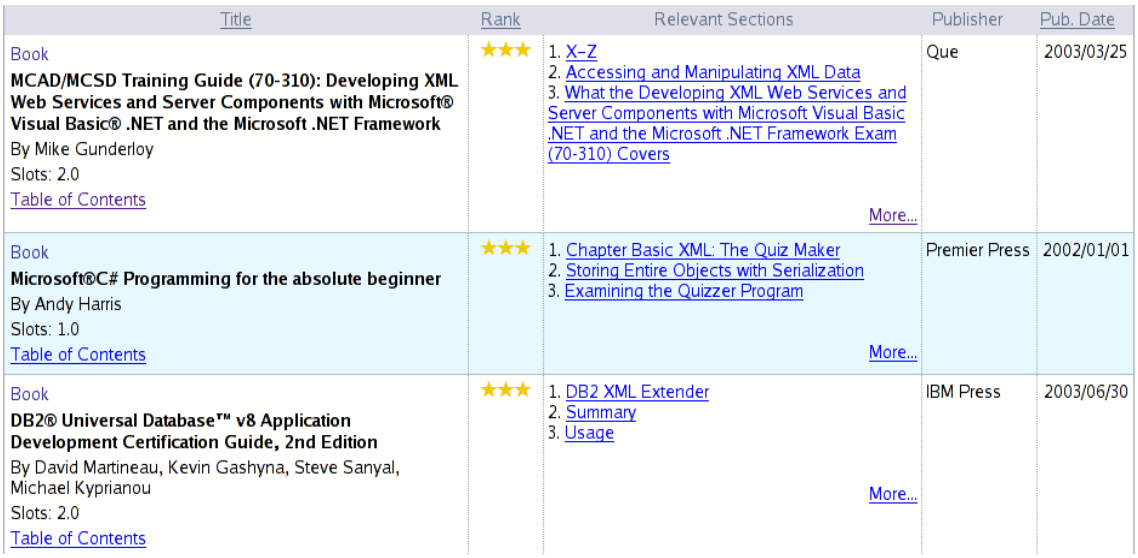


Figure 2: Safari search: View by Book

thus allows overlapping results to be displayed; it is meant to be a system-oriented task that aims at finding out whether a search engine can find *all* relevant results. The "Focused" task disallows overlapping results, so the search engine has to decide which result is more relevant to the user.

Surprisingly, Books24x7 offers no flat results, and Safari offers only a view that closely resembles the "Thorough" task, called "View by Section". This view includes a short snippet from the relevant sections, with the search terms highlighted, and a hyperlink to the complete section; see Figure 3. The search can also be restricted to a single book, so that the relevant sections inside a single book can be identified easily.

2.3 Content-and-Structure Search

One interesting research topic is whether structural hints in the query—for example, "find articles about information retrieval that cite the INEX proceedings"—help the retrieval engine. INEX uses NEXI, a special search language derived from XPath [4]; this language is not suitable for ad-hoc queries, but it can help to evaluate whether structural hints have any positive effects.

Obviously, casual users of online digital libraries do not have intricate knowledge of the internal schemas of the documents, so the library services offer only limited support for structural queries in their advanced search interfaces: You can search in meta information such as author or publisher or in book titles.

Safari's advanced search interface also offers limited content-and-structure search by offering a choice of one of the following options:

- The full text
- Code fragments only
- Section title words only
- Tips and how-tos only

These options appear to be used as retrieval hints only (vague content-and-structure search): Neither is the granularity of the retrieval results affected, nor are only sections returned that fulfil this condition.

Even this simple form of structured queries is not used as the default search interface. This might indicate that the default (content-only) search interface is sufficient for most queries and users, but that the more complex interface is needed for advanced searchers and more complex information needs.

2.4 Book Search without Element Retrieval

For comparison, we also briefly examined a search engine for books that does not use element retrieval because the books in its index are not available in a semistructured format. Google Book Search[4] differs from Safari and Books24x7 in that it does not offer access to the full text of the books it has indexed. In their own help text[5], they state:

> Google Book Search helps you discover books, not read them online. To read the whole book, we encourage you to use a "Buy this book" link to purchase it or the "Find this in a library" link to look for a local library that has it.

Along the results still under copyright, they provide links to several online book stores. As such, their search service can be seen as a means to find references to works satisfying the information need, instead of fragments that themselves satisfy the information need. Some publishers allow Google to show a few relevant pages scanned from the paper versions, with the matching terms highlighted. The pages do not necessarily correspond to logical units in the text, so it might well happen that the search term appears at the end of a page, but the relevant information is wrapped to the next page. Element retrieval has the potential to offer better results, but it cannot be used in this case because the books are not available to Google in a semi-structured format.

3. FURTHER NOTES

Apart from the search interface, several other aspects are of interest to the element retrieval community. In this section, we speculate how the different subscription models might affect the demands of the users. Next, we look at the history of Safari's search interface to show that the current interface is at least usable (unfortunately, we have been unable to reconstruct old versions of the Books24x7 interface).

3.1 Subscription Models

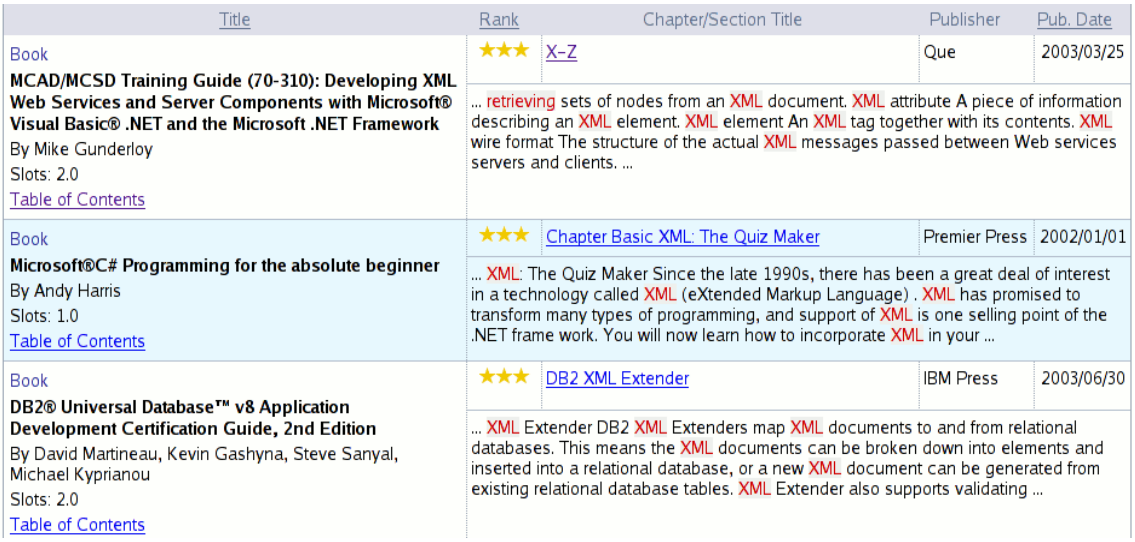
Both Books24x7 and Safari are subscription-based, but the type of subscription differs substantially: Books24x7 subscribers have full access to all books at all times. Safari users only have access to a limited number of books of their own choice at any given point in time: They have a limited number of slots on their virtual bookshelf, and once they put a book on there, it must stay there for at least a month.

These different subscription models affect the users' requirements on the search interfaces: Books24x7 users have no access restrictions, so they might well be interested in locating small, very specific parts of the books to answer the queries; diversity (results from many different books) can be helpful to get the complete picture. Safari users, on the other hand, should avoid putting books on their bookshelf that are not useful to them, so the search interface should help them find books which contain the highest amount of relevant information. Finding relevant sections for a specific query is not such a high priority here, because putting a book on one's bookshelf just for reading a single section might be wasteful. The "View by Section" feature is most probably used mainly for searching the books on one's bookshelf (to find relevant sections in the available books), or possibly to get short fragments of the texts in the result list, which is not available in the "View by Book" result list.

3.2 Development of Safari Search

We can assume that the search interfaces of the book services are demand-driven, which means that unhelpful features would be removed after some time. Thus, it is interesting to see that the search interface of Safari has been virtu-

[4]see http://books.google.com/
[5]see http://books.google.com/intl/en/googlebooks/help.html



Title	Rank	Chapter/Section Title	Publisher	Pub. Date
Book **MCAD/MCSD Training Guide (70-310): Developing XML Web Services and Server Components with Microsoft® Visual Basic® .NET and the Microsoft .NET Framework** By Mike Gunderloy Slots: 2.0 Table of Contents	★★★	X-Z	Que	2003/03/25
	... retrieving sets of nodes from an XML document. XML attribute A piece of information describing an XML element. XML element An XML tag together with its contents. XML wire format The structure of the actual XML messages passed between Web services servers and clients. ...			
Book **Microsoft®C# Programming for the absolute beginner** By Andy Harris Slots: 1.0 Table of Contents	★★★	Chapter Basic XML: The Quiz Maker	Premier Press	2002/01/01
	... XML: The Quiz Maker Since the late 1990s, there has been a great deal of interest in a technology called XML (eXtended Markup Language) . XML has promised to transform many types of programming, and support of XML is one selling point of the .NET frame work. You will now learn how to incorporate XML in your ...			
Book **DB2® Universal Database™ v8 Application Development Certification Guide, 2nd Edition** By David Martineau, Kevin Gashyna, Steve Sanyal, Michael Kyprianou Slots: 2.0 Table of Contents	★★★	DB2 XML Extender	IBM Press	2003/06/30
	... XML Extender DB2 XML Extenders map XML documents to and from relational databases. This means the XML documents can be broken down into elements and inserted into a relational database, or a new XML document can be generated from existing relational database tables. XML Extender also supports validating ...			

Figure 3: Safari search: View by Section

ally unchanged since at least October 2002, as witnessed by the Web Archive's page from October 13[6]. Even the newest changes from June 2006 do not affect the basic search model, the changes are mostly cosmetic. The most notable difference is that the default view switched from "View by Book" to "View by Section".

Unfortunately, the oldest version of the documentation is available in the Web Archive's cache from August 2002[7] lacks the relevant screen shots, so it is hard to tell what exactly was changed from this version to the next; from the textual description, it appears that a variant of the "View by Book" interface was available, whereas "View by Section" was missing. If this is the case, it may indicate that this feature was requested by users. Along with the recent change of the default view, this suggests that a flat result list is an important user interface for element retrieval.

The stability of the search interface does not imply that the current version is the best possible interface, but it does suggest that element retrieval is useful, and that there is some use to both book-based and section-based result lists.

4. DISCUSSION POINTS

We have seen that the models of element retrieval that are used in the real world do not always match the models assumed in the research community and INEX. This does not imply that one side is right and the other side is wrong; in particular, the commercial entities using element retrieval do not appear to have conducted extensive usability studies for their user interfaces. The fact that these user interfaces have been in use for several years implies that they are at least acceptable, so we can assume they are reasonable starting points for further refinements. We still need to investigate what we should adapt, and we definitely need to do more usability studies.

[6]see http://web.archive.org/web/20021209040844/safari.oreilly.com/?mode=Help

[7]see http://web.archive.org/web/20020818170606/safari.oreilly.com/mainhlp.asp?help

The following questions might be starting points for a discussion:

- Is element retrieval useful for texts of all lengths, or is it primarily useful for long texts such as books?
- Is a document-based display of results more natural than an element-based one?
- Is the "Thorough" task *really* system-oriented? Both online library services present overlapping results, so users apparently do not mind too much.
- What user models can we derive from these use cases?
- Can we cooperate with a provider of a digital library for INEX? (How do our results compare to those returned by the default search engines?)

5. REFERENCES

[1] Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Gabriella Kazai, editors. *INEX 2005 Proceedings*. Springer, 2006.

[2] Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltán Szlávik, editors. *INEX 2004 Proceedings*. Springer, 2005.

[3] Andrew Trotman. Wanted: Element retrieval users. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, 2005.

[4] Andrew Trotman and Börkur Sigurbjörnsson. Narrow extended XPath I (NEXI). In Fuhr et al. [2].

XOR - XML Oriented Retrieval Language

Shlomo Geva
Queensland University of Technology (QUT)
2 George St, Brisbane
Q 4001, Australia
s.geva@qut.edu.au

Marcus Hassler
Universität Klagenfurt
Universitätsstrae 65-67 9020
Klagenfurt
Austria
Marcus.Hassler@hekkas.com

Xavier Tannier
Ecole Nationale Supérieure des Mines
158 Cours Fauriel
42023 Saint-Etienne, France
tannier@emse.fr

ABSTRACT

The wide acceptance and rapidly growing use of XML as a standard storage and retrieval data format blurs the historical divide that exists between Information Retrieval and Database Retrieval. On the structured database retrieval side it is now possible to support highly structured access to documents using XML specific tools such as XPath, XQuery, XQL and more. On the information retrieval side it is possible to support access to the XML documents using XML specific retrieval query languages such as NEXI. None of the above are intended for end-users, but rather as enabling back-end technologies. In this paper we introduce XOR - a new XML Oriented Retrieval language that is designed to facilitate query specification with a strong IR flavour. XOR is backwards compatible with NEXI, but significantly extends its functionality overcoming many of its restrictions and limitations. While XOR itself is not an end-user tool, it is designed with the explicit goal of supporting IR, and more specifically, user oriented interfaces such as Natural Language Queries (NLQ) or interactive user interfaces. XOR provides the missing functionality that none of the existing XML retrieval tools support, and which advanced IR requires.

1. INTRODUCTION

A historical divide exists between Information Retrieval and Database Retrieval. The former is mostly concerned with text documents or web pages with minimal structure, while the later is primarily concerned with highly and strictly structured documents. XML supports the representation of all types of documents, catering for the full spectrum - from unstructured to highly structured documents. The gap between IR and traditional Database approaches is closing. Indeed, many approaches to XML-IR rely on database technology as a back-end system, rather than rely on IR specific file structures. XML retrieval tools from W3C, such as XPath [1] or XQuery [2] are highly sophisticated query languages. For information retrieval applications, XPath and XQuery are arguably completly over the top. Information retrieval queries are often loosely defined, vague, or even ambiguous. NEXI [3] is an alternative query language that was designed to drastically cut-down XPath, while being extended with explicit IR flavoured functionality and with implicit IR flavoured interpretation.

NEXI (Narrowed Extended XPath I) is a language for Information Retrieval over XML document collections (XML-IR), proposed and used by INEX - the INitiative for the Evaluation of XML Retrieval - since 2004[1]. NEXI offers a good compromise between the need to formally express structural and textual constraints on the one hand, and the ability to write IR flavoured queries on the other hand.

At the same time, since 2004, INEX ran a Natural Language Processing task. Rather than requiring participants to implement NLP based XML search engines, the main task of the NLP track is the automatic translation of an expressed natural language information need into a formal NEXI query. The automatically generated formal queries are then evaluated by a standard XML search engine that is provided by the track organizers.

The results from the first two years are very encouraging, but it appears now that NEXI specifications constrain further research and development of this approach. Indeed, an analysis of natural language queries can lead to identification of interesting features or relations between terms (or elements). But the need to use NEXI as a pivot language prevents the use of this knowledge when formulating queries. It should be noted that XPath does not support the necessary functionality either - it is not the simplification from XPath to NEXI that is the cause of the problem.

Here we introduce the XML-Oriented Retrieval (XOR) formal query language, an extension of NEXI that supports new features. The extensions are not a re-introduction of XPath features that were removed when NEXI was designed; rather, the extensions are geared towards more expressive, albeit more complex, queries; however it is primarily intended for use by automatic query generators (such as natural language interfaces or interactive user interfaces that are guided by explicit user feedback in response to clarification requests from the system). This is an important trait to notice, because XOR is a formal XML-IR language that is NOT designed for direct use by people - not even XML experts who are the users of XPath and XQuery. The

[1] http://inex.is.informatik.uni-duisburg.de/2006/

SIGIR 2006 XML Element Retrieval Methodology workshop August 10, 2006, Seattle, Washington, USA.

language is designed purely as an intermediary in IR application, and in our case specifically with NLP in mind to facilitate explicit support of natural language queries.

Nevertheless, the language can be used by end-users or expert-users if desired. It is still much simpler than XPath, XQuery, or XQL, for instance. the *XOR* language is extensible and backwards compatible with NEXI, meaning that any query written in NEXI can be successfully parsed and processed by the *XOR* parser. This is very important because it allows researchers and developers to test systems that are based on *XOR* with the datasets and assessment data and tools that were developed by the INEX initiative over the past 5 years, as well as into the future.

XOR is designed from the outset as an open ended extensible language. The support of specific functionality in *XOR* is left to the implementation of a search engine and is not part of the *XOR* parser. However, *XOR* provides a concise and simple syntax for extension. This is another feature that distinguishes *XOR* from most other XML query language specifications. For instance, if the query specifies constraints over part-of-speech (POS) tags, then it is a search engine implementation issue with respect to which POS tagger it supports; furthermore, it is even possible that the search engine will choose to ignore the POS constraints altogether.

The *XOR* parser implementation is also required to perform an additional transformation that is not commonly found when a language syntax is defined. The parser converts the query from infix notation to postfix notation (or Reverse Polish Notation). This transformation is designed to assist the development of the back-end search engines that support *XOR* and to lower the threshold to participation in eveluation forums like INEX.

In what follows we describe the *XOR* language and provide examples of the syntax and of queries. We then describe the query transformation to Reverse Polish Notation (RPN), and provide some early results that were obtained with *XOR* over the INEX repository of XML documents, topics, and relevance assessments. The appendix provides the BNF diagrams of *XOR*.

2. NEXI

NEXI is a formal query language that is based on XPath. It has been designed to allow a simple but efficient representation of information needs for XML information retrieval.

The syntax of NEXI is similar to XPath, however, it only uses the descendant axis step, and extends XPath by incorporating an *"about"* clause to provide IR flavour to queries. NEXI's syntax is:

```
//A[about(.//B,C)]
```

where A is the context path, B is the relative path and C is the content requirement.

It is possible for a single NEXI query to contain more than one information request.[2]. Therefore the query *"Return paragraphs about watermarking in article containing a paragraph about data embedding"* can be represented as follows:

```
//article[about(.//p, "data embedding")]
       //p[about(.,watermarking)]
```

The query contains two information requests (or sub-topics):

```
//article//p[about(.,watermarking)]
```

And:

```
//article[about(.//p, "data embedding")]
```

In NEXI each information request is specified by an *'about'* clause. However, elements matching the rightmost *'about'* clause, here the first request, are returned to the user. INEX refers to these requests and elements as "target requests" and "target elements". Elements that match other *"about"* clauses, here the second request, are used to support the return elements in ranking. We refer to these requests and elements as "support requests" and "support elements". In order to be valid, each NEXI query must have at least one target request, along with any number of support requests.

While NEXI does support the specification of more complex queries using parenthesis and the boolean operators AND and OR, the interpretation of such features is not strict. In standard IR query terms are regarded as retrieval hints, and therefore query expansion is allowed (even expected). In the same manner, in the interpretation of NEXI, all structural specifications are also taken merely as hints. The NEXI expression is not regarded as deterministic and it is left to the search engine to interpret it. For instance, the AND operator is commonly evaluated with OR semantics [4] [5] [6] [7] by search engines. For sure, any system that implements a simple keyword search, such as represented by the title element of an INEX topic, effectively performs an implicit OR (because the title element contains all keywords that appear in the castitle element, but the structure and boolean conditions are lost.) A common approach to the implementation of AND and OR is to use the fuzzy-like operators whereby scores are multiplied (AND) or added (OR). The AND operator is no longer interpreted striclty and takes on an OR flavour.

3. LIMITATIONS OF NEXI

Translation of natural language queries into a formal language like NEXI presents some limitations, mainly due to the fact that the natural language preprocessor cannot specify certain constraints to the retrieval system. The formal language, if not specifically designed with this aim, is pivotal in preventing helpful "communication" between both systems.

[2]NEXI does support multiple path specification whereby a query can be be return multiple elemtn types. It does not however provide explicit support to multiple distinct search requests

For example, it is not possible to consider the following features within single NEXI queries[3]:

- NEXI allows only single queries (with only one target element). This is very limiting when trying to express the same information need in several ways. For example, suppose that we are seeking information concerning Einstein's 1905 article about electrodynamics. We may directly look for this article:

```
//article[.//year = 1905
         AND about(.//author, Einstein)
         AND about(.//*, electrodynamics)]
```

  But we could also want to see some of the many articles that explain or discuss this article...

```
//article[about(., Einstein article 1905)
         AND about(., electrodynamics)
```

  ... or 1900's articles on this subject to have an idea of the state of the art at this period:

```
//article[about(.//year, 1900)
         AND about(., electrodynamics)
```

  Thus a single information need (*"I want to understand everything about this famous Einstein's 1905 article about electrodynamics"*) may be represented in at least three different complementary queries.

- NEXI handles only the `'about'` predicate, while others could be of interest for the search process. For example, with NEXI, whether the terms should be matched strictly or with potential syntactic, semantic variations like stemming or term expansion, is up to the back-end system. An NLP system cannot intervene and specify the desired interpretation even if it is available.

- NEXI does not allow the user to refer to more than one article. Many requests in INEX concern bibliographic references, but search engines are not explicitly asked to look at referred articles (if available) and, to date, all implementations of NEXI are restricted to search the references section titles - a very narrow window to referenced articles indeed.

  e.g.: *"Find bibliographic references that are about text categorisation where Support Vector Machines (SVM) categoriser is used."* (Topic 136), which is translated in NEXI as:

```
//bib[about(., text categorisation)
  AND about(., "Support Vector Machines" SVM)]
```

- Finally, NEXI lacks a way to express any additional desired features concerning the tags or the search terms[4]. Among these features, we can cite the minimum / maximum size of the element, the type of interpretation (strict or vague), part-of-speech, word case, language, or any other useful feature imagined. *XOR* is designed to support an open ended set of selection qualifiers.

NEXI has been designed as a reduction of XPath to handle only information retrieval oriented features. We think that it is now time to extend NEXI with more powerful IR oriented features.

4. XOR LANGUAGE SPECIFICATION

The *XOR* language is almost entirely compatible with the previously defined NEXI specification [3]. The extensions mainly concern the (automatic) query formulation capabilities of combining several queries into a single query, more elaborate specifications of paths and terms, and a larger set of matching predicates for specific information needs.

4.1 Negation operator

The sign `'-'` of NEXI is supported, but it is sufficient to clearly express that a term must not appear in returned elements. We propose the negation of the *about* clause, semantically more adequate, and syntactically more powerful.

For example, a query like *I am not looking for devices for computer-based training.*[5] does not mean that terms *"devices"* and *"computer-based training"* must not appear in elements (as would NEXI by `-devices -"computer-based training"`), but that they must not be found *together*[6]. An expression like the following better suits the information need:

```
NOT about(., devices "computer-based training")
```

4.2 Logical operators for queries

We justified in the introduction the utility of allowing multiple expressions in the same query. XOR enables the specification of a set of CAS queries combined with boolean operators.

This step includes strict bracketing to avoid ambiguities and may contain negations (using `NOT`). An examples of syntactically correct *XOR* query, that is not valid NEXI is

```
(//A[about(.,B)] AND //A[about(.,D)]) OR
(//A[about(.,B)] AND NOT (//A[about(.,C)] OR
                  //A[about(.,D)]))
```

[3]In addition to this list, NEXI is not designed to deal with many database-oriented constraints, particularly when dealing with strongly typed elements, but we are not concerned with this here

[4]See Sigurbjornsson and Trotman "Queries: INEX 2003 working group report" where they state "There already exist two data types, numeric and string. This is anticipated to expand in the future to include names, units of measure, and even geographic locations. The language must be extensible to include these at a future date."

[5]From Topic 196, INEX 2004

[6]The complete query is about education problems raised by computer-based training, and then the term *"computer-based training"* is found in a relevant element.

Thus, the example of the 1905 Einstein article can be simply translated into

```
//article[about(.//year,1905)
         AND about(.//author, Einstein)
         AND about(.//*, electrodynamics)]
  AND
//article[about(., Einstein article 1905)
         AND about(., electrodynamics)
  AND
//article[about(.//year, 1900)
         AND about(., electrodynamics)
```

In practice, as search engines return a list of elements that are independent from each other, and not groups of elements, the operators `'AND'` et `'OR'` usually have exactly the same semantics[7]. It remains as a challenge for the search engine to merge (or fuse) the results of the distinct queries which may possess completely different statistics.

4.3 Path extensions

XOR supports the specification of additional path constraints. This information is optional and expressed within curly brackets as a set of {`key:value`} pairs. Thus, this information can easily be extended to include further types of matching. Currently, the following `key:value` pairs are supported in our back-end, but more are possible. Consider the following examples:

`match:strict|vague` Specifies the kind of structural requirement matching, influencing the result set and ranking. For instance:

```
//article[about(.//year{match:strict},1905)
         AND about(.//author, Einstein)
         AND about(.//*, electrodynamics)]
```

Here insisting that "1905" must be found in an element tagged as "year". Or:

```
//section{match:vague}[about(.,cars)]
```

Here indicating that a section-like element is required. The default is implementation defined, but vague seems most appropriate in the IR context.

Besides additional types of paths, wildcards are allowed to specify node names. The following patterns are valid:

`//*` e.g., all node names, //xyz, //jim

`//node*` e.g., //node6, //nodename

`//*node` e.g., //mynode, //this_test_node

`//*node*` e.g., //mynodeextension, //the_node_quantifier

[7]For example, asking for *sections and paragraphs* or *sections or paragraphs* will lead in both cases to a ranked list of sections and paragraphs.

The intended use of this feature is in situation where the DTD is not available, or too complex to enumerate possible matches. Far from being the exception, this may well be the norm, particularly with private or dynamic collections. The use of wildcards is nevertheless resting on the assumption that meaningful tag names are used in the collection (in a natural language). The Wikipedia XML collection that is used by INEX in 2006 is a good example of precisely this situation.

4.4 Term extensions

For the purpose of more exact query matching *XOR* enables the addition of further information to a given term (in the same manner as to the path). Again, the additional information is optional and expressed within curly brackets as a set of `key:value` pairs. Consider the following examples:

`POS: part-of-speech` Specifies the kind of Part-Of-Speech (POS) tag.

```
//abstract{match:vague}[about(.,Go{POS:Noun})]
```

Here we are looking for the game "GO" not the verb.

`CASE: upper|lower` Specifies the case of the text - useful for acronyms for instance.

```
//section{match:vague}[about(.,AJAR{CASE:upper})]
```

Here we are looking for the acronym for "Acronyms, Jargon, Abbreviations and Rubbish", not ajar meaning slightly open.

4.5 Logical operator qualifiers

It is possible with XOR to qualify the logical operators. The interpretation of the qualifiers is again left to the search engine. For instance,

```
//article[about(.,Germany)
  AND{mode:strict} about(.,football)]//sec[about(.,Europe)]
```

Here we insist on strict interpretation of the logical AND operator.

4.6 Additional predicates

In the context of heterogeneous information needs and highly sophisticated search techniques, a single `about` predicate for matching seems too restrictive. For this reason *XOR* implements several additional predicates, having similar format to the *about*() function:

LinkTo((XLink—XPointer),keywords) matches documents that are linked to by the context element. For instance, the implementation could check that the linked-to element is *about*() the keywords.

LinkFrom() matches elements which link to the context element. For instance, the implementation could check that the linked-from element is *about*() the keywords.

Contains() This is the same as the XPath function and has a strict interpretation.

lt(), eq(),gt() for less than, equal, greater than respectively. Necessary for numeric element comparisons, or fixed format fields because XML files do very often contain both free text and strictly typed elements. The motivation for using functions rather than the traditional symbols is twofold - it keeps the language much simpler and all operators are treated uniformly.

The XOR specifications do not require the parser to check that functions actually exist. This is left to the backend search engine. So any implementation of XOR can create new functions and the parser does not get involved as long as the syntax of the function call is valid. The about() and eq() functions, for instance, are both treated identically in the syntax and both are left to the search engine to implement.

5. REVERSE POLISH NOTATION

In order to support the implementation of back-end processors, the actual XOR parser checks the validity of XOR expressions and returns a vector (of text lines) containing the translation of the expression from infix to postfix notation, or as it is often known *Reverse Polish Notation* (RPN) [8]. Each line in the RPN is a simple NEXI expression. The following example illustrates this transformation.

***XOR* Query:**

```
//article[about(.,Germany)
  AND NOT about(.,football)]//sec[about(.,Europe)]
AND
//article[about(., European union enlargement)
  AND about(.//*,German point of view)]
```

***RPN*:**

```
//article[about(.//*, German point of view)]
//article[about(., European union enlargement)]
AND
//article//sec[about(., Europe)]
//article[about(., football)]
//article[about(., Germany)]
ANDNOT
SUPPORT
AND
```

This notation should be read with the following binding:

```
//article[about(.//*, German point of view)]
//article[about(., European union
enlargement)]
AND
//article//sec[about(., Europe)]
  //article[about(., football)]
  //article[about(., Germany)]
  ANDNOT
SUPPORT
AND
```

The binary operator $SUPPORT(X, Y)$ means that the second argument (here, an article about Germany but not football) is used as a support to the selection of the target element, which is the first argument (here, a section about Europe). Implementation of AND, ANDNOT, OR, SUPPORT are up to the search engine. This is where the developers of the search engine have the freedom of interpretation - this is akin to the IR flavoured about() function in contrast to the XPath strict contains() function.
Inverted lists are generated as a product of the atomic function calls within the XOR filters, like about() or contains() etc. These atomic units are presented as separate lines (search requests) in the RPN representation. The advantage of the RPN is that it allows for unlimited nesting of parenthesis and any path expression depth (one of the limitations of NEXI). Furthermore, the RPN format lends itself to simple implementation of search algorithms by systems that are based on the processing of inverted lists, a stack, and binary or unary list operators. We were able to easily incorporate XOR into GPX, a search engine that supports NEXI queries, and which is based on inverted list processing.

5.1 Important differences

Important to stress are the following oddities to XPath, etc.

- `//*` in NEXI means any descendant node, in XOR it means the context node or any descendant node. We often find that nodes contain both direct text and descendant nodes that contain more text. So selecting //sec//*[...] means select sections or descentedants of sections that satisfy the condition. To select only descendents of section in XOR we use //sec/*[...]

- `=` in XOR the function eq() is used instead. Similarly for other comparison operators. Instead of the NEXI query:

  ```
  //section[.//year = 1905]
  ```

  in XOR we would write:

  ```
  //section[eq(.//year,1905)]
  ```

  or perhaps:

  ```
  //section[eq(.//year{match:vague},1905{match:strict})]
  ```

[8] http://en.wikipedia.org/wiki/Reverse_Polish_Notation

6. IMPLEMENTATION EXAMPLE: GPX-XOR

We have used the GPX search engine as a back end system to test the *XOR* parser. The prupose of this experiment is no more than a sanity check and an example of how *XOR* might be implemented with existing search engines, that can already process simple NEXI expressions. GPX is an XML search engine that was used at INEX in 2004 and 2005. GPX is based on inverted lists - a detailed description can be found in [4], but suffice to say that the retrieval and score calculation for elements in each search request in the *XOR* RPN expression is largely unchanged. Each of the elements in the lists is scored with a TF-IDF variant by the standard GPX algorithm. The *XOR* operators AND, OR, ANDNOT and SUPPORT were implemented as described in the following sections.

6.1 OR(X,Y)

The OR operator is a union of two inverted lists, X and Y. Items in the lists identify XML result elements by file-id, full XPath expression, and relevance score. The OR operator performs a set union whereby elements that appear in both lists are merged and their scores combined. Other elements keep their original score.

6.2 AND(X,Y)

The AND operator was optionally implemented in one of three different ways. The default option is to simply implement it as OR(X,Y). This seems to work quite well in most instances, and also on average. However, in some queries the user *really* means AND. The second option is to implement it as a strict set intersect. Only XML elements that appear in both X and Y are kept, and their scores combined. This option is too restrictive because sometimes the lists contain overlapping elements and then the relationship with respect to AND is unclear. By insisting on a strict match many relevant results are lost. The third implementation keeps overlapping nodes, combines the scores, but keeps only the largest node (deepest common ancestor). In the experiments that we report in the next section, we used the first (default) option.

6.3 ANDNOT(X,Y)

The ANDNOT operator is implemented in a straight forward manner, and we adopted the the strictest interpretation - elimination. Any node in X that has an exact match in Y is eliminated. We assume that when users just want to discourage some keyword from appearing they will use the milder *"-keyword"* form of query specification. The list X is then returned. The *XOR* parser only allows the use of the NOT operator only in conjunction with AND, that is - X AND NOT Y - hence ANDNOT.

6.4 SUPPORT(X,Y)

The SUPPORT operator takes a list of nodes in Y that provide support to the selection of nodes from list X. For instance, when we look for paragraphs about Americium in articles with abstract about the Periodic Table, the result elements are paragraphs, and paragraphs are supported by abstracts about the Periodic Table. Both the support and result elements must have a common ancestor within the document tree, so the supporting abstract must appear in the same document as the supported paragraphs. The support operator identifies for each result element in X, all the support elements in Y, and combines the scores. All the elements from X are returned but those with support have an increased score.

6.5 Preliminary Results

The conversion of our existing search engine to support *XOR* took about one day (although it took a bit longer to iron out some bugs.) We were able to test *XOR* interactively with numerous queries with very pleasing results. We also tested GPX-XOR, and the RPN approach, against the INEX 2005 tasks with the Context and Strucure (CO+S) topics. These topics were all specified in NEXI - a subset of *XOR*. Figures 1 to 3 depicts the performance of the GPX-XOR system with the three best performing official submissions in INEX 2005 in the COS.Thorough task. Each of the baseline submissions (TWENETE, QUT, IBM) produced the best result in either the Strict, Generalised, or GenLifted quantization respectively, as measured by the MAep value. The results that we obtained with *XOR* are very promising, and the performance exceeded that of the baseline submissions in all 3 cases. The uppermost curve in all figures belongs to GPX-XOR. Of course this result can only be taken as a sanity check. This is not a definitive evaluation since there is a risk of overfitting the results to assessments when experimenting (and debugging) with known qrels. Similarly good results were obtained with all the INEX tasks, when compared with the GPX (NEXI) baseline system. We will be able to test *XOR* more rigorously with unseen qrels at INEX 2006. The point that we wish to make is not the specific performance of GPX-XOR, but rather the simplicity of converting an existing NEXI search engine to *XOR* without any loss in performance.

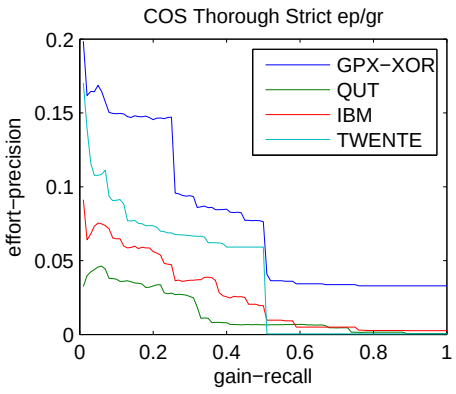


Figure 1: Strict quantization

7. CONCLUSIONS

We have presented *XOR*, a language explicitly designed to support IR in XML collections. More specifically, *XOR* was designed with the experience gained in the INEX natural language queries task, to support more elaborate search options than would be possible with NEXI. Yet, *XOR* is not extended with XPath like functionality, but rather with functionality that is IR oriented and that is not supported by existing XML search languages. More specifically, *XOR* extends selection specification of search terms, allowing for

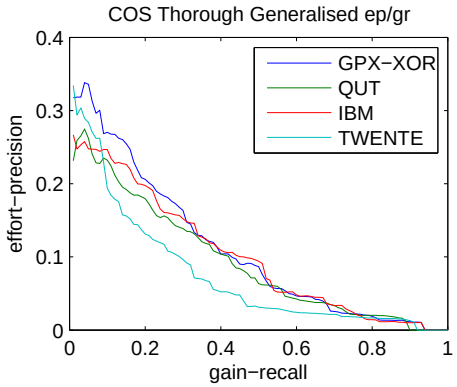


Figure 2: Generalised quantization

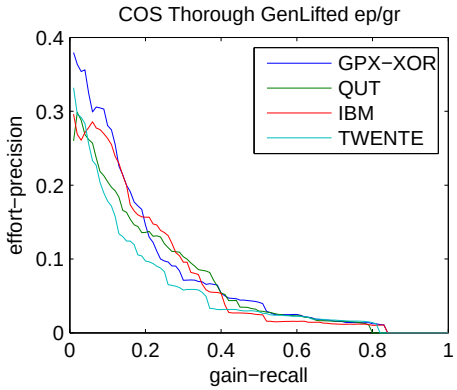


Figure 3: GenLifted quantization

more refined control of query content expansion. It also extends the selection specification of XPath expressions with wildcards, extends the allowable overall query complexity, and specifies a transformation from infix to postfix notation for easier integration into existing search engines. *XOR* is open ended and thus future work will concentrate on providing more functionality in *XOR* and on open source search engine implementation. *XOR* can support easier integration of advanced XML IR techniques. Support for *XOR* will reduce the need to develop complete search engines to implement powerful user interfaces to XML IR systems, such as natural language query interfaces.

8. REFERENCES

[1] XML Path Language (XPath) 2.0, W3C Candidate Recommendation 8 June 2006. http://www.w3.org/TR/xpath20/

[2] W3C XML Query (XQuery), XML Query is currently a W3C Candidate Recommendation. http://www.w3.org/XML/Query/

[3] A. Trotman and B. Sigurbjrnsson, Narrowed Extended XPath I (NEXI). In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlàvik, editors, *Advances in XML Information Retrieval. Third Workshop of the Initiative for the Evaluation of XML retrieval (INEX)*, volume 3493 of *Lecture Notes in Computer Science*, pages 16–40, Schloss Dagstuhl, Germany, December 6-8, 2004, 2005. Springer-Verlag, New York City, NY, USA.

[4] S. Geva, GPX - Gardens Point XML IR at INEX 2005. Proceedings of INEX 2005, Schloss Dagstuhl, Germany, November 2005, in Springer, Lecture Notes in Computer Science LNCS 2006. To Appear. Pre-proceedings - http://inex.is.informatik.uni-duisburg.de/2005/pdf/inex-2005-preproceedings.pdf

[5] V. Mihajlovi?, G. Ramrez, T. Westerveld, D. Hiemstra, H. Ernst Blok and A. P. de Vries, TIJAH Scratches INEX 2005 Vague Element Selection, Overlap, Image Search, Relevance Feedback, and Users . Proceedings of INEX 2005, Schloss Dagstuhl, Germany, November 2005, in Springer, Lecture Notes in Computer Science LNCS 2006. To Appear. Pre-proceedings - http://inex.is.informatik.uni-duisburg.de/2005/pdf/inex-2005-preproceedings.pdf

[6] P. Arvola, J. Keklinen and M. Junkkari, TRIX Experiments at INEX 2005 . Proceedings of INEX 2005, Schloss Dagstuhl, Germany, November 2005, in Springer, Lecture Notes in Computer Science LNCS 2006. To Appear. Pre-proceedings - http://inex.is.informatik.uni-duisburg.de/2005/pdf/inex-2005-preproceedings.pdf

[7] R. Schenkel and M. Theobald, Relevance Feedback for Structural Query Expansion. Proceedings of INEX 2005, Schloss Dagstuhl, Germany, November 2005, in Springer, Lecture Notes in Computer Science LNCS 2006. To Appear. Pre-proceedings - http://inex.is.informatik.uni-duisburg.de/2005/pdf/inex-2005-preproceedings.pdf

APPENDIX

A. XOR SPECIFICATIONS

```
SKIP ::= " " | "\t" | "\n" | "\r" | "\f"

OR          ::= "or" | "OR"
AND         ::= "and" | "AND"
NOT         ::= "not" | "NOT"

ALPHANUMERIC ::= ["a"-"z","A"-"Z","_"]
NUMERIC      ::= ["0"-"9"]

STUFF        ::= "&" | "'" | "~" | "" | "#" | "`" | "_" | "" | "^" |
       "" | "" | "$" | "" | "" | "%" | "" | "?" | "!" | "" | ""
TERMRESTRICTION ::= "+" | "-"
SLASH           ::= "/" >
ASTERISK        ::= "*" >
ATTR            ::= "@" >
PIPE            ::= "|" >
LPAR            ::= "(" >
RPAR            ::= ")" >
LBRACK          ::= "[" >
RBRACK          ::= "]" >
LBRACE          ::= "{" >
RBRACE          ::= "}" >
COMMA           ::= "," >
COLON           ::= ":" >
DOT             ::= "." >
ARITHMETIC      ::= "<" | ">" | "=" | "<=" | ">="
//////////
// NON terminals
//////////
Start        ::= Query
Query        ::= (Cas | "(" Query ")") Query2
Query2       ::= ((AND | OR) Query Query2 | "")
Cas          ::= AbsolutePath

AbsolutePath     ::= ( "/" ["/"] (Node | Attribute) [PathConstraints] [Filter] )+
RelativePath     ::= "." [AbsolutePath]
Node             ::= ["*"] Word ["*"] | "*" | "(" Node ("|" Node)+ ")"
Attribute        ::= "@" Word
PathConstraints  ::= "{" PathConstraint ("," PathConstraint)* "}"
PathConstraint   ::= Word ":" Word

Word             ::= (NUMERIC | ALPHANUMERIC)+

Filter           ::= "[" FilteredClause "]"
FilteredClause   ::= SimpleFilter | "(" FilteredClause ")
FilteredClause2  ::= ((AND | OR) FilteredClause FilteredClause2 | "")
SimpleFilter     ::= PredicateClause | ArithmeticClause
PredicateClause  ::= Predicate "(" RelativePath "," Keywords ")"
ArithmeticClause ::= RelativePath ("<" | ">" | "=" | "<=" | ">=") Word

Keywords           ::= (Keyword | Keyphrase | RestrictedKey)+
RestrictedKey      ::= ("+"|"-") (Keyword | Keyphrase)
Keyword            ::= Word [KeywordConstraints]
Keyphrase          ::= "\"" (Keyword)+ "\"" [KeywordConstraints]
KeywordConstraints ::= "{" KeywordConstraint ("," KeywordConstraint)* "}"
KeywordConstraint  ::= Word ":" Word
```

Understanding Differences between Search Requests in XML Element Retrieval

Jaap Kamps[1,2] Birger Larsen[3]

[1] Archives and Information Studies, University of Amsterdam, Amsterdam, The Netherlands
[2] Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands
[3] Information Studies, Royal School of Library and Information Science, Copenhagen, Denmark

ABSTRACT

XML retrieval, a very active branch of IR, studies the focused retrieval of semi-structured data. Although much progress has been made, especially through the annual INitiative for the Evaluation of XML retrieval (INEX), very little is known about XML element retrieval in action: What do users expect from an element retrieval system? What kind of information needs do they have? What sort of results do they request? Etc. In an effort to recover some of the answers, an extensive questionnaire was part of the peer topic creation process at INEX 2006. In this paper we present an analysis of the responses of topic authors. Our main general finding is that there is a great variety in the responses, and hence in the expectations about XML element retrieval.

Categories and Subject Descriptors

H.2 [**Database Management**]: H.2.3 Languages—*Query Languages*; H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries

General Terms

Measurement, Performance, Experimentation

Keywords

XML Retrieval, Search Requests, User Expectations

1. INTRODUCTION

Research in XML element retrieval attempts to take advantage of the structure of explicitly marked up documents to provide more focused retrieval results [7]. A special problem for this research area is that we have little knowledge about the expectations that potential users might have: As research in XML element retrieval is in its initial stages there are no operational systems with established user groups from which such expectations can be learned [15]. In this paper, we study a particular group of users who have worked intensively with an XML element retrieval system, in order to get some idea of their expectations of such systems.

SIGIR 2006 Workshop on XML Element Retrieval Methodology
August 10, 2006, Seattle, Washington, USA

The task of XML element retrieval is a much more complicated one than standard document retrieval. Not only must XML element retrieval systems be able to identify relevant content; in addition a suitable granularity of the returned elements must be decided on along with how to handle overlap among elements [9]. As a consequence the creation of test collections for XML element retrieval is a notable challenge in itself. The main research effort in this area has since 2002 been the INitiative for the Evaluation of XML Retrieval [INEX 7]. Mainly due to INEX, much progress has been made with dedicated retrieval techniques [e.g., 3, 6, 8].

In addition, INEX includes an interactive track from 2004 onwards that has as purpose to investigate the behavior of users as they interact with XML element retrieval systems [11, 13]. However, the users studied in the INEX interactive track have no prior experience in searching XML element retrieval systems and only interact with them in a single session. Therefore the track is to a certain extent limited to studying novice users.

A hitherto unstudied user group is the authors of topics for the test collection. The test collection topics are created collectively by members of the research groups participating in INEX. The topics are created through a number of steps which involve repeated exploratory searches in an XML element retrieval system, and the assessment of a large number of elements [10]. Thus on the one hand the task is a very specific one, but on the other hand it demands that the system is used extensively over several days. The topic authors thereby become one of the most experienced groups of XML element retrieval users. Because of the collaborative effort most users in this group are drawn from people close to the participating research groups and are as a result more closely resembling real users and real tasks than in most other IR research settings [e.g., TREC 14].

Therefore we added an on-line topic questionnaire in INEX 2006, which the topic authors completed immediately after submitting the final version of their topics. The questionnaire consisted of 19 questions about the topic familiarity, the type of information requested and expected, results presentation, and the use of structured queries. It is important to stress that the questionnaire data is collected in the initial phase of the INEX campaign, before the retrieval tasks, metrics, or assessment instructions have been finalized. From the responses we hope to learn, in an indirect manner, more about user expectations for XML element retrieval systems. Moreover, we plan to distribute these data together with the test collection and hope that they will prove to be a valuable addition.

Table 1: Number of candidate topics per topic author at INEX 2006.

Min	Max	Median	Mean	Std. deviation
1	6	2	2.41	1.69

Table 2: (B1) How familiar are you with the subject matter of the topic?

Answer	Frequency	Percentage
Not familiar	8	4%
	139	71%
Very familiar	48	25%

Table 3: (B2) Would you search for this topic in real-life?

Answer	Frequency	Percentage
yes	186	95%
no	9	5%

Table 4: (B3) Does your query differ from what you would type in a web search engine?

Answer	Frequency	Percentage
yes	33	17%
no	162	83%

The paper is structured as follows: Section 2 presents the questionnaire and presents an analysis of the main results. Section 3 discusses relations between the questions, and Section 4 gives conclusions and points to future work.

2. CANDIDATE TOPIC QUESTIONNAIRE

An IR test collection consists of a collection of documents, a set of search topics, and relevance judgments. For INEX 2006, the document collection is an XML'ified version of the English Wikipedia [5]. At INEX, search requests or topics are authored (and also judged) by the INEX participants [10].

At INEX 2006, 81 different topic authors submitted a total of 195 topics (see Table 1 for some statistics). A total of 125 of the candidate topics have been selected as the topics for the INEX 2006 ad hoc retrieval tasks.

Directly after submitting a candidate topic (see [10] for details), the topic author was presented with a new page containing a questionnaire consisting of 19 questions and an open space for comments on the questionnaire. The 19 questions dealt with various issues related to the background of the search request and the topic author.

- the topic author's familiarity with the topic;
- the type of information requested;
- the type of search results expected;
- the type of results presentation preferred; and
- the meaning of structured queries.

Below we summarize the responses to all 19 questions of the candidate topic questionnaire at INEX 2006.

2.1 Topic Familiarity

The topic questionnaire featured three questions dealing with the familiarity and naturalness of the topics:

B1 How familiar are you with the subject matter of the topic? (yes/no)

B2 Would you search for this topic in real-life? (yes/no)

B3 Does your query differ from what you would type in a web search engine? (yes/no)

Table 2 shows the familiarity with the subject matter of the topic at hand.[1] It is reassuring that the vast majority of topic authors is familiar with the subject, although there are still 4% of the topics where topic authors venture into unfamiliar terrain.

Table 3 shows whether the topic corresponds to a the real-life search. The responses are overwhelmingly yes. For topic authors answering no, there was a follow-up question asking for their motivation. Typical responses where knowing the answer already, or not being interested in the answer.

At INEX 2006, the topic statement consists of a short keyword title, and an optional structured query [10]. Table 4 shows whether the provided topic statement differs from what the topic author would issue as a query to a web search engine. For 83% of the topics, there is no difference. For topic authors answering yes, there was a follow-up question asking for their motivation. For many of the topics that are different, the topic authors consider the structured query as the search request (and mention that this is not supported on standard web search engines).

Based on the three questions, we can conclude that the majority of topic authors search for familiar subject matter, provide a real-life search task, and provide a standard web search engine query.

2.2 Type of Information Requested

The questionnaire contains seven questions dealing with the type of information requested:

B4 Are you looking for very specific information? (yes/no)

B5 Are you interested in reading a lot of relevant information on the topic? (yes/no)

B6 Could the topic be satisfied by combining the information in different (parts of) documents?

B7 Is the topic based on a seen relevant (part of a) document? (yes/no)

B8 Can information of equal relevance to the topic be found in several documents? (yes/no/don't know)

B9 Approximately how many articles in the whole collection do you expect to contain relevant information?

B10 Approximately how many relevant document parts do you expect in the whole collection?

[1]Due to a problem with the form for question B1, categories 2 and 4 of the original five point scale have been collapsed in the answers logs. We derive a three point scale for B1 by grouping the answers to categories 2, 3, and 4 as a single intermediate category.

Table 5: (B4) Are you looking for very specific information?

Answer	Frequency	Percentage
yes	114	58%
no	81	42%

Table 6: (B5) Are you interested in reading a lot of relevant information on the topic?

Answer	Frequency	Percentage
yes	123	63%
no	72	37%

Table 7: (B6) Could the topic be satisfied by combining the information in different (parts of) documents?

Answer	Frequency	Percentage
yes	160	82%
no	35	18%

Table 8: (B7) Is the topic based on a seen relevant (part of a) document?

Answer	Frequency	Percentage
yes	74	38%
no	121	62%

Table 9: (B8) Can information of equal relevance to the topic be found in several documents?

Answer	Frequency	Percentage
yes	163	84%
no	12	6%
don't know	20	10%

Table 5 shows whether topics are asking for very specific information. For 58% of the topics, the response is yes, indicating many topics can likely be answered by a relatively small amount of text.

Table 6 shows whether topics authors are interested in reading a lot of relevant information. Now, for 63% of the topics the answer is yes, indicating that recall is appreciated for most of the topics.

Table 7 shows whether the topics can be satisfied by combining information in different (parts of) documents. Here, for no less than 82% of the topics, the anwers is yes. This can be interpreted to indicate that many topics are multi-faceted.

These three questions, B4-6, try to assess the scope of the topics. The outcome is mixed: B4 indicates a narrow scope, but B6 indicates a broad scope. We return to the relation between the responses to these questions in Section 3 below.

Table 8 shows whether topics are based on a seen relevant document. Here, for 62% of the topics, the response is no, indicating that these are clearly not "known-item" topics.

Table 9 shows whether information of equal relevance can be found in different documents. For 84% of the topics, the reponse is yes, indicating that these are informational search topics rather than navigational topics [1].

Tables 10 and 11 show some statistics on the expected number of articles and elements with relevance. The distributions are both fairly skewed, but showing that relevance is expected in a wide range of articles and elements.

Table 10: (B9) Approximately how many articles in the whole collection do you expect to contain relevant information?

Min	Max	Median	Mean	Std. deviation
2	15,000	20	128	1097

Table 11: (B10) Approximately how many relevant document parts do you expect in the whole collection?

Min	Max	Median	Mean	Std. deviation
2	20,000	50	289	1671

Table 12: (B11) Could a relevant result be (check all that apply)?

Answer	Frequency	Percentage
a single sentence	81	42%
a single paragraph	139	71%
a single (sub)section	170	87%
a whole article	160	82%

Table 13: (B12) Can the topic be completely satisfied by a single relevant result?

Answer	Frequency	Percentage
yes	74	38%
no	121	62%

These four questions, B7-10, try to assess to what extent search requests resemble known-item search topics or ad hoc retrieval topics. Based on the responses, we can conclude that the topics are predominantly general informational topics.

2.3 Type of Results Expected

The questionnaire has four questions zooming in on the type of search results expected:

B11 Could a relevant result be (check all that apply): a single sentence; a single paragraph; a single (sub)section; a whole article.

B12 Can the topic be completely satisfied by a single relevant result? (yes/no)

B13 Is there additional value in reading several relevant results? (yes/no)

B14 Is there additional value in knowing all relevant results? (yes/no)

Table 12 shows the expected result granularity (note that multiples answers are possible). Some observations present themselves. First, for no less than 42% of the topics a single sentence could be a relevant result, indicating a very specific information need that can be answered by a single sentence. Second, for no less than 82% of the topics a whole article could be a relevant result.

Table 13 shows which topics can be completely satisfied by a single relevant result. For 38% of the topics this is the case.

Table 14: (B13) Is there additional value in reading several relevant results?

Answer		Frequency	Percentage
Not important	1	7	4%
	2	15	8%
	3	36	18%
	4	71	36%
Very important	5	66	34%

Table 15: (B14) Is there additional value in knowing all relevant results?

Answer		Frequency	Percentage
Not important	1	21	11%
	2	41	21%
	3	49	25%
	4	53	27%
Very important	5	31	16%

Table 16: (B15) Would you prefer seeing?

Answer	Frequency	Percentage
only the best results	82	42%
all relevant results	106	54%
don't know	7	4%

Table 17: (B16) Would you prefer seeing?

Answer	Frequency	Percentage
isolated document parts	69	35%
the article's context	105	54%
don't know	21	11%

Table 14 shows the importance of reading several relevant results. For 70% of the topics there is clear importance (4 or 5 on the 5-point scale).

Table 15 shows the importance of reading all relevant results. Now we see a very even distribution of topics over importance.

These three questions, B12-14, try to assess the relative importance of precision and recall for the search requests. We see that for most topics, the topic authors are interested in recall.

2.4 Results Presentation

The questionnaire has two questions zooming in on result presentation:

B15 Would you prefer seeing: only the best results; all relevant results; don't know

B16 Would you prefer seeing: isolated document parts; the article's context; don't know

Table 16 shows how many of the relevant results topic authors prefer to see. The outcome is mixed: for 54% of the topics, all results should be shown, and for 42% of the topics only the best results need to be shown.

Table 17 shows whether results should be shown in their original article's context. For 54% of the topics, a presentation in context is preferred, whereas for 35% of the topics isolated results are preferred.

Table 18: (B17) Do you assume perfect knowledge of the DTD?

Answer	Frequency	Percentage
yes	24	12%
no	171	88%

Table 19: (B18) Do you assume that the structure of at least one relevant result is known?

Answer	Frequency	Percentage
yes	65	33%
no	130	67%

Table 20: (B19) Do you assume that references to the document structure are vague and imprecise?

Answer	Frequency	Percentage
yes	121	62%
no	74	38%

These two questions, B15-16, show that topic authors have different preferences on the presentation of XML element retrieval results.

2.5 Structured Queries

The questionnaire featured three questions dealing with structured queries, the so-called content-and-structure (CAS) queries formulated in the NEXI language [16].

B17 Do you assume perfect knowledge of the DTD? (yes/no)

B18 Do you assume that the structure of at least one relevant result is known? (yes/no)

B19 Do you assume that references to the document structure are vague and imprecise? (yes/no)

Even though these questions where also optional (because formulating a structured query was no requirement), the questions were answered for all topics.

Table 18 shows whether topic authors assumed a perfect knowledge of the collection's mark-up structure. As it turned out, for 12% of the topics, perfect knowledge of the DTD is assumed.

Table 19 shows whether the mark-up structure of at least one relevant result is known. Now, for 33% of the topics it is assumed that the structure at least one result is known.

Table 20 shows how to interpret structural references in the search request. Here, for 62% of the topics, structural references are considered vague and imprecise. However, in 38% of the topics, the structural hints are meant to be interpreted literally.

These three questions, B17-19, address the meaning of structured queries in XML element retrieval. The results show that for a majority of topics the structural references are merely search hints, but that for a seizable fraction structure should be taken seriously.

3. RELATIONS

In this section, we analyze the relation between responses to different questions in the questionnaire. Table 21 show the relations between pairs of questions in the questionnaire. We will discuss these relations in detail.

First we focus on topic familiarity.

Table 21: Relationship between answers for pairs of questions (chi-square test at percentiles 0.95 and 0.99).

	B1	B2	B3	B4	B5	B6	B7	B8	B12	B13	B14	B15	B16	B17	B18	B19
B1																
B2	-															
B3	-	-														
B4	0.99	-	-													
B5	-	-	0.95	0.99												
B6	-	0.99	-	0.95	-											
B7	-	-	-	0.95	-	-										
B8	-	-	-	-	-	0.95	-									
B12	-	-	-	0.99	0.99	-	-	-								
B13	-	-	-	-	0.99	0.99	-	0.95	0.99							
B14	0.95	-	-	-	0.99	-	-	-	0.99	0.99						
B15	-	-	-	-	0.99	0.95	-	-	0.99	0.99	0.99					
B16	-	-	-	-	0.95	-	0.99	-	-	-	-	-				
B17	0.95	-	-	-	-	0.95	0.99	-	-	-	-	-	-			
B18	-	-	0.99	0.95	-	-	0.95	-	-	-	-	-	-	0.99		
B19	-	-	-	-	-	-	-	-	-	-	-	0.95	-	-	-	

B1,B4 Topics which the author is very familiar with are more often very specific.

B1,B14 Topics which the author is moderate familiar with, have a moderate importance of knowing all the relevant results.

B1,B17 Topics which the author is very familiar with the subject matter of the topic at hand, do more often assume perfect knowledge of the DTD.

Here, the relation between B1 and B4 is interesting: topic authors ask more specific queries about familiar subject matter. This simple observation has a bearing on the sort of users and tasks for which XML element retrieval system is most suitable.

Second, we focus on the naturalness of the topic and query.

B2,B6 Real-life topic are more often satisfied by combining information in different (parts of) documents.

B3,B5 Topic statements identical to Web search engine queries, make reading a lot of information more interesting.

B3,B18 Topic statements identical to Web search engine queries, less often assume that the structure of at least one relevant result is known.

These relations suggest that these topics are resonating closely with the sort of search request issued in real world information gathering.

Third, we look at the specificness of the topics:

B4,B5 Topics asking for very specific information, make reading a lot of information less interesting.

B4,B6 Topics asking for very specific information, do not expect answers from combining information in different (parts of) documents.

B4,B7,B18 Topics asking for for very specific information, are often based on a seen relevant document; and more often assume that the structure of at least one relevant document is known.

B4,B12 Topics asking for very specific information, are more often completely satisfied with a single relevant result.

Here the relation between B4 and B6 is clearly inverse, indicating that very specific topics are mono-faceted. The general suggestion is that specific topics form a category with distinct characteristics.

Fourth, we discuss the importance of reading relevant information:

B5,B12,B13,B14,B15 Topics for which it is of interest to read a lot of relevant information; are less often completely satisfied with a single relevant result; make reading several relevant results more important; make knowing all results more important; and make seeing all results more important.

B5,B16 Topics for which it is of interest to read a lot of relevant information, it is preferred to read information in the article's context.

Here the general suggestion is that topics for which reading a lot of relevant information is important also from a distinct category, which, considering the inverse relation between B4 and B5, is roughly complementing the category of specific topics.

Fifth, we focus on topics that can be satisfied by the combination of information in different (parts of) documents:

B6,B8,B13,B15 Topics that can be satisfied by combining information in different (parts of) documents, more often have information of equal relevance in different documents; and make reading several relevant results more important; and make seeing all relevant results more important.

B6,B17 For all topics assuming perfect knowledge of the DTD, it is assumed that they can be satisfied by combining information in different (parts of) documents.

Given that B4 on topic specificity was inversely related with B6, these relations reaffirm the differences between specific topics, and topic for which reading a lot of relevant information is interesting.

Sixth, we continue with seen relevant documents:

B7,B16 For topics based on a seen relevant document, it is less often preferred to see the article's context.

B7,B17,B18 Topics based on a seen relevant document are more often assuming perfect knowledge of the DTD; and are more often assuming that the structure of at least one relevant document is known.

These relations clearly suggest the prior knowledge assumed on the part of the searcher for these topics.

Finally, seventh, we look at vague structural hints:

B15,B19 For topics with vague structural hints, it is more often preferred to see only the best results.

This is an interesting relation, which could be interpreted to mean to vague structural hints are provided to improve the ranking of certain elements.

The responses to the two numerical questions on the number of relevant articles and document parts, B9 and B10, are clearly related (Pearson correlation 0.9215), as may be expected.

We excluded above the responses to question B11 (about the granularity of potential relevant results) since multiple answers are possible. As it turns out, there are three relations between the responses to B11: sentence is related to paragraph, paragraph is related section, and section is related to article. For all other pairs of responses (e.g, sentence-article), we find no relation.

Finally, recall that most topic authors submitted multiple candidate topics. We analyzed the relation between the topic author and the questions above. For the twelve questions, B1, B3, B5, B7, B8, B11, and B14–B19, the responses are related to the particular topic author at hand. This suggests that some of the responses are mainly related to the topic at hand, whereas others are mainly related to the particular user.

4. CONCLUSIONS

Studying the expectations of the INEX topic authors as an example of an XML element retrieval user group has its advantages and disadvantages. The task they have performed is a highly specific and somewhat artificial one (that of producing a test collection topic) compared to the natural tasks of real users. However, the INEX topic authors are probably much closer to real users than in other IR test collection building efforts because they are mainly recruited among the participating research groups. It is therefore reassuring that most topic authors searched for familiar subject matter and real-life tasks using queries similar to web queries. In addition to this the Wikipedia collection covers a very broad range of subject matter, and the topic authors have generally extensive experience with XML element retrieval systems. Arguably, the results of this study will extend our understanding of what users expect from an XML element retrieval system.

Perhaps the most striking observation is that there is such great variety in the expectations of the topic authors. This may, in turn, indicate that there is a range of several different XML retrieval tasks types. This give broad support to the decision at INEX to define a number of distinct retrieval tasks [4]. In particular, we have found that there are a number of relations worth considering. Among these is that great topic familiarity lead to more specific topics, and that the specific topics tend to be mono-faceted and can be completely satisfied by a single relevant result. For more complex topics where moderate or even high recall is desired it is preferred to read more information and to present the results in the articles context. In addition, it appears that there are two distinct views on the meaning of structural hints: the majority regards them as only vague hints, but for a sizable fraction they should be taken seriously.

In this paper, we only reported the responses to the questionnaire as a survey amongst candidate topic authors, which can be construed as a particular group of XML element retrieval users. However, recall that 125 of the candidate topics were selected as the INEX 2006 ad hoc retrieval topics, and—at a later stage of the INEX campaign—the topic authors will be asked to make relevance judgments for pooled sets of elements. That is, the questionnaire data also becomes part of the evaluation test-suite that will be constructed during INEX 2006, providing valuable contextual data on the topics of request and their topic authors.

This enriched test set will have a number of unique features. First, it will allow to breakdown the set of topics in various meaningful categories, and zoom in on the relative performance for such a group of topics. Second, zooming on particular topic categories will help to explain diverging results between different techniques, tasks, and metrics. Third, it will reveal the importance of each of the variables measured in the questionnaire for the various INEX tasks [4]. Fourth, it may help us understand what are the fundamental differences between tasks, which will lead in turn to better retrieval techniques for individual tasks. In short, the rich contextual information from the topics questionnaire will significantly boost the value of the test suite constructed during INEX 2006, and greatly increase the potential reuse of the test suite in the future.

The Cranfield tradition of test collection development tries to abstract away from individual differences between assessors [17]. Yet at the same time, it is known for long that individual difference are one of the greatest sources of variation in relevance judgments and system failure [2, 12]. Given that the task of XML element retrieval is of a higher complexity that standard document retrieval, due to the document structure, the fine-grained judgments, and, perhaps, a lack of consensus on the precise retrieval task, it is more than plausible that individual differences have a much greater impact. The questionnaire data will shed light on the impact of these differences—even zoom in on the relative impact of specific features—and at the same time provide a handle to deal with them.

Acknowledgments

Thank you to all INEX 2006 topic authors, and to Mounia Lalmas and Saadia Maalik for their help with the questionnaire and data collection.

Jaap Kamps was supported by the Netherlands Organization for Scientific Research (NWO) under project numbers 612.066.513, 612.066.302, and 640.001.501.

Birger Larsen was supported by the NORSLIS Research School.

REFERENCES

[1] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[2] C. Buckley. Why current ir engines fail. In M. Sanderson, K. Järvelin, J. Allan, and P. Bruza, editors, *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 584–585. ACM Press, New York NY, USA, 2004.

[3] D. Carmel, Y. S. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer. Searching XML documents via XML fragments. In C. Clarke, G. Cormack, J. Callan, D. Hawking, and A. Smeaton, editors, *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 151–158. ACM Press, New York NY, USA, 2003.

[4] C. Clarke, J. Kamps, and M. Lalmas. INEX 2006 retrieval task and result specification. In *INEX 2006*, 2006.

[5] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 40:64–69, 2006.

[6] N. Fuhr and K. Großjohann. XIRQL: A query language for information retrieval in XML documents. In D. H. Kraft, W. B. Croft, D. J. Harper, and J. Zobel, editors, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 172–180. ACM Press, New York NY, USA, 2001.

[7] INEX. INitiative for the Evaluation of XML Retrieval, 2006. `http://inex.is.informatik.uni-duisburg.de/2006/`.

[8] J. Kamps, M. de Rijke, and B. Sigurbjörnsson. Length normalization in XML retrieval. In M. Sanderson, K. Järvelin, J. Allan, and P. Bruza, editors, *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 80–87. ACM Press, New York NY, USA, 2004.

[9] G. Kazai, M. Lalmas, and A. P. de Vries. The overlap problem in content-oriented XML retrieval evaluation. In *Proceedings of the 27th Annual International ACM SIGIR Conference*, pages 72–79. ACM Press, New York NY, USA, 2004.

[10] B. Larsen and A. Trotman. INEX 2006 guidelines for topic development. In *INEX 2006*, 2006.

[11] B. Larsen, S. Malik, and T. Tombros. The interactive track at INEX 2005. In N. Fuhr, M. Lalmas, S. Malik, and G. Kazai, editors, *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005)*, volume 3977. Springer Verlag, Heidelberg, 2006.

[12] T. Saracevic. Relevance: A review of and a framework for the thinking on the notion in information science. *JASIS*, 26:321–343, 1975.

[13] A. Tombros, B. Larsen, and S. Malik. The interactive track at INEX 2004. In *Advances in XML Information Retrieval. Third Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2004*, volume 3493 of *Lecture Notes in Computer Science*, pages 410–423. Springer Verlag, Heidelberg, 2005.

[14] TREC. Text REtrieval Conference, 2006. `http://trec.nist.gov/`.

[15] A. Trotman. Wanted: Element retrieval users. In A. Trotman, M. Lalmas, and N. Fuhr, editors, *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, pages 63–69. University of Otago, Dunedin New Zealand, 2005.

[16] A. Trotman and B. Sigurbjörnsson. Narrowed Extended XPath I (NEXI). In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors, *Proceedings of the Initiative for the Evaluation of XML Retrieval (INEX 2004)*, Lecture Notes in Computer Science, pages 16–40. Springer Verlag, Heidelberg, 2005.

[17] E. M. Voorhees. The philosophy of information retrieval evaluation. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Evaluation of Cross-Language Information Retrieval Systems, CLEF 2001*, volume 2406 of *Lecture Notes in Computer Science*, pages 355–370. Springer, 2002.

What does Shakespeare have to do with INEX?

User Queries, Assessment Behaviour and Best Entry Point Selection Strategies in XML Retrieval

Gabriella Kazai
Microsoft Research Cambridge
gabkaz@microsoft.com

Elham Ashoori
Queen Mary, University of London
elham@dcs.qmul.ac.uk

ABSTRACT

Since 2002, the INitiative for the Evaluation of XML Retrieval (INEX) has been building an XML test collection for the evaluation of content-oriented XML search systems. In 2006, INEX extended its range of investigated user tasks to include the Best in Context task, where systems are required to return Best Entry Points (BEPs) to the user. In this paper we take a look back at a small user study conducted at Queen Mary, University of London, which resulted in the construction of the Shakespeare XML test collection. This test collection includes - in addition to the standard components of documents, user queries and relevance assessments - BEP judgments, where BEPs were defined as optimal points for browsing a document's structure to access relevant information. We examine some of the findings of topic author and assessor behaviours in the Shakespeare study and draw comparisons to findings reported at INEX. In addition, we provide a detailed analysis of users' BEP selection strategies and review related user studies with the aim to help guide efforts at INEX.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

XML test collection, Best Entry Points

1. INTRODUCTION

The Shakespeare user study [9] involved 11 English and Drama students at Queen Mary, University of London and resulted in the construction of a small XML test collection[1] (10MB). Each XML document in the collection is a Shakespeare play consisting of the original text of the plays and the XML markup. The markup follows the logical structure of the plays with the following main structural components: PLAY (root nodes), ACT, SCENE, SPEECH (composite nodes), and LINE or STAGEDIR (leaf nodes).

Participants of the study were asked to come up with user queries for 3 plays of their choice, to provide relevance assessments using binary relevance scale and a yellow-marker design (the same assessment procedure which is to be adopted at INEX this year), and finally to provide BEP assessments.

We report on some of the observations of the study regarding the types of user information needs in the context of XML retrieval (Section 2) and user behaviour during relevance assessments (Section 3). In Section 4, we examine the relationship of relevance assessments at INEX 2005 to semantic units. In Section 5 we provide detailed analysis of users' BEP selection strategies.

2. QUERIES

Based on participants' familiarity, 12 plays were selected (out of 37) for the study. Participants were asked to formulate queries addressing real information needs, and covering topics related to their chosen plays that were of interest to them. It was desirable to obtain queries of varying complexity, and two main types were identified in this context:

- Factual questions, where it is likely that a small number of short passages will provide the answer, e.g. "How old is Juliet?"
- Essay topics, where it is likely that reference will have to be made to many complex passages, e.g. "The character of Lady Macbeth".

2.1 CAS vs. CO queries

Participants were not told about possibilities to query using structural constraints, for example, to limit the context of possible answers. Any decision to impose such structural constraints within a query was left up to the participants. The aim was to obtain an unbiased query set where it could be observed whether there is in fact a real world need for the different query types, i.e. content-only (CO) and content-and-structure (CAS).

A total of 215 queries were submitted (average 18 per play and 19.5 per participant). Table 1 shows their distribution across the different query categories. 43% of the queries were CAS and 57% were CO queries. This shows that both CO

[1]Available at http://qmir.dcs.qmul.ac.uk/Focus/resources.htm

SIGIR 2006 Workshop on XML Element Retrieval Methodology
August 10, 2006, Seattle, Washington, USA.

	CO	CAS	Total
Factual question	54	15	69
Essay topic	68	78	146
Total	122	93	215

Table 1: Distribution of the original 215 submitted queries across query types

and CAS queries are naturally needed and used by novice users when searching structured documents.

In contrast to the above methodology, INEX explicitly instructs topic authors to create both CO and CAS queries, while the complexity of the query is left unspecified. A number of studies, however, showed that INEX topics can also be classified as specific (narrow) or general (broad) topics [5, 14]. It could be argued that users' requests for more specific information is related to the assumed advantage of XML IR over traditional IR: the ability to locate exact relevant fragments within documents. Given that such relevant fragments may intuitively be thought of as smaller, more focused components, this could inadvertently influence users in requesting more specific information and hence ask more factual questions. This has been raised by [20, 12] commenting that INEX participants struggle to come up with queries that can take advantage of the structure of the collection (and make sense at the same time).

2.2 Influence of a semantic unit

A closer look at the CAS queries of the Shakespeare study reveals that the most commonly used structural constraint is simply to limit the context of the query to the level of PLAY (e.g. "How is Sebastian feminised in the play?") and even to a specific play (e.g. "Trickery and treachery in Much Ado about Nothing."). 80% of the CAS queries were of this type. Only 18 of the 93 CAS queries contained explicit structural references to ACT, SCENE or even LINE elements.

The fact that the majority of the CAS queries only specifies the unit of the whole play as structural constraint suggests that these semantically coherent and independent units of information represent the default context for users. Some users may then go further and explore the inner structure of the individual documents, but the unit of the documents themselves are typically identified first. These findings go hand in hand with the investigations of the FERMI project on multimedia IR [2], and provide support for the fetch and browse strategy proposed there: retrieving whole documents (fetch) then focusing the users attention to the most specific components within the documents (browse).

It is necessary to note, however, that the findings of the Shakespeare study were heavily biased due to the experimental setup, whereby participants were asked to come up with queries for their selected plays. So naturally, the obtained queries tended to be limited to the scope of a given play. The same can be said for CO queries, most of which were also meant with specific plays in mind, even though the context of the query was not explicitly stated. For example the query "To what extent is Hamlets madness a pretence?" implicitly assumes that the context is the play Hamlet.

Studies of the INEX topics have shown similar results, highlighting users' natural association of complete ARTICLE elements as overall semantic units (27% of INEX 2003/2004 CAS topics targets ARTICLEs) [12, 20]. Although it has also been argued that this is due to forced pressure on topic authors to introduce structure into their information needs. An analysis of the relevance assessments for CO topics, on the other hand, revealed that users do generally prefer to be returned smaller, more specific elements [15, 16] regardless of the existance of an article level semantic unit. The combination of these findings again supports the fetch and browse strategy, which is explored since 2005 at INEX, as an intuitive user task.

3. RELEVANCE ASSESSMENTS

From the original pool of 215 queries submitted by the participants of the Shakespeare study, 43 were selected into the final set for which relevance assessments were collected. A binary relevance scale was employed and assessments were collected from multiple judges. Following the yellow-marker design, participants were provided with printed versions of the plays and queries and were asked to highlight relevant passages on the printed documents by hand. Relevant passages were described as those that they would consult (read or reference) in order to answer a given query.

The highlighted passages were then converted into assessments on structured documents, where the derived set of relevance assessments consists of all the leaf nodes that contain highlighted parts. The obtained 117 sets of relevance assessments (from the 11 participants for the 43 queries) lead to a total of 6,296 relevant leaf level XML elements. The multiple sets of relevance assessments were then merged for each query to form the final set of assessments for the test collection. After merging, a total of 4,898 unique leaf level XML elements were obtained in 43 query sets (average 114 leaf XML elements per query).

3.0.1 Assessor agreement

Assessor agreement was measured as the size of the intersection of the different relevant sets, obtained by the different participants for the same query, divided by the size of the union of the relevant sets [21].

Since assessments were collected at the leaf node level, in order to investigate assessor agreement at higher structural levels (e.g. SPEECH or SCENE), an optimistic relevance propagation strategy was employed [19]. According to this, a container element is judged relevant if at least one of its contained elements is relevant.

The resulting assessor agreement data can be found in Table 2. It shows that agreement increases consistently with higher structural levels. This leads to the overall conclusion that while participants are likely to disagree about the exact location of the relevant information, they tend to agree on the general area in which the answers to a query can be found. The results also show that query type and complexity do not have a strong effect on assessor agreement, although factual queries show slightly higher agreement at most structural levels.

The above agreement levels are (expectedly) much superior

	Leaf	SPEECH	SCENE	ACT	PLAY
Factual question	35	43	59	84	100
Essay topic	27	30	68	76	100
CO	29	35	65	80	100
CAS	30	30	63	73	100
Average	31	35	64	78	100

Table 2: Average assessor agreement (as %) for the different query types at various structural levels

to those reported for INEX (e.g. 0.27 for INEX 2003 and 0.39 for INEX 2004 data [13, 16]) due to the implicit selection of a PLAY as the context of a query.

3.0.2 *Effect of result presentation assumptions on assessor agreement*

A closer look at the collected assessments reveals a possible reason for low assessor agreement at the lower structural levels. Looking at the patterns of highlighted texts, two clearly identifiable trends emerge: some assessors tended to highlight only the very minimal text fragments which provide the most direct answer to a query, while some assessors followed a different strategy and highlighted large contiguous text fragments. During the interviews it came to light that the latter approach was chosen in order to ensure that contextual information was not missed. This provides direct evidence that relevance assessments can be influenced by assumptions about how information may be returned by a retrieval system to the users. The assumption that relevant information is presented to the user as highlighted text within its context could lead to stricter assessments, where only the most specific fragments are marked relevant. On the other hand, assuming that the user is returned only the highlighted information without its context may encourage assessors to be more liberal regarding their criteria for relevance.

This finding may bear significance when evaluating the various tasks at INEX, given that assessors may be influenced by the actual assessment interface in their assessment task. In particular, the presentation of results grouped by articles and the task of highlighting relevant text fragments provides a close match with the Relevant in Context task. However, it is not clear how the evaluation of, e.g., the Focused task may be influenced by the way relevance assessments are collected.

4. RELEVANT VS. SEMANTIC UNITS AT INEX

Apart from the main semantic unit of a whole document, a document can be considered as a sequence of semantically coherent units or topics. Documents can be semantically decomposed through the application of a topic segmentation algorithm. To this end, we employ the topic segmentation of TextTiling[2] [6], which is based on lexical cohesion where change in vocabulary signifies a topic shift. TextTiling is a linear segmentation algorithm which considers the discourse unit to correspond to a paragraph and therefore subdivides the text into multi-paragraph segments. The algorithm determines the number of segments, referred to as tiles, assigned to each document, by considering segment boundaries to correspond to gaps with depth scores above a certain threshold.

Our aim is to understand how people provide relevance assessments, why and how do they highlight text fragments. We are interested in finding out whether people tend to highlight text fragments which form semantic units, i.e. when strong coupling exist within the fragment which is then loosely coupled to its neighbours, or if they highlight fragments that form only some part of a semantic unit. For example, in the Shakespeare study, we found that some assessors highlighted whole sections, while others highlighted only a couple of important lines from the section.

We investigate whether the text segments produced by TextTiling tend to match up with what is highlighted by the assessors as this would provide some level of evidence that people tend to choose such semantic units over smaller fragments. This in turn could provide evidence that people may be influenced in their assessment task by how they imagine results would be returned by a system. For example, if whole semantic units are more often highlighted, then assessors may assume that users would prefer to see the whole context of relevant information. If text fragments within semantic units are highlighted then assessors may assume that it is more important to point the user's attention to the specific relevant part.

The semantic decomposition of an XML document is used as a basis to calculate the matching between the highlighted passages and the semantic segments based on the relevance assessments v.7 for 29 CO+S and 34 CAS topics of the INEX 2005 data set [11]. We set the TextTiling algorithm's parameters to $W = 20$ and $K = 6$ (recommended values [6]).

We calculate the following measures:

1. Length ratio: length of highlighted text / length of text tiles that completely cover the highlighted text
2. Tile count: average number of text tiles that cover a highlighted text fragment

4.1 Results

For the purpose of our investigation, we consider paragraph elements[3] to be the lowest possible level of granularity of a retrieval unit. Due to the segmentation procedure, out of the 4280 (5942) highlighted passages for CO (CAS) topics, we were only able to use 3309 (4323) passages which start and end somewhere inside a paragraph.

Table 3 shows the calculated statistics for length ratio, passage size and tile count. Results are reported for both CO and CAS topics averaged over all highlighted passages (1st column) and over statistics calculated per query (2nd column).

Comparing the average passage size for both topic set, 768.91 for CO vs. 1463.1 for CAS topic, clearly shows that the highlighted passages for CAS topics are on average larger than

[2]http://elib.cs.berkeley.edu/src/texttiles/

[3]Paragraph elements are any elements of the "para" entity as defined in the INEX document collection DTD (<!ENTITY % para "ilrj|ip1|ip2|ip3|ip4|ip5|itemnone|p|p1|p2|p3">).

Table 3: Statistics of the matching between the highlighted relevant passages and the TextTiling semantic segments

	All Passages		Per Query	
	CO	CAS	CO	CAS
Average passage size	768.91	1463.1	1286.86	2127.27
Average tile count	1.90	3.05	2.73	4.56
Average length ratio	33.04	46.70	40.83	49.21
Standard deviation for length ratio	28.98	32.52	12.07	19.03

those for CO topics. This observation is somewhat surprising as one would expect that CAS topics would reflect more specific information needs, associated with shorter relevant snippets. However the finding does accord well with those of the Shakespeare study (Table 1), where most CAS queries were essay topics.

Looking at the length ratio, we find that text fragments highlighted by assessors as relevant tend to form only 33% of semantic units for CO topics, compared with 46.7% for CAS topics. This could mean that assessors tend to highlight more context for CAS queries. A counter argument may be that since users are not restricted by the requested structure for CO topics, they are more free to select smaller passages.

Comparing the standard deviation of length ratio for the two averages shows that although in general the length ratio for passages varies highly, smaller deviation exists among assessors when we group passages per query. This suggests that different users follow similar procedures for highlighting passages.

Overall, we found that (for our restricted subset of elements) users highlighted longer passages for CAS topics, where these passages closer matched the semantic segments within the documents.

5. BEP ASSESSMENTS

In the Shakespeare study BEPs have been defined as document components that represent optimal starting points for browsing and accessing relevant information in structured documents.

BEP assessments were solicited by interviewing participants individually. BEPs were identified by consulting the merged relevance assessments collected from all assessors of a query. The selection of BEPs required the use of an interface that allowed participants to browse the document structure and the relevant information within. The purpose of the interface was to show the context of the relevant fragments, and allow the user to form an intuitive understanding of the costs associated with finding relevant information through browsing from potential BEPs. Using the interface, participants were asked to identify BEPs as those document components that they would prefer to be retrieved by a search engine in response to a query.

As a result, a total of 928 BEPs were collected from the 11 participants for the 43 queries (in 117 sets). This number was reduced to 521 by removing duplicates. The average number of BEPs per query was hence 21.58 for non-unique elements and 12.12 for unique elements.

	Leaf	SPEECH	SCENE	PLAY	All
Factual question	63	52	67	-	67
Essay topic	46	62	41	0	57
CO	55	60	45	-	62
CAS	35	59	50	0	53
Average	49	58	51	0	60

Table 4: Average BEP agreement for the different query types at various structural levels (shown as %)

	Leaf	SPEECH	SCENE	ACT	PLAY
Factual q.	58	33	4	0	0
Essay topic	41	53	6	0	0
CO	46	49	5	0	0
CAS	29	57	7	0	0
Average	44	50	5	0	0

Table 5: Distribution of BEPs for different query categories across structural levels (given as %)

5.0.1 BEP assessment agreement

Table 4 shows the results for assessor agreement for BEPs. Compared with assessor agreement for relevance assessment, agreement is much higher for BEPs for all query categories at leaf and SPEECH element levels.

Agreement at higher structural levels is heavily influenced by the sparseness of the sample data (see Table 5), e.g. there are no ACT BEPs, only 5% of all BEPs are SCENE nodes, and only one participant chose the PLAY node as BEP for one query (appears as 0%).

Highest agreement across all levels is for the factual queries. This is likely to be due to the fact that factual queries have a smaller number of relevance fragments (which are also usually tighter clustered) than queries from other categories, so there was less potential for disagreement between participants when choosing BEPs.

Overall, it can be seen that a reasonable level of BEP agreement is achieved for all query categories across all structural levels (with the exception of PLAY), showing that the concept of BEP was found to be intuitive by the participants. These results show, especially given the comparatively low levels of assessor agreement on relevance, that BEPs may provide a more stable basis for retrieval evaluation. A disadvantage is that BEP data tends to be much sparser than relevance data (since one BEP usually represents a whole cluster of relevant nodes), which then has an inverse effect on evaluation stability [1].

5.0.2 BEP selection strategies

The distribution of BEPs in Table 5 shows the overwhelming dominance of leaf and SPEECH level BEPs, which together make up 94% of all BEPs. This suggests that participants generally preferred more specific, focused components as entry points.

A comparison of the different query categories shows that factual queries led to the most specific BEPs, with above average number of leaf level BEPs and below average number of BEPs at higher structural levels. This is likely to be related to the question answering nature of factual queries, where users tend to seek short focused answers. Such answer nodes are also then seen as best candidates for entry points. On the other hand, CAS queries show a trend contrary to this, where SPEECH level BEPs were found the most popular choice for BEPs. This finding is more likely a result of a combination of the structural aspects in CAS queries, i.e. some queries explicitly target SPEECH elements, and the influence of essay topic type queries that make up 10 of the 12 CAS queries, which are often associated with long extents of relevant texts.

To further investigate participants' BEP selection strategies, the relationship of the nominated BEPs to the cluster of relevant information for which they provide an entry point is examined next.

The following tree main types of BEP strategies were identified:

- Container BEP (PBEP): when the parent node of relevant elements is selected as BEP.
- "Start reading here" BEP (SBEP): when a leaf node in a sequence of relevant leaf nodes is selected as BEP. This is usually (but not always) the first node of a sequence that makes up a relevant text fragment. To distinguish between these two cases, BEPs which are the first nodes in a sequence are denoted as SBEP-1, and BEPs which are from somewhere inside the sequence are labeled as SBEP-M. In addition to these, in a number of cases, a BEP was chosen to represent a single relevant leaf node (e.g. when only a single LINE element was highlighted by an assessor), these are denoted as SBEP-SL.
- Combined BEP (CBEP): when a parent node in a sequence of relevant parent nodes was selected, e.g. the first SPEECH node in a sequence of SPEECH elements. Again, usually the first node of a sequence is selected as BEP, these are denoted as CBEP-1, but sometimes nodes in the middle or end of a sequence were picked, these are denoted as CBEP-M.

Table 6 shows the distribution of the different types of BEPs, based on the 521 unique BEPs. Note that micro-averaging was used in these calculations as the sample size for the different BEP types varied widely across queries (hence macro-averaging here would likely lead to skewed averages[4]). By using micro-averaging all BEPs belonging to queries of a given query category were first pooled and then their distribution with respect to the BEP type was examined. This way each BEP was counted with equal importance.

According to the findings, the most popular type of BEPs, with 44.9%, was the "Start reading here" BEP (SBEP), which means that participants in most of the cases simply selected a leaf level entry point, representing the point where they would prefer to be directed to and where they would want to start reading the text. From the SBEP type BEPs (taken as 100%), in the majority of the cases (62%) the BEP chosen was the first leaf node of the sequence of relevant leaf nodes (SBEP-1). Interestingly, however, 10% of SBEP type BEPs were leaf nodes that were selected from somewhere inside the sequence of relevant leaf nodes (SBEP-M). In a few cases this node was the very last node in the sequence. During the interviews, it was explained that such BEPs were usually selected when they contained highly relevant information or for factual questions when they provided the actual answer. One raised point was that once users are directed to these mid-sequence entry points, they can just browse around in the text to read the context if required, but it was more important that the first thing they would see is the relevant information. Finally, a large percentage of the cases concerned SBEP-SL types (28% of all SBEPs), where the single relevant leaf nodes were simply nominated as BEPs themselves. These were usually single LINE nodes that stood relatively separated from any other relevant fragments.

The next most popular BEP type, with 30.8%, was the Container BEP (PBEP). These were nodes at varying levels of the hierarchy: the vast majority being SPEECH nodes (80.12%) then SCENE nodes (19.28%). The remaining 0.06% is a result of the single PLAY BEP chosen by one participant.

Finally, 24.30% of all BEPs were of the combined BEP (CBEP) type, where a SPEECH node is chosen from a sequence of relevant SPEECH nodes. 94% of these were BEPs where the first node is chosen from the sequence. In 6% of the cases, just as for SBEPs a node from the middle of the sequence was chosen, again, for reasons to do with containing highly relevant information.

The following trends can be seen when looking at the breakdown of the distribution of BEPs for the different query categories: the more general queries, i.e. CO and essay topics, have a much larger number of BEPs than the more restricted queries, i.e. factual and CAS. 83.30% of BEPs belong to essay topic queries, compared to the 16.7% belonging to factual queries. This is expected since factual queries tend to have much shorter and more compact relevant fragments, which are typically associated with a single BEP, while more general queries tend to have lots of relevant fragments of various size, distributed over longer streches of texts, which may then be associated with multiple entry points. Similarly, CAS queries (32.65%) tend to focus the relevant information better and hence require less entry points than CO queries (67.35%), where relevant information may be spread over the entire play.

One of the most important characteristics of BEPs is that they represent an entry point to relevant information. This next analysis hence aims to examine the different BEP types with respect to the proportion of relevant information (measured at leaf node level) that is accessible from a given BEP. This is calculated as the percentage of relevant leaf nodes included in the cluster of the relevant text that the BEP represents, to the total number of leaf nodes contained within the cluster. For example, if the BEP is the first node in

[4]For example one query with a single BEP may distort the averages over the 43 queries as it would contribute 1/43-th of the overall statistics

	SBEP				PBEP	CBEP			Total
	-1	-M	-SL	Total		-1	-M	Total	
Factual	5.57	1.30	2.78	9.65	3.71	3.15	0.19	3.34	16.70
Essay topic	22.26	3.34	9.65	35.25	27.09	19.66	1.30	20.96	83.30
CO	19.11	3.90	6.68	29.69	20.78	15.58	1.30	16.88	67.35
CAS	8.72	0.74	5.75	15.21	10.02	7.24	0.19	7.42	32.65
CO.Factual	4.27	0.93	2.41	7.61	2.23	1.11	0.19	1.30	11.14
CAS.Factual	1.30	0.37	0.37	2.04	1.48	2.04	0	2.04	5.56
CO.Essay	14.84	2.97	4.27	22.08	18.55	14.47	1.11	15.58	56.21
CAS.Essay	7.42	0.37	5.38	13.17	8.54	5.19	0.19	5.38	27.09
All	27.83	4.64	12.43	44.90	30.80	22.81	1.49	24.30	100

Table 6: Distribution of BEPs according to BEP types for different query categories

the sequence of 5 relevant, 2 non-relevant, 1 relevant, 8 non-relevant and 4 relevant LINE nodes, then its proportion of relevant accessible leaf nodes is (5+1+4)/(5+2+1+8+4) = 50%. When the BEP is at a higher structural level, all leaf nodes that are contained within the higher level node are counted. For example, if the BEP is the second node in a sequence of 10 SPEECH nodes, then the total number of relevant leaf nodes within the sequence is divided by the total number of leaf nodes contained in the 10 SPEECH elements.

Table 7 shows the resulting scores (as percentages). At a first glance, the most salient finding is the overall difference between the ratio of relevant information accessible by the different type of BEPs: SBEPs are the most focused with 90% of the contained leaf nodes being relevant, while in general a third of the content of BEPs at higher structural levels is irrelevant.

Combining this information with the distribution of BEPs, one can conclude that the majority of users have a strong preference to the most specific, most focused components that contain the most amount of relevant information and the least amount of irrelevant content. This is since 44.9% of all BEPs contain only 10% irrelevant content, 30.8% contain 33% irrelevant content and 24.3% contain 38% irrelevant content.

Of the SBEP type entry points, the most focused nodes are those selected for factual queries and in particular for CAS.Factual queries, which is the most restrictive as to the location of relevant information.

It is interesting that when container nodes are voted as BEPs, participants tend to be more liberal with the inclusion of irrelevant content. Remember that Table 6 showed that such higher level nodes were usually chosen as BEPs for more general queries, e.g. essay topic and CO, where the inclusion of contextual information may contribute to the understanding of the content, rather than being strictly irrelevant. Based on this observation, an expectation here would be that when container nodes are selected as BEPs for factual queries, they would be more focused than those for essay topic queries. This is however not the case, in fact the findings show quite the opposite. For factual queries, on average, half of the container BEPs' content is actually irrelevant. Again, since the data here is based on a small sample size (a total of 20 nodes for the factual set and only 12 for the CO.Focussed set), outliers do have a larger impact on the overall results. Such an outlier is a `SPEECH` BEP of query no. 19, which contains 2 relevant and 30 irrelevant leaf nodes. However, the data does contain other BEPs whose irrelevant content is in the region of 80-30%. It is not clear why such BEPs were indeed chosen. A possible reason is that the contained relevant content's degree of relevance is not that different from the rest of the node's content. For example, if the highlighted fragments were not actually very relevant, participants may have felt that this did not justify an entry point just by itself.

Unlike PBEPs, the findings for combined BEPs follow the intuition and factual queries are characterised by more focused BEPs: 73% of the content being relevant. Another anomalie, however, is that CO.Factual queries have a higher score (79%) than CAS.Factual queries (69%). This again may be due to small sample size (there are only 2 CAS.Factual queries) or other currently unknown factors of the user behaviour that cannot be further analysed here.

Note that the surprising score of 17% for SBEP-SL in the CAS essay topic query category is a result of sparse data: there are only 2 samples in this set, both with atypical characteristics. Their effect on the overall scores is, however, neglegible. Other odd results are the less than 100% scores for SBEP-SL. This is due to a couple of strange BEP selections, where single non-relevant nodes were nominated as BEPs. This seems more of an issue related to disagreement between judges about relevance assessments.

5.0.3 *Related studies of BEP selection strategies*

The study in [3] identified similar BEP types: browsing BEPs (equivalent to SBEPs here) and container BEPs (equivalent to PBEPs). In a separate study in [17, 18], similar and more detailed investigations were carried out (although most statistics were calculated using macro-averaging). The aim of the analysis there was to investigate aspects of BEP that could then be used for automating BEP identification. The work in [18] defined an additional three BEP types: relevance judgment BEP (which is essentially the same as SBEP-SL), combination BEPs (same as CBEPs above) and context BEP, which are non-relevant nodes that are intended to provide contextual information for a relevant fragment. These were not separately identified in this study as they were too rare to provide sufficient sample data for analysis.

In addition to the analysis of BEP data obtained from the Shakespeare study, [18] also investigated the results of a

	SBEP				PBEP	CBEP			Total
	-1	-M	-SL	Total		-1	-M	Total	
Factual	94	99	93	95	53	73	71	73	81
Essay topic	86	80	100	89	68	60	64	60	75
CO	88	91	97	91	69	61	60	61	76
CAS	86	59	100	90	65	64	100	65	76
CO.Factual	93	99	92	93	46	80	71	79	82
CAS.Factual	98	100	100	99	63	69	0	69	78
CO.Essay	87	88	100	90	70	60	58	60	75
CAS.Essay	83	17	100	88	65	62	100	63	76
All	87	86	99	90	66	62	65	62	76

Table 7: Ratio of relevant and total leaf nodes accessible from a BEP, broken down by BEP types and query categories (given as %)

small study conducted on the INEX 2002 test collection. Their analysis showed that combination and container BEPs were hardly used by subjects participating in this test. In fact they claim that 55% of BEPs were of two new types: partial relevance judgement BEPs and so-called new BEPs. The former, accounting for 50% of BEPs, where defined as sub-parts of a relevance judgement. The example mentioned is that of a participant choosing a paragraph as BEP from a relevant section. This, however, seems to point to a methodological issue within the evaluation. The BEP types defined based on the Shakespeare data built on the notion of a smallest unit, i.e. the leaf nodes. BEPs hence could not be chosen at a lower level than this. The analysis of the INEX data, however, it seems was based on different principles, which raises the question whether the two studies could actually be compared reasonably. Unfortunately, no other studies exist yet that could provide an insight into what aspects may characterise a BEP within the INEX test collection.

6. CONCLUSIONS

The Shakespeare test collection, aimed for the evaluation of focussed retrieval approaches to structured document retrieval (SDR), was constructed based on the methodology described in [9] and resulted in a small (around 10MB) test collection (with around 180 000 XML elements). The test collection has proved especially suitable for experimentations regarding user's search behaviour in a focussed SDR environment [3, 10, 17, 18].

A finding concerning the analysis of the collected queries is their wide variation of complexity: from the simplest factual questions, through more general essay topics, to complex queries that are closer in nature to actual user tasks than search topics. The main result of this study regarding user queries is the evidence that both CO and CAS queries are in fact types of queries that are needed by real users in real information seeking situations. The use of structural constraints in queries appears as natural to novice users as the traditional use of CO queries. At the same time the use of CO queries confirms the need for their support by XML IR systems.

Conclusions regarding assessor agreement showed that while participants were likely to disagree about the exact location of the relevant information, they did in fact agree on the general area in which the answers to a query were to be found. The observed agreement statistics at leaf level were slightly worse than those reported for TREC in [21, 22]. In general, factual queries showed highest agreement and CAS queries the lowest.

A closer look at the relevance assessments revealed that the low level of agreement was partly due to assessors' varying implicit assumptions about how the retrieval results may be presented to users.

The BEP assessments were investigated with the aim to derive conclusions regarding users' preferences in what they consider would be the best document components that an XML IR system should return to them in response to a query. Assessor agreement results for BEPs showed that the concept of BEP is an intuitive one. The evidence found suggests that users prefer to be pointed directly at the most specific relevant information. If there are key relevant fragments, these are preferred as users would then browse around to obtain any necessary contextual information. BEPs were usually chosen at the level of the relevance assessments, i.e. leaf level, or one level up in the hierarchy. Similar findings were reported in the Tess study [7, 8], where in the majority of the tasks, users preferred entry points into the documentation that was equal to a relevant item.

In comparison, a study of BEPs for the INEX test collection, reported in [17, 18], showed that section nodes were most often preferred. It is however not clear if this may be due to the generality of the used queries or the different nature of the collection, where more context may be required. In addition, the different definition of BEP types in this study makes the comparison of the results across the different studies questionable.

7. REFERENCES

[1] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *SIGIR'00: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40, New York, NY, USA, 2000. ACM Press.

[2] Y. Chiaramella, P. Mulhem, and F. Fourel. A model for multimedia information retrieval. Technical Report FERMI ESPRIT BRA 8134, University of Glasgow, 1996.

[3] K. Finesilver and J. Reid. User behaviour in the context of structured documents. In F. Sebastiani,

editor, *ECIR*, volume 2633 of *Lecture Notes in Computer Science*, pages 104–119. Springer, 2003.

[4] N. Fuhr, M. Lalmas, S. Malik, and Z. Szlavik, editors. *Advances in XML Information Retrieval. Third Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2004), Schloss Dagstuhl, 6-8 December 2004*, volume 3493 of *Lecture Notes in Computer Science*. Springer, 2005.

[5] K. Hatano, H. Kinutani, M. Watanabe, Y. Mori, M. Yoshikawa, and S. Uemura. Keyword-based xml fragment retrieval: Experimental evaluation based on inex 2003 relevance assessments. In N. Fuhr, M. Lalmas, and S. Malik, editors, *INEX*, pages 81–88, 2003. http://inex.is.informatik.uni-duisburg.de:2003/proceedings.pdf.

[6] M. A. Hearst. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 9–16, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[7] M. Hertzum and E. Frøkjær. Browsing and querying in online documentation: A study of user interfaces and the interaction process. *ACM Trans. Comput.-Hum. Interact.*, 3(2):136–161, 1996.

[8] M. Hertzum, M. Lalmas, and E. Frøkjær. How are searching and reading intertwined during retrieval from hierarchically structured documents? In *INTERACT '01: Proceedings of the IFIP TC 13 International Conference on Human- Computer Interaction*, pages 537–544, Amsterdam, 2001. IOS Press.

[9] G. Kazai, M. Lalmas, and J. Reid. Construction of a test collection for the focussed retrieval of structured documents. In F. Sebastiani, editor, *Advances in Information Retrieval, Proceedings of the 25th European Conference on IR Research, Pisa, Italy*, volume 2633 of *Lecture Notes in Computer Science*, pages 88–103. Springer, April 2003.

[10] M. Lalmas and J. Reid. Automatic identification of best entry points for focused structured document retrieval. In *CIKM '03: Proceedings of the 12th international conference on Information and knowledge management*, pages 540–543, New York, NY, USA, 2003. ACM Press.

[11] S. Malik, G. Kazai, M. Lalmas, , and N. Fuhr. Overview of inex 2005. volume 3977 of *Lecture Notes in Computer Science*. Springer-Verlag, 2006.

[12] R. A. O'Keefe. If inex is the answer, what is the question? In Fuhr et al. [4], pages 54–59.

[13] J. Pehcevski and J. A. Thom. Hixeval: Highlighting xml retrieval evaluation, 2006.

[14] J. Pehcevski, J. A. Thom, S. M. M. Tahaghoghi, and A.-M. Vercoustre. Hybrid xml retrieval revisited. In Fuhr et al. [4], pages 153–167.

[15] B. Piwowarski and M. Lalmas. Providing consistent and exhaustive relevance assessments for XML retrieval evaluation. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 361–370, New York, NY, USA, 2004. ACM Press.

[16] B. Piwowarski, A. Trotman, and M. Lalmas. Sound and complete relevance assessments for XML retrieval. 2006. Submitted for publication.

[17] J. Reid, M. Lalmas, K. Finesilver, and M. Hertzum. Best entry points for structured document retrieval - part i: Characteristics. *Inf. Process. Manage.*, 42(1):74–88, 2006.

[18] J. Reid, M. Lalmas, K. Finesilver, and M. Hertzum. Best entry points for structured document retrieval - part ii: Types, usage and effectiveness. *Inf. Process. Manage.*, 42(1):89–105, 2006.

[19] T. Rölleke, M. Lalmas, G. Kazai, I. Ruthven, and S. Quicker. The accessibility dimension for structured document retrieval. In F. Crestani, M. Girolami, and C. Rijsbergen, editors, *Advances in Information Retrieval, 24th BCS-IRSG European Colloquium on IR Research, Glasgow, Scotland*, volume 2291 of *Lecture Notes in Computer Science*, pages 284–302. Springer, 2002.

[20] A. Trotman. Wanted: Element retrieval users. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, pages 58–64, 2005.

[21] E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *SIGIR'98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 315–323, New York, NY, USA, 1998. ACM Press.

[22] E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management*, 36(5):697–716, 2000.

Designing User Studies for XML Retrieval *

Miro Lehtonen
Department of Computer Science
P. O. Box 68
FIN–00014 University of Helsinki
Finland
Miro.Lehtonen@cs.Helsinki.Fi

ABSTRACT

Ever since the research on XML retrieval started, we have seen little cooperation between the researchers and the system developers in the field. Consequently, some of the issues that seem fundamental to the researchers are trivial to the developers. For example, the existence of the real users of XML retrieval is rarely questioned by those who make money selling such software. As an attempt to bridge the gap between the far too separated parties, we discuss some issues that have been subject to user studies in recent years and suggest improvements for them.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation*; H.5.2 [**Information Interfaces and Presentation**]: User interfaces

General Terms

Design, Human Factors

Keywords

XML Retrieval, search engine, user study, user interface

1. INTRODUCTION

The seemingly long societal distance between the real users of XML search engines, the developers and vendors of such systems, and the researchers in the field — including the INEX community — shows in the research questions the latter are trying to investigate about the first group. The controversies rarely surface as the researchers conduct their self-designed user studies where users from their own academic circles use a self-designed or other experimental system for XML retrieval. It is counter-productive that the assumptions that originate in the experimental implementations have such a strong influence on the user studies including the user tasks, the test environment, and the interpretation of the test material. It has been unclear whether the results of the user studies generalise, and if they do, we do not seem to know *how* they generalise. For example, we have trouble proving that the assumptions and conclusions hold for more than one document collection or more than one system for XML retrieval. Therefore, we are challenged by the fact that our user studies rarely lead to results that have an impact outside the academic world.

This paper has been inspired by the numerous misunderstandings and the terminological ambiguity the author has had to deal with, coming from the XML side of the world and speaking XML, and once again, approaching the scientists and scholars of Information Retrieval, like a visitor in their home field. The purpose of the paper is to bring up the differences in view between those who see the big picture around issues related to IR and those who understand the essence of XML, and, ultimately, to help bring those two sides closer to each other.

As user studies are a major source of controversy, we present arguments for and against the questions we are studying about the users of XML retrieval. In Section 2, we consider the frequently asked question about user preferences: Are XML elements better than whole XML documents? Section 3 is a response to a recent analysis where the structural hints in queries were regarded. Examples of the users of XML retrieval, who were never really lost, are introduced in Section 4. The kind of issues that are considered relevant according to the author of this paper are discussed in Section 5, followed by concluding remarks in Section 6.

2. XML ELEMENTS AND DOCUMENTS

It is a common argument to the benefit of XML retrieval systems that, instead of whole documents, we may also see document fragments in the list of results [4], or, alternatively, that we may start our navigation from a relevant entry point in the result document in addition to the beginning of the document [5]. In order to emphasise this advantage, we seek support for element retrieval systems in user studies where users show interest either in whole XML documents, or single XML elements, or both [7]. Without the support, the entire need for element retrieval systems could be questioned[1]. In a general setting, however, "ele-

*SIGIR 2006 Workshop on XML Element Retrieval Methodology August 10, 2006, Seattle, Washington, USA.

[1]See the Call for Papers of this workshop.

ments or documents?" seems to be an irrelevant research question.

Firstly, the XML constructs such as elements and documents are part of the technical implementation of the document collection. The same content may be stored at various levels of granularity so that we have an XML document for each article, each journal, or even a whole volume of journals. When talking about Enterprise Content Management (ECM) and XML, the modern documentation tends to consist of much smaller units where a couple of paragraphs constitute a whole XML document. Consequently, the producers of the content do not even know what kind of publications (those too are documents) their content will be part of [2]. When it comes to the users of XML retrieval, they do not know, nor do they care, how much content a single XML element or an XML document holds. Given an arbitrary answer from the result list to some query, the user can hardly know whether they are inspecting a whole XML document or a part of an XML document. It is all just content to the user.

Secondly, when content is stored in the XML format, we always retrieve XML elements, either small ones or big ones. If the content is stored in an XML database, the retrieved answers can be whole documents, as well, though not necessarily the same ones as stored in the database. The most common logical units of XML that are returned include single XML elements, sequences of XML elements, and whole XML documents. Nevertheless, the concept of a "whole XML document" is rather useless for the research on IR as it is merely a technical detail. In order to straighten up the problem with the wording, we may want to say that the size of the answers in XML retrieval is dependent on the query instead of being fixed to the size of a document in traditional document retrieval.

As it makes no sense to ask the users whether they prefer XML elements to whole XML documents, at least according to the presented argumentation, we may still ask them if the size of the answers is "too small", "too big" or "just right" [6]. In the experiments of the Interactive Track of INEX, the users are expected to assess each answer's need of context by selecting one of the following values for each answer: Broad, Exact, Narrow (Task C). The context is defined as the content of the source document that is not included in the answer. The relevance of this question too can be disputed. In a realistic setting, the user may know nothing about the context, or even that there is one outside the returned XML element. If we are given a paragraph that is extracted from a scientific article, we cannot always tell that it actually comes from a whole article. When an answers seems too small to satisfy the information need, there is no guarantee that any bigger answer such as the "parent element" is somehow more appropriate to the user, but, knowing the context might just make the otherwise good answer look too small (Narrow Answer). The user assessment is thus unjustly biased by the choices made in the design of the user study.

3. STRUCTURAL CONSTRAINTS

XML is sometimes called a "metalanguage" because the document structure including XML elements is expected to describe the character content (text). What makes searching XML documents different — and more interesting — than searching plain text or hypertext documents is that the queries may include conditions on the document structure, as if we were querying a textual database. The structural conditions have been introduced to the IR researchers in the Content-And-Structure (CAS) queries of INEX in the past few years. XQuery [12] is a common query language that supports such conditions, but a simpler language called NEXI [11] is used in the context of INEX.

For a good reason, the structural conditions have been given the role of serving as hints rather than requirements for the search engine: they are not necessary or even very useful when the search engines evaluate the official queries of the past INEX initiatives. Trotman and Lalmas go even further in their interpretation, according to which, "structural hints in queries do not help XML retrieval" [10]. They do mention that this might not hold for arbitrary document collections, but is it even true of the single test collection? Questioning the value of the structural hints is justified, but generalising the claim to all the queries or all types of "structural hints" is not. Examples will follow.

Trotman and Lalmas also suggest that the users be particularly bad at giving structural hints. By this claim, in fact, they imply that it is possible to give such structural hints that help XML retrieval in the context of INEX and the IEEE collection. In this paper, we have more faith in the searchers as we point our finger towards the document structures as we present our claim: **Whether the structural hints help or not depends on the document type of the XML documents.** To be more exact, specifying structural constraints is useful and even necessary when the structure describes the content. However, the element names in the INEX IEEE collection of scientific journals do not describe the content, but instead, they describe the document structure such as paragraphs, sections, and article bodies. The structural hints given by the user thus describe the size of the answer they expect. Why specifying the constraints rarely helps is because users cannot know how much content is required to answer their query in the particular test collection. If an entire article describes the topic the user was interested in, we can hardly return a section to the user that would summarise the entire article. It is thus better to let search engines determine the best granularity of the relevant answers.

The Lonely Planet document collection of the INEX Multimedia Track[2] serves as an example of a case where the structural hints are useful. For example, someone who is interested in taking "cold showers" may want to specify that the keyphrase is not found in the `weather` element but in other elements such as `activities` (or why not `amenities`?) instead. Medical patient records in XML format [3] are another good example. If we want to know how to cure "fever", we want to see that keyword in the `diagnosis` element and exclude the occurrences in the `complaint` and `side-effect` elements. The relevant answers would most likely be `treatment` elements. In these examples, the most natural interpretation of the structural conditions is to treat

[2]These documents describing travel destinations are also known as the WorldGuide.

them as strict requirements, which further emphasises their importance.

Studying whether the structural hints help or not might be interesting, but it is rather unclear whether the results would have any impact in practical applications. We take the same attitude towards user studies investigating whether users can or cannot specify structural conditions in a specific query language such as NEXI. Whether any query language is too complex for the users is not a real issue, because the users and the query languages never meet each other. In practice, the structural hints are next to trivial for the users to specify as the search interfaces accommodate the procedure. The search conditions are typically defined with checkboxes, lists, and input fields, as shown in the example in Appendix A. Wildcards, logical operators, and regular expressions are naturally supported, but not required, as the input can be directly inserted into an XQuery expression. When the user interface is appropriately designed and when the structure of the documents is consistent, the users that were not able to specify good structural hints for the INEX queries can most likely give the exact "structural hints" that they need.

4. LOST & FOUND: THE USERS

Studying what users think of experimental systems for XML retrieval is likely to lead to experimental results, but nothing more, as the setting is often artificial. Experimental systems that index test documents rarely have a true demand that has originated in a user community. Studying user behaviour with real-life systems is thus considerably more reliable as a source of useful results. The biggest challenge to the INEX community, so far, seems to be the lack of contacts with the users of real-life element retrieval systems [9]. Although XML retrieval is a rather young field of research, the vendors of such systems have been happily selling their products for years. Moreover, the earliest user studies date all the way back to 2000 [13] which is two years before the first round of INEX. Rather than trying to find the users, we are tempted to ask an even more interesting question: How did we lose sight of the users of XML retrieval?

The answers are not simple. First of all, the search engines were not called systems for XML retrieval until quite recently. Secondly, XML does not have any bigger role in the systems than that of the document format. The users never have to see any XML markup when they use such systems. Consequently, most users of today's XML retrieval systems are not aware of being ones. Six years ago, the users liked their XML retrieval systems only because "it uses XML" [13]. By the time the concept of XML retrieval was well established, the users were no longer excited by hearing the three letters; they only expect to have access to the relevant content of their XML repository.

The third cause of confusion is in the definition of an XML (element) retrieval system. To some, including the author, any search engine that indexes XML documents and returns the content to the user falls in the category of XML retrieval systems. To others, it is enough that the systems gives users access to incomplete documents which they call (XML) elements. However, as such systems do not require any XML technology in their implementations, the latter conception is somewhat questionable. What most people seem to agree about is that XML retrieval systems let users give structural hints about the searched documents and the returned answers. Regardless of our definition, the systems for XML retrieval are widespread.

In addition to the XML search engines that are used in private companies and enterprises, there are a number of such systems online that are available to public use. A common feature in the real-life systems is that they are not general-purpose XML search engines, but they specialise in indexing and searching specific document collections. The search engine and the indexing methods are developed together with the document type[3] as they both are a part of the document management system. A brief list of vendors and off-the-shelf software products that come with element-level search capabilities is presented in Appendix B.

A quite recent example of an XML retrieval system was developed for the New England Journal of Medicine.[4] The online user interface lets us search the full-text of various journals as well as medical case records and educational material. Only by using the system, it is nearly impossible to know that one is searching XML documents. However, all the users are what we are looking for — *users of XML retrieval*.

Another example of an online XML search engine gives access to the letters of Dolley Madison.[5] The user may browse the collection, as well as search for text, search by time period, people, topic, and location. Again, it is impossible to see that we are searching XML documents, but in this case, it is mentioned on the main page.

The user interface presented in Appendix A shows how useful the Document Type Definition (DTD) can be in the UI design. The users need not understand XML, DTDs, or query languages to be able to formulate accurate queries on XML documents. How to conduct user studies on these users is a real challenge unless we want to redesign the tests starting from the user tasks and ending at the interpretation of the results.

5. RELEVANT QUESTIONS

So far, we have questioned a whole lot of issues that the contemporary user studies address. In order to make the criticism constructive, we regard which issues are relevant enough to deserve more attention in the future user studies.

5.1 Assessing the size of the returned answer

If we use a system that returns entry points to relevant documents, it is not meaningful to assess the size of the answer because whole documents are returned. Nonetheless, if the system returns answers that are extracted from the source documents, we are interested in how good the system is, in the user's opinion, at determining the correct granularity of the answers. The Interactive Track of INEX already

[3] An XML DTD or an XML Schema development usually goes hand-in-hand with the development of the document type.

[4] http://content.nejm.org/

[5] http://rotunda.upress.virginia.edu:8100/dmde/

includes the assessment of the size of the document component in their Task C [6]. However, they instruct the users to estimate the size in terms of the context (the source document), which is everything but a user-oriented question for a user study. If the answer's being "too small" (Narrow) or "just right" depends on the content outside the answer, we are assessing the performance of the system. How satisfied the users are with the given standalone-type answer is not dependent on the context of the answer in the source document. To conclude, we want to know how content the users are with the size of the returned answers, but we need to assess the quality in absolute terms which a function of the need for its context is not.

5.2 Opinions on the search interface

The user interfaces of the operational XML retrieval systems online typically have a web form which the searcher first fills in and then submits to the server. The number of input fields varies, as well as the number of selections made by default. The search interfaces directed to public use do not allow the users to access the XML documents through any XML query language, but is the low-level access even desirable, as long as the users have a way to specify the structural constraints discussed in Section 3? It is still unclear to the author of this paper, what kind of users and what kind of search tasks would benefit from a different kind of a user interface for entering the query. Anyway, it is sensible to study the user's opinion on a user interface, even if only to improve the general UI design.

Although the query forms look similar from one system to another, there are big differences in how the search results are presented to the user. In a similar fashion to web search engines, we are often given a list of links along with a summary or metadata about the answer. Each link anchor may come with multiple options and targets. The targets of the links may include extracts from the source documents as well as XML fragments of various sizes. Thanks to the inherent nature of XML, the answers are simple to convert into HTML or PDF. What kind of browsing interfaces are the most suitable for the result lists is an open question as well as an interesting topic for a user study.

5.3 Comparative studies

Although some may argue that scientific articles are atomic units of retrieval [9], we also have users who presumably prefer smaller answers to their queries [7]. Rather than testing users on a single system for XML retrieval that gives them a choice between XML elements and whole documents, it is more sensible to let the users try out two different systems: 1) a traditional search engine, and 2) an XML retrieval system. If the users prefer having a choice of entry points to not having more than one, we can conclude that this aspect of XML retrieval is meaningful. The same applies to XML-aware systems that return XML fragments to the user instead of whole documents.

When these tests are performed on the INEX IEEE collection, however, the results do not automatically generalise to different XML collections. For example, even if users preferred whole articles to single paragraphs, we could not draw similar conclusions concerning the Lonely Planet collection. It is more tempting to assume similar preferences about other scientific literature, anyway. Despite being a popular research question, whether users prefer sections or articles does not really have anything to do with XML. User studies investigating the issue do not even require systems for XML retrieval.

Another way to study the benefits of XML retrieval is to compare different collections with each other instead of comparing different systems. For example, we may let the users search both the original Wikipedia documents and the converted XML documents. In the case of the Wikipedia documents, we see many similarities in the two versions. For example, both document collections can be segmented into small document fragments. The segmentation of plain text may even result in more natural segments than those that follow the boundaries of XML elements. The major difference, in general, would be that the structure of the XML documents can be included in the queries, whereas, querying the structure of the non-XML documents is far less trivial. In the particular case of the Wikipedia documents, though, the XML structure is not very useful, and the benefits of XML might not be so great.

6. CONCLUSIONS

We have presented a whole lot of arguments in the hopes of improving the potential impact of user studies. One of the key points to remember is that a typical user of an XML retrieval system does not know when they are searching XML documents. Moreover, because there is no one-to-one correspondence between the traditional documents and XML documents, the users can hardly appreciate the benefits of only being shown the relevant parts of the XML documents. They do appreciate seeing relevant content, though, and they dislike being shown irrelevant content. Furthermore, the users, who now might seem quite ignorant, are not aware of being in the process of specifying structural constraints for their query when they fill in the fields of an advanced-looking search form. The last concern is the user's ability to judge the technology behind the implementation. A user study may show that users appreciate certain functionality that XML retrieval systems offer, but the very users could not possibly judge the details specific to the implementation, including the use of XML. All these issues should be taken into account when designing user-oriented user studies for XML retrieval.

In this paper, we have also learnt other little details about XML. The structure of XML documents was originally designed to serve as metadata about the content. Including the structure in the queries should help the search engine find the relevant answers only as long as the structure describes the content. We should also keep in mind, that, for any textual documents, XML is the enabling technology rather than a straight jacket posing limitations.

7. REFERENCES

[1] ASTM International. *E2210-02 Standard Specification for Guideline Elements Model (GEM)-Document Model for Clinical Practice Guidelines*, 2003.

[2] D. Clark. Rhetoric of present single-sourcing methodologies. In *SIGDOC '02: Proceedings of the 20th annual international conference on Computer*

documentation, pages 20–25, New York, NY, USA, 2002. ACM Press.

[3] J. S. Hooda, E. Dogdu, and R. Sunderraman. Health level-7 compliant clinical patient records system. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 259–263, New York, NY, USA, 2004. ACM Press.

[4] J. Kamps, M. Marx, M. de Rijke, and B. Sigurbjörnsson. XML retrieval: what to retrieve? In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 409–410, New York, NY, USA, 2003. ACM Press.

[5] M. Lalmas and J. Reid. Automatic identification of best entry points for focused structured document retrieval. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 540–543, New York, NY, USA, 2003. ACM Press.

[6] B. Larsen, S. Malik, and A. Tombros. The interactive Track at INEX 2005. In *INEX*, volume 3977 of *Lecture Notes in Computer Science*, pages 398–410, 2006.

[7] B. Larsen, A. Tombros, and S. Malik. Is XML Retrieval Meaningful to Users? Searcher Preferences for Full Documents vs. Elements. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, Aug. 2006. To appear.

[8] R. Shiffman, G. Michel, A. Essaihi, and E. Thornquist. Bridging the guideline implementation gap: a systematic, document-centered approach to guideline implementation. *Journal of American Medical Information Association*, 11(5):418–426, Sep-Oct 2004.

[9] A. Trotman. Wanted: Element Retrieval Users. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*. Department of Computer Science, University of Otago, July 2005.

[10] A. Trotman and M. Lalmas. 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 in information retrieval*, Aug. 2006. To appear.

[11] A. Trotman and B. Sigurbjörnsson. Narrowed Extended XPath I (NEXI). In *Advances in XML Information Retrieval, Third International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2004), Dagstuhl Castle, Germany, 6–8 December 2004, Revised Selected Papers*, volume 3493 of *Lecture Notes in Computer Science*, pages 16–40. Springer, 2005.

[12] W3C. *XQuery 1.0: An XML Query Language, W3C Candidate Recommendation*, 3 November 2005. Available at `http://www.w3.org/TR/xquery/`.

[13] L. Weitzman, S. E. Dean, D. Meliksetian, K. Gupta, N. Zhou, and J. Wu. Transforming the content management process at ibm.com. In *CHI '02: Case studies of the CHI2002 / AIGA Experience Design FORUM*, pages 1–15, New York, NY, USA, Apr. 2002. ACM Press.

APPENDIX

A. AN OPERATIONAL USER INTERFACE

If guidelines for clinical practice were stored in a plain text format, finding relevant information would require highly sophisticated methods for Information Retrieval. Thanks to the metadata provided by the XML format, we can easily make the queries so accurate that simple term weighting methods are sufficient. Figure 1 shows the beginning of a user interface of such system for entering search terms for the query.

Besides search terms, the structure of the indexed documents can be taken advantage of. Figure 2 shows how simple it can be — for the particular document type. User interfaces for searching other kind of documents should be modified accordingly to be functional.

More multiple choices are shown in Figure 3. From these screenshots we can see that the content producers may include quite a lot of metadata about the guidelines they describe. The DTD of the collection is public [1, 8] and also available online[6].

B. VENDORS

Table 1 shows a non-comprehensive list of vendors providing support for XML Element Retrieval.

[6]http://www.astm.org/

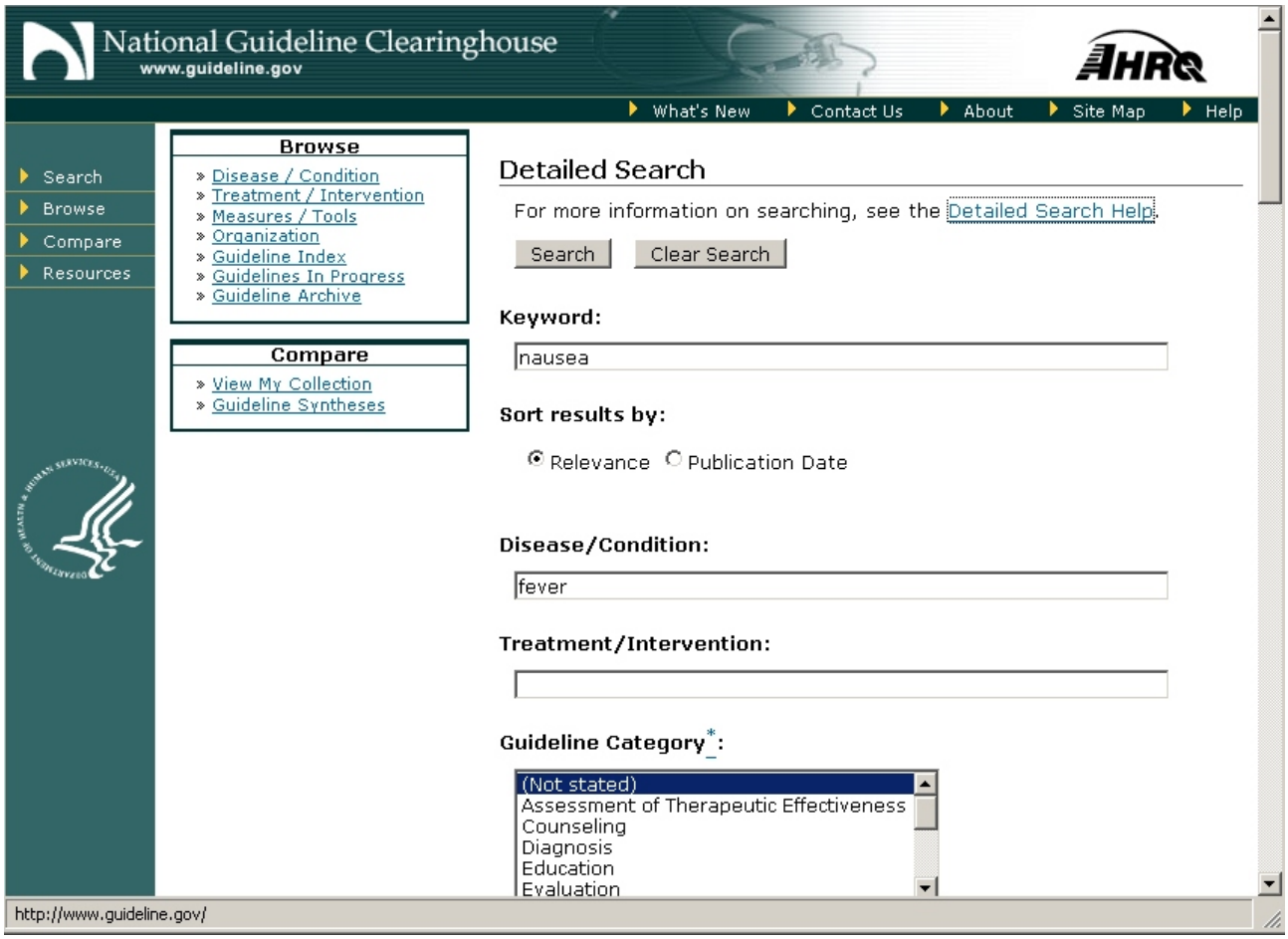


Figure 1: Input fields for entering keywords.

Vendor	Product	URL
Astoria Software	Astoria XML Content Management Platform	www.astoriasoftware.com
IBM	WebSphere Information Integrator OmniFind Edition	www.ibm.com
IXIASOFT	TEXTML Server	www.ixiasoft.com
Mark Logic Corp.	MarkLogic Server	www.marklogic.com

Table 1: Companies providing XML search engines followed by the product name.

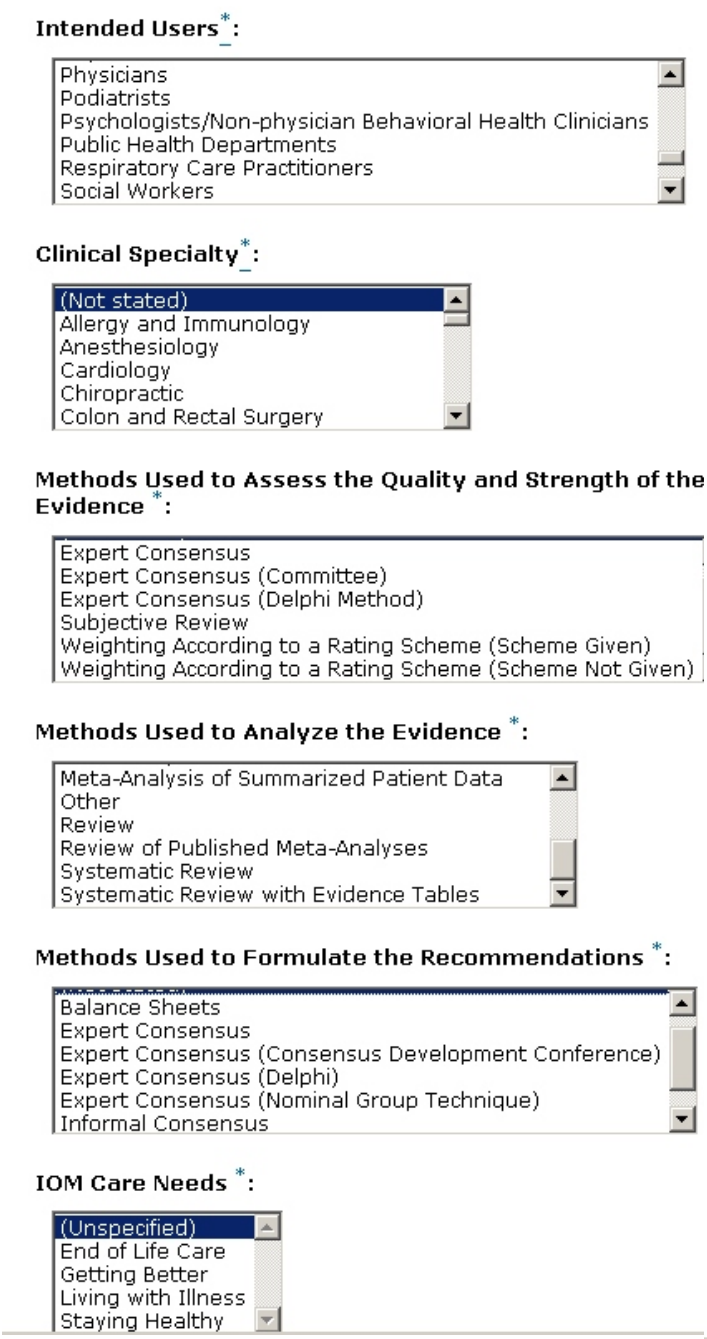


Figure 2: Lists helping users specify structural constraints for their query.

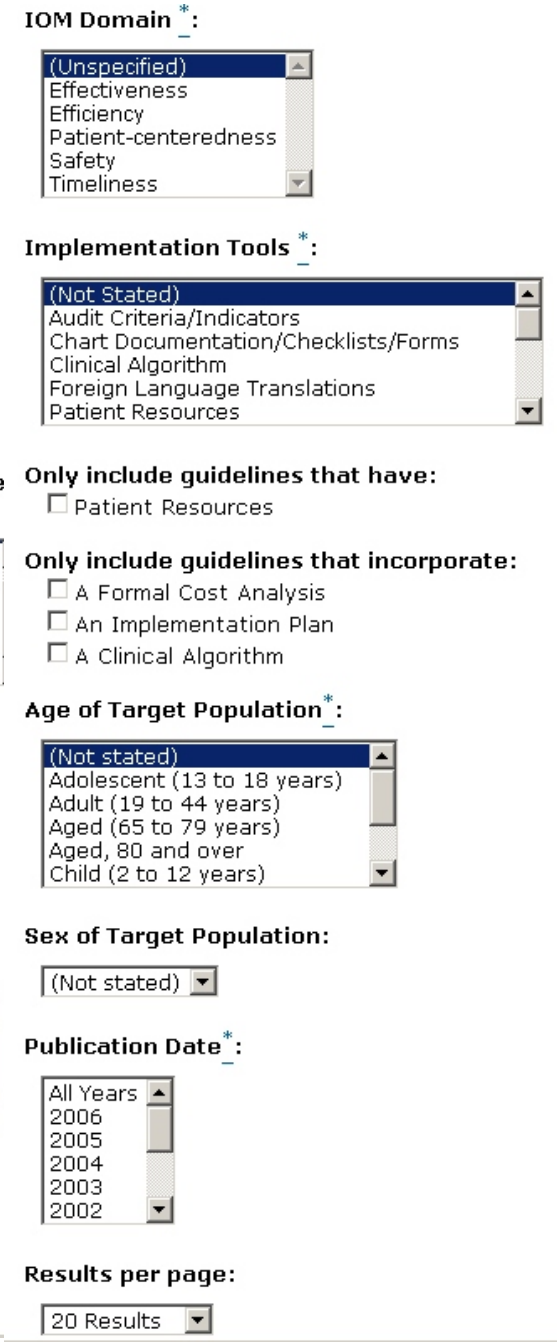


Figure 3: More options for making the query more precise.

Relevance in XML Retrieval: The User Perspective

Jovan Pehcevski
School of CS & IT
RMIT University
Melbourne, Australia
jovanp@cs.rmit.edu.au

ABSTRACT

A realistic measure of relevance is necessary for meaningful comparison of alternative XML retrieval approaches. Previous studies have shown that the current INEX relevance definition, comprising two dimensions based on topical relevance, is too hard for users to understand. In this paper, we propose and evaluate a new relevance definition that uses five-point scale to assess the relevance of returned elements. We perform a comparative analysis of the judgements obtained from interactive user experiments and the INEX 2005 relevance assessments to demonstrate the usefulness of the new relevance definition for XML retrieval.

1. INTRODUCTION

It is a commonly held view that *relevance* is one of the most important concepts for the fields of documentation, information science, and information retrieval [8, 14]. Indeed, the main purpose of a retrieval system is to retrieve units of information estimated as *likely to be relevant* to a user information need. To build and evaluate effective information retrieval systems, the concept of relevance needs to be clearly defined.

In traditional information retrieval, a binary relevance scale is often used to assess the relevance of an information unit (usually a whole document) to a user request (usually a query). The relevance value of the information unit is restricted to either zero (when the unit is not relevant to the request) or one (when the unit is relevant to the request). However, binary relevance is not deemed to be sufficient in XML retrieval, primarily due to the hierarchical relationships among the units of retrieval [13].

Each year since 2002, a new set of retrieval topics has been proposed and assessed by participants in INEX.[1] Analysing the behaviour of *assessors* when judging the relevance of returned elements may provide insight into possible trends within the relevance assessments [4, 13]. An interactive track was established for the first time in INEX 2004 [2] to investigate the behaviour of *users* when elements of XML documents (rather than whole documents) are presented as answers [15].

At INEX 2003 and 2004, two relevance dimensions — *Exhaustivity* and *Specificity* — were used to measure the extent that an element respectively *covers* and is *focused on* an information need. Each dimension used four grades to reflect how exhaustive or specific an element was: "none", "marginally", "fairly", and "highly". To assess the relevance of an element, the grades from each dimension were combined into a single 10-point relevance scale. In our previous work we have performed an empirical analysis of the two INEX 2004 relevance dimensions, where we have demonstrated that the highest level of agreement between the assessor and the users was at the end points of the relevance scale (representing highly relevant and non-relevant elements, respectively), and that the two INEX 2004 relevance dimensions were perceived as one (mostly because the two INEX dimensions are based on topical relevance) [11]. When the two INEX 2004 relevance dimensions were separately analysed, we observed that there was more overall agreement for *Exhaustivity* than for *Specificity*. The most likely reason for this was that both assessors and users seemed to have less understood an important property of the INEX 2004 *Specificity* dimension: an element should be judged as *highly specific* if it *does not* contain *non-relevant* information.

At INEX 2005 the relevance definition was slightly changed, and a highlighting assessment approach was used to gather the relevance assessments [1, 5]. A second interactive track was also established, comprising three tasks and two different XML document collections [6]. In Section 2 we briefly describe the INEX 2005 relevance definition, and present some findings about the assessor understanding of the two relevance dimensions. In Section 3 we propose a new definition of relevance for XML retrieval that uses a five-point scale to assess the relevance of returned elements. In Section 4 we demonstrate the usefulness of the new relevance scale through a comparative analysis of the judgements obtained from the INEX 2005 relevance assessments and those from users in the INEX 2005 Interactive track. We show that users perceive the new five-point relevance scale to be relatively simple, and that the grades of the new relevance

[1]INEX, INitiative for the Evaluation of XML Retrieval. `http://inex.is.informatik.uni-duisburg.de/`

SIGIR 2006 Workshop on XML Element Retrieval Methodology
August 10, 2006, Seattle, Washington, USA.

```
<file collection="ieee" name="co/2000/r7108">
<element path="/article[1]" exhaustivity="1" size="13556" rsize="5494"/>
<element path="/article[1]/bdy[1]" exhaustivity="1" size="9797" rsize="4594"/>
<element path="/article[1]/bdy[1]/sec[1]" exhaustivity="1" size="1301" rsize="409"/>
<element path="/article[1]/bdy[1]/sec[1]/p[1]" exhaustivity="1" size="531" rsize="408"/>
<element path="/article[1]/bdy[1]/sec[2]" exhaustivity="1" size="2064" rsize="2064"/>
<element path="/article[1]/bdy[1]/sec[2]/st[1]" exhaustivity="?" size="30" rsize="30"/>
<element path="/article[1]/bdy[1]/sec[2]/p[2]" exhaustivity="1" size="738" rsize="738"/>
<element path="/article[1]/bm[1]" exhaustivity="1" size="3267" rsize="900"/>
<element path="/article[1]/bm[1]/app[1]" exhaustivity="1" size="2085" rsize="900"/>
<element path="/article[1]/bm[1]/app[1]/p[3]" exhaustivity="1" size="438" rsize="438"/>
</file>
```

Figure 1: A sample from the INEX 2005 CO topic 203 relevance assessments for the relevant file `co/2000/r7108`. For each judged element, `exhaustivity` shows values for *Exhaustivity* (possible values `?`, `1`, or `2`), `size` denotes the element size (measured as total number of contained characters), while `rsize` shows the actual number of highlighted characters.

scale can easily be deduced from the amount of highlighted text in the relevant elements.

2. INEX 2005 RELEVANCE

The highlighting assessment task used at the INEX 2005 ad hoc track to gather relevance assessments for the retrieval topics had three main steps [5]. The assessor was first required to highlight the relevant content in each returned article. The assessment tool automatically identified the elements that enclosed the highlighted content, and the assessor was then asked to judge the *Exhaustivity* of these elements, and of all their ancestors and descendants. Last, the tool automatically computed the *Specificity* as the ratio of highlighted to fully contained text. The highlighting assessment task was also used at the INEX 2005 multimedia (MM) track, with the difference that the assessor was not asked to judge the *Exhaustivity* of the elements that contained highlighted content [17].

Figure 1 shows a sample of the relevance assessments obtained for the INEX 2005 Content Only (CO) topic 203. For each judged element, `exhaustivity` shows the *Exhaustivity* value of the element, with possible values of `?` (too small), 1 (partially exhaustive), and 2 (highly exhaustive); `size` denotes the total number of characters contained by the element; and `rsize` shows the actual number of characters highlighted by the assessor.

To measure the *relevance* of an element, a quantisation function is used to normalise the values obtained from the two INEX 2005 relevance dimensions [3]. For example, if the observed `exhaustivity` value is 1 and both values for `size` and `rsize` are the same (see Figure 1), the element is deemed as *highly specific* but only *partially exhaustive* [5].

To examine the extent to which the assessors understand the two INEX 2005 relevance dimensions, we have performed an analysis of the level of assessor agreement on the five topics that were double-judged at INEX 2005 [10]. The results show that there is good reason to ignore the *Exhaustivity* dimension during evaluation, since it appears to be easier for assessors to be consistent when highlighting relevant content than when choosing one of the three exhaustivity values [10].

This suggests that a much simpler relevance scale would be a better choice for evaluation in INEX and XML retrieval in general. Indeed, in their analysis of relevance judgements obtained from the users of the INEX 2004 Interactive track, Pharo and Nordlie [12] also observed the following: "A combined measure of relevance with so many alternatives as the one used in this experiment proves difficult for the searchers to relate to. In further experiments it might be fruitful to use another scale and resort to two separate assessments". In the next section we propose one such relevance scale.

3. A NEW DEFINITION OF RELEVANCE FOR XML RETRIEVAL

In this section we present a new relevance definition for XML retrieval. We describe the aspects and the two dimensions of the new relevance definition, and its five-point relevance scale. To demonstrate the simplicity of the new relevance scale, we also analyse user feedback gathered from the INEX 2005 Interactive track.

3.1 Aspects and dimensions

We base our new relevance definition on three aspects:

- There should be only *one* dimension of relevance based on *topical relevance*;
- The first relevance dimension should use a *three-graded* relevance scale, which will determine whether an XML element is either *highly relevant*, *relevant*, or *not relevant* to an information need; and
- There should be a *second* dimension of relevance, based only on the intrinsic hierarchical relationships among the XML elements.

Using only one topical relevance dimension allows the new relevance definition to be more intuitive than the INEX 2004

and INEX 2005 relevance definitions, which have two relevance dimensions based on topical relevance.

The first relevance dimension is inspired by our analysis of the level of agreement between the assessor and the users on the INEX 2004 CO topics, where the highest level of agreement was shown to be on *highly relevant* and on *non-relevant* elements [11]. However, in addition to the above two grades we also allow for a third relevance grade, *relevant*, to be incorporated in our first relevance dimension. This is supported by the fact that — to explore the effect of incorporating only highly relevant documents in the retrieval evaluation — most recent web tracks in TREC have adopted a similar three-point scale based on topical relevance [18].

The second dimension of relevance, as introduced in the third aspect above, is based only on the hierarchical relationships which are intrinsic to XML documents. O'Keefe [9] analyses some properties of the INEX 2004 IEEE document collection, and finds that elements that are highly coupled to their context are more difficult to judge than elements with low coupling. In this scenario, what matters most is "not how big the fragments are but how tightly they are coupled to their context" [9]. O'Keefe also argues that the usefulness of the XML retrieval task would also depend on the size of the retrieved information units; indeed, the appropriate units of retrieval should be self-contained, with a reasonable size, and at the same time with some coupling to their containing documents. Trotman [16] also examines these properties in detail.

We follow the above reasoning and allow three grades for our second relevance dimension: *just right*, *too large*, and *too small*. An XML element is *just right* if it is reasonably self-contained, and at the same time has enough coupling to be bound to its containing XML document. Alternatively, the element can be either *too large* or *too small*. An XML element is *too large* if it is reasonably self-contained, but it is either too big to be examined as an answer, or its coupling is so low that it can represent a free-standing XML document. An XML element is *too small* if it is not self-contained and its content is highly dependent on the context (high coupling), which makes it too small to be examined as an answer.

This second dimension of relevance is similar to *document coverage* used in INEX 2002 [4]. Indeed, document (or component) coverage was used as a relevance dimension in INEX 2002 to measure how specific (or focused) the unit of retrieval is to the information need. In a similar way to our second dimension, some aspects of document coverage depend on the context of the element; indeed, for a *too small* element Kazai et al. state that "the component is too small to act as a meaningful unit of information when retrieved by itself" [4]. However, the other two relevance grades, *too large* and *just right*, were not explicitly captured by the document coverage relevance dimension.

3.2 Relevance scale

As described above, our new relevance definition uses two *dimensions* to calculate the assessment score of an XML element.

	Questions	
Value	Q4.5	Q4.6
Mean	**2.51**	**2.96**
Minimum	1	1
Maximum	5	5
Median	2	3
StDev	1.27	1.29

Table 1: Analysis of responses on questions Q4.5 and Q4.6 gathered from 29 users that participated in Task C of the INEX 2005 Interactive track. For both questions, users were required to choose from five available answers, ranging from 1 ("Not at all") to 5 ("Extremely"). Mean average values obtained for each question are shown in bold.

The first relevance dimension determines the extent to which an XML element *contains relevant information* for the search task. It can take one of the following three values: *highly relevant*, *relevant*, or *not relevant*. The second relevance dimension determines the extent to which an XML element *needs the context* of its containing XML document to make full sense as an answer. It can take one of the following three values: *just right*, *too large*, or *too small*.

Thus, the final assessment score of an XML element can take one of the following five nominal values:

- **Exact Answer** (EA), if-and-only-if the XML element is *just right* and *highly relevant*;
- **Partial Answer** (PA), if-and-only-if the XML element is *just right* and *relevant*;
- **Broad Answer** (BA), if-and-only-if the XML element is *too large* and either *relevant* or *highly relevant*;
- **Narrow Answer** (NA), if-and-only-if the XML element is *too small* and *highly relevant*; and
- **Not Relevant** (NR), if the XML element does not cover any of the aspects of the information need.

To demonstrate that the above scale is not hard for users to understand, next we present analysis of the user responses obtained from the questionnaires collected for Task C of the INEX 2005 Interactive track.

3.3 User satisfaction

To measure the user satisfaction while using the new five-point relevance scale, users were asked to provide answers to the following two questions:

- Was it hard to understand and use the five-point relevance scale? (question Q4.5)
- Would it have been better if a simpler relevance scale was used instead? (question Q4.6)

For both questions, users were required to choose from five available answers, ranging from 1 ("Not at all") to 5 ("Extremely"). Table 1 shows an analysis of the responses gathered from 29 users for the two questions. The relatively

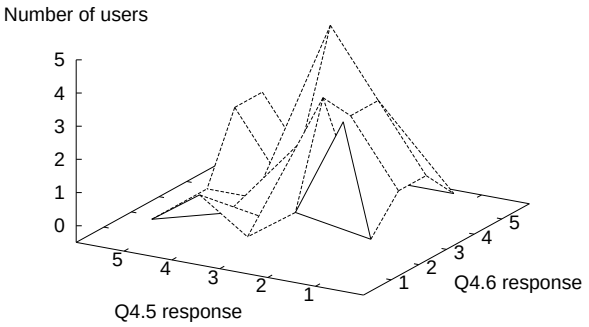


Figure 2: A 3D histogram of user responses on Q4.5 and Q4.6.

low mean average value (2.51) of responses to question `Q4.5` shows that users had little difficulty in understanding the new five-point relevance scale. At the same time, the mean average value of responses to question `Q4.6` (2.96) indicates that it was not really necessary to have a simpler relevance scale than the one used.

Figure 2 shows a more detailed analysis of the user responses to questions `Q4.5` and `Q4.6`, allowing us to explore whether there is any correlation between the responses to the two questions. We find that for question `Q4.5`, around 83% of the users chose one of the first three answers (1, 2, or 3). Of these, the largest number of users (38%) chose answer 2, while 24% and 21% of the users chose answers 3 and 1, respectively. Around 67% of the users who chose answer 1 for question `Q4.5` also chose the same answer for question `Q4.6`. The correlation is similar for answer 2, where the highest percentage of users who chose this answer for question `Q4.5` also chose the same answer for question `Q4.6`. These statistics show that users participating in Task C of the INEX 2005 Interactive track did not perceive the new five-point relevance scale to be hard to use.

4. EXPERIMENTS WITH THE NEW RELEVANCE DEFINITION

In this section, we present experiments that demonstrate the usefulness of our new relevance definition for XML retrieval. We first compare the new relevance scale to the one used in INEX 2005, and design a mapping between their respective relevance grades. We then present a performance analysis of simulated runs that use this mapping to construct their answer elements.

4.1 Comparison to the INEX 2005 relevance

Three tasks were explored in the INEX 2005 Interactive track [6]:

- Task A, where users searched three topics using a common baseline system with the INEX 2005 IEEE XML document collection (which was also used in the INEX 2005 ad hoc track);
- Task B, where groups with a working interactive XML retrieval system could test their system against the baseline system; and
- Task C, where users searched four topics using alternative system with the Lonely Planet XML document collection (which was also used in the INEX 2005 MM track).

In the following analysis, we focus on results obtained from Tasks A and C.

Task A judgements

For Task A, six topics grouped in two categories (General and Challenging) were selected for users, who were required to choose and search on only one topic per category. The six topics were derived from selected topics used in the INEX 2005 ad hoc track. We analyse relevance judgements obtained from a number of users for topics G1 (21 users) and G2 (18 users) of the General topic category, and relevance judgements for topics C2 (17) and C3 (26) of the Challenging category. We chose these four topics as all of them have corresponding assessor judgements available,[2] which makes it possible to analyse and compare the extent to which *both* assessors and users perceived the relevant answers for those topics. A simple three-point relevance scale was used by users of Task A, with the following values: `Relevant` (2), `Partial` (1), and `Not Relevant` (0). This relevance scale closely reflects the one used for the INEX 2005 *Exhaustivity* dimension. Our aim in the following analysis is to deduce a relationship between the two points of this scale that are assigned to *relevant* elements by users and the actual judgements assigned to the same elements by assessors.

Table 2 shows a statistical analysis of the overall distribution of user and assessor judgements across the four topics. For a relevance grade (`Relevant` or `Partial`), the `Total` values show the total number of (non-zero) elements judged by users across the four topics. Of these elements, the `MA` values show the number of those elements that were also mutually agreed to be non-zero by the assessor. The `E2`, `E1`, and `E?` values show the actual distribution of assessor judgements on the `MA` elements. For example, of the total 486 elements judged as `Relevant` by users, 352 were also judged as having non-zero relevance by assessors (denoted as `MA`). However, assessors did not always agree that these elements were `Relevant` (denoted as `E2` in the assessor judgements). In fact, 256 of the 352 `MA Relevant` elements were judged by assessors as `E2`, 96 were judged as `E1`, while none were judged as `E?` (too small). The `Agreement` values show the actual agreement between users and assessors on a relevance grade (for example, the overall agreement for the `Relevant` grade is $256/352 = 73\%$). As shown in the table, for a relevance grade we also measure the proportion of the relevant information contained by the agreed `MA` elements (`av_prel`) along with the corresponding standard deviation (`StDev`).

From the numbers shown in Table 2 we observe that, first, the overall agreement between assessors and users seems

[2]We used the relevance assessments that belong to the INEX 2005 CO topics 235 and 241 for topics G1 and C2, and those that belong to the INEX 2005 VVCAS topics 256 and 257 for topics C3 and G2, respectively.

User judgements	Non-zero		Assessor judgements					
	Total	MA	E2	E1	E?	av_prel	StDev	Agreement
Relevant	486	352	256	96	0	**0.57**	**0.32**	**0.73**
Partial	388	202	142	60	0	**0.49**	**0.27**	**0.30**

Table 2: Statistical analysis of the overall distribution of user and assessor judgements calculated across the two General (G1 and G2) and the two Challenging (C2 and C3) topics used in Task A of the INEX 2005 Interactive track.

to be higher for `Relevant` than for `Partial` *relevant* elements (73% compared to 30%); and second, the proportion of relevant information contained by the `Relevant` elements seems to be larger than for `Partial` elements (57% compared to 49%). However, these observations should be treated with care, since results from only four topics are used in this analysis.

The first observation seems to be in line with our previous finding on the INEX 2004 topics, where highly relevant answers were perceived better than partially relevant answers [11]. The second observation allows for a mapping to be established between the proportion of relevant information contained by a relevant element and the two grades, exact (`EA`) and partial (`PA`), that can be assigned to the relevant element using our five-point relevance scale. However, this does not provide any indication as to how broad (`BA`) and narrow (`NA`) elements should be mapped. Intuitively, from their definition we expect the `NA` elements to be the smallest in size and to contain the highest proportion of relevant information. Likewise, the `BA` elements should be the largest in size, and should contain the smallest proportion of relevant information.

We now explain how these expectations are validated by comparing the relevance judgements provided by users in Task C of the INEX 2005 Interactive track to the relevance assessments obtained from the INEX 2005 MM track.

Task C judgements

For Task C, eight topics — some derived from the INEX 2005 MM track topics — were arbitrarily grouped in two categories. Users were asked to choose and search on two topics in each category, and assess relevance using our five-point relevance scale. We analyse relevance judgements obtained from a number of users for topics LP1 (11) and LP2 (18) of the first topic category, and relevance judgements for topics LP5 (22) and LP7 (13) of the second category. These four topics also have assessor judgements available.[3]

Table 3 shows a statistical analysis of the overall distribution of user and assessor judgements calculated across the four topics. We observe that the number of user judgements is highest for the broad (`BA`) elements, and that these elements also have the highest number of mutually agreed relevant (`MA`) elements. As expected, on average the `BA` elements contain a very small proportion of relevant information (9%), and, for most of the mutually agreed `BA` elements, the proportion of found relevant information falls in the range 0%–32%. For the `EA` elements, the average proportion of relevant information is similar to that observed for Task A (Table 2), whereas for `PA` and `NA` elements we observe a different proportion of relevant information than that reported (and expected) previously. This can be attributed to the very low number of mutually agreed relevant elements.

[3]We used the relevance assessments for INEX 2005 MM topics 4 and 21 for topics LP1 and LP2, and for INEX 2005 MM topics 6 and 25 for topics LP5 and LP7, respectively.

User judgements	Non-zero		Assessor judgements	
	Total	MA	av_prel	StDev
Exact (EA)	59	17	**0.59**	**0.40**
Partial (PA)	93	9	**0.22**	**0.37**
Broad (BA)	120	39	**0.09**	**0.23**
Narrow (NA)	66	5	**0.55**	**0.50**

Table 3: Statistical analysis of the overall distribution of the user and assessor judgements calculated across four topics (LP1, LP2, LP5, and LP7) used in Task C of the INEX 2005 Interactive track.

In light of these statistics, a *reasonable* mapping between the continuous relevance scale of the INEX 2005 *Specificity* dimension and our five-point relevance scale would be as follows:

1. `EA` $\in (0.66, 1.00]$
2. `PA` $\in [0.33, 0.66]$
3. `BA` $\in (0.00, 0.33)$
4. `NA` $= 1.00$
5. `NR` $= 0.00$

In this mapping, there may be cases where both `EA` and `NA` elements are mapped as highly specific (1.00) elements. This property — illustrated in Figure 3 — is an important property of the above mapping, which as we discuss next primarily ensures to correctly identify the `NA` elements.

Figure 3 shows how the proposed mapping can be used to identify the four types of answer elements from the sample of relevance assessments for document `co/2000/r7108` of the INEX 2005 CO topic 203 (previously shown in Figure 1). The figure shows 10 relevant elements, and for each element the number in parentheses shows the proportion of contained relevant information. An element is identified as a `NA` element if it contains only relevant information (1.00) *and* at the same time its parent also contains only relevant information. There are two such elements shown in Figure 3 (`st[1]` and `p[2]`). However, although two elements, `sec[2]`

Value	CO			VVCAS		
	Total (elements)	av_size (chars)	av_prel	Total (elements)	av_size (chars)	av_prel
EA						
Mean	**332**	**1 145**	**0.98**	**572**	**1 960**	**0.98**
Minimum	17	155	0.95	23	29	0.90
Maximum	1 568	7 250	1.00	3 440	9 329	0.99
Median	269	800	0.98	375	965	0.98
StDev	355	1 318	0.01	693	2 191	0.02
PA						
Mean	**61**	**6 369**	**0.48**	**70**	**10 556**	**0.48**
Minimum	1	489	0.43	3	81	0.44
Maximum	271	26 379	0.55	295	40 798	0.59
Median	32	2 969	0.47	48	5 636	0.48
StDev	73	7 374	0.02	64	10 161	0.03
BA						
Mean	**204**	**19 367**	**0.11**	**186**	**25 351**	**0.13**
Minimum	13	10 225	0.08	16	8 371	0.03
Maximum	995	39 345	0.17	615	47 955	0.19
Median	105	17 054	0.11	130	23 303	0.12
StDev	238	6 933	0.02	150	10 789	0.04
NA						
Mean	**1 635**	**92**	**1.00**	**5 493**	**97**	**1.00**
Minimum	13	9	1.00	1	9	1.00
Maximum	13 994	272	1.00	44 600	283	1.00
Median	234	75	1.00	2 318	85	1.00
StDev	3 252	59	0.00	9 056	70	0.00

Table 4: Statistical analysis of the distribution of EA, PA, BA and NA relevant elements across the 29 CO and 34 VVCAS topics at INEX 2005. For a relevance grade, the Total values show the actual number of relevant elements that belong to that grade, while av_size and av_prel represent averages for the size of the relevant elements (in characters) and the proportion of relevant information contained by the relevant elements, respectively. Mean average values (calculated across all the CO or VVCAS topics) are shown in bold.

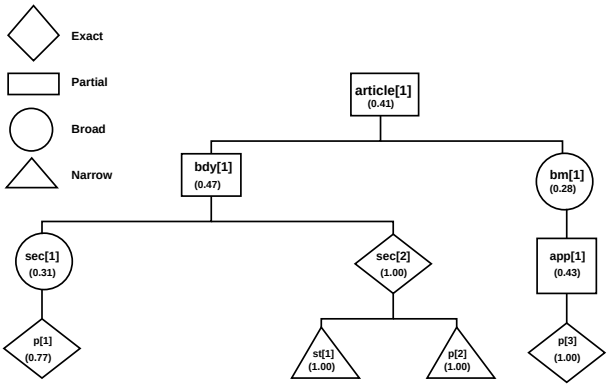


Figure 3: Identifying Exact, Partial, Broad, and Narrow answer elements from the relevance assessments sample that belongs to file co/2000/r7108 of the INEX 2005 CO topic 203. For each element, the number in parentheses shows the proportion of contained relevant information.

and p[3], also contain only relevant information, both are nevertheless identified as EA elements. The above example also shows that full article elements need not always be identified as BA elements; indeed, it is the proportion of contained relevant information in an element that determines its element type. Next, we use the proposed mapping and the INEX 2005 relevance assessments to find the actual distribution of the four element types across the INEX 2005 CO and Vague Content And Structure (VVCAS) topics.

INEX 2005 CO and VVCAS judgements

Table 4 shows a statistical analysis of the distribution of EA, PA, BA and NA relevant elements across the 29 CO and 34 VVCAS[4] topics at INEX 2005, when using the proposed mapping. As expected, the assessment trends are clear for both types of topics: the NA elements are the most common, the smallest in size, and contain only relevant information. The PA elements are the least common elements, while the BA elements are the largest in size, and contain the smallest proportion of relevant information. The EA elements are smaller in size than the PA elements, but contain higher proportion of relevant information.

To investigate the relationship between the four relevance grades and the three values of the INEX 2005 *Exhaustivity* dimension, we also analyse the distribution of the three *Exhaustivity* values across the four types of relevant elements. Table 5 shows this distribution, which is calculated separately for the INEX 2005 CO and the VVCAS topics. We observe that for the INEX 2005 CO topics the majority of EA elements were judged as partially exhaustive (E1), while for the INEX 2005 VVCAS topics most of the EA elements were judged as too small. This is somewhat surprising, showing that (on average) INEX 2005 assessors considered the

[4]We analyse relevance assessments for both parent and child VVCAS topics.

Value	CO				VVCAS			
		Exhaustivity				Exhaustivity		
	Total	E2	E1	E?	Total	E2	E1	E?
EA								
Mean	332	0.16	**0.48**	0.36	571	0.19	0.35	**0.46**
PA								
Mean	61	0.32	**0.63**	0.05	70	0.35	**0.57**	0.08
BA								
Mean	204	0.27	**0.69**	0.04	186	0.28	**0.68**	0.04
NA								
Mean	1 635	0.08	0.11	**0.81**	5 493	0.02	0.07	**0.91**

Table 5: Distribution of the three Exhaustivity values across the EA, PA, BA and NA relevant elements found for the 29 CO and 34 VVCAS topics at INEX 2005. For each of the four types of relevant elements, the Total values show the actual number of relevant elements, while E2, E1 and E? represent values for the proportion of those relevant elements that were assigned a corresponding Exhaustivity value. The highest values are shown in bold.

elements that contain most of the highlighted content to either discuss only some aspects of the underlying information need or to be too small. The partially exhaustive elements also represent the majority in both cases of PA and BA elements, while not surprisingly, most of the NA elements were correctly judged to be too small.

4.2 Performance analysis

In the following, we aim at investigating which of the four element types yields the best value in retrieving (non-overlapping) relevant information, which we believe could represent valuable knowledge in tuning the XML retrieval system parameters for optimal performance. We use the INEX 2005 CO topics to evaluate the performance of six simulated runs, four of which were created by only considering relevant elements that belong to the corresponding four element types (EA, PA, BA, and NA). The fifth run contains all the (overlapping) relevant elements found for the INEX 2005 CO topics (FullRB). To also investigate the XML retrieval performance when only the highlighted passages are units of retrieval, the sixth simulated run was created such that it contains (provisional) elements with sizes that strictly match the sizes of the corresponding passages.

For each run and an INEX 2005 topic, at most 1 500 elements were considered in the final answer list, where retrieved units were ranked in descending order according to the harmonic mean between precision (the proportion of relevant information to all the information retrieved from the element) and recall (the proportion of relevant information retrieved from the element to all the relevant information found for the topic). Overlapping answer elements were allowed in the answer lists of the five element runs. We use the HiXEval evaluation metric to measure the retrieval performance [10], with a parameter setting that penalises the retrieved overlapping relevant information among elements. A *system*-oriented retrieval task is considered for this performance analysis, where runs are rewarded if they retrieve as much non-overlapping relevant information as possible (high recall), without also retrieving a substantial amount of non-relevant information (high precision).

The graph in Figure 4 shows the retrieval performance of the six simulated runs. Perfect retrieval performance is achieved

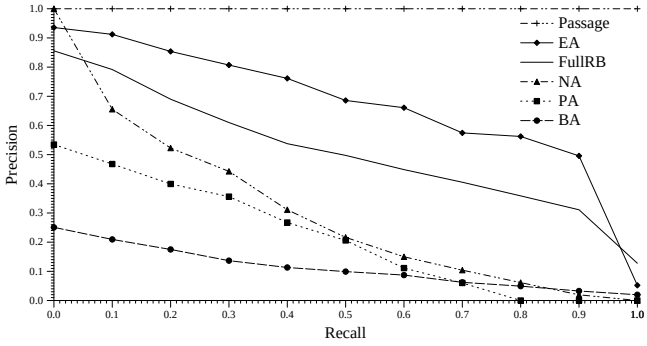


Figure 4: Performance evaluation of the six simulated runs on the 29 INEX 2005 CO topics using the HiXEval evaluation metric.

with the passage run, the EA run performs the best among the five element runs, while the BA run, which only contains broad answer elements, performs the worst. When the performance of the FullRB run is compared to that of the other element runs, we observe that the EA run performs better than FullRB. This shows that, when overlap is considered by HiXEval, better value in retrieving relevant information is achieved by identifying the (overlapping) exact answers, and not by retrieving all the (overlapping) relevant elements. Of the other two simulated runs, the NA run performs better than the PA run. Two factors influence this performance behaviour: first, as shown in Table 4 the average number of NA elements across the INEX 2005 CO topics is approximately 27 times that of PA elements, which allows for the NA simulated run to achieve higher overall recall than that achieved by the PA run; and second, the proportion of retrieved relevant information from the NA elements is always higher than that retrieved from the PA elements, which also leads to higher overall precision for the NA run.

The system-oriented retrieval task highlights the importance of identifying the *exact* answer elements. Indeed, the above knowledge that — of all the relevant elements retrieved for this task — the EA elements bring the best value in retrieving relevant information could influence the choice of tuning the XML retrieval system parameters for optimal performance.

5. CONCLUSIONS

In this paper we have presented an empirical analysis of what the experience of assessors and users suggests about how *relevance* should be defined and measured in XML retrieval. We have proposed a new relevance definition that is founded on results obtained from interactive XML retrieval experiments, and which uses a five-point relevance scale to assign an assessment score for an answer element.

There is a recent argument that a complex relevance scale may lead to an increased level of obtrusiveness in interactive user environments [7]. We have demonstrated that the new relevance scale was successfully used in Task C of the INEX 2005 Interactive track, where users did not find it to be very hard to use.

By analysing results from the topics judged by the assessors in INEX 2005 and by the users participating in the INEX 2005 Interactive track, we have been able to empirically establish a mapping between the continuous scale used by the *Specificity* dimension at INEX 2005 and our new five-point relevance scale. This mapping has allowed us to analyse the distribution of the four types of relevant elements in the INEX 2005 relevance assessments. We have presented an analysis of the performance of four simulated runs, each containing elements that belong to one of the four element types, and have shown that identifying and retrieving exact answer elements yields the best value in retrieving relevant information.

The performance evaluation shown in the last section is a *system*-oriented than a *user*-oriented evaluation. We plan to experiment with different types of relevance assessments, which may reflect different models of user behaviour, to more closely investigate whether or not the user model influences the best value in retrieving relevant information.

Acknowledgements

We thank James Thom, Saied Tahaghoghi, and anonymous reviewers for their comments on earlier drafts of this paper.

6. REFERENCES

[1] N. Fuhr, M. Lalmas, S. Malik, and G. Kazai, editors. *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005), Dagstuhl 28-30 November 2005*, volume 3977 of *Lecture Notes in Computer Science*. Springer-Verlag, January 2006.

[2] N. Fuhr, M. Lalmas, S. Malik, and Z. Szlavik, editors. *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-Verlag, May 2005.

[3] G. Kazai and M. Lalmas. INEX 2005 evaluation measures. In Fuhr et al. [1], pages 16–29.

[4] G. Kazai, S. Masood, and M. Lalmas. A study of the assessment of relevance for the INEX 2002 test collection. In *Proceedings of the 26th European Conference on IR Research (ECIR)*, pages 296–310, Sunderland, UK, 2004.

[5] M. Lalmas and B. Piwowarski. INEX 2005 relevance assessment guide. In *INEX 2005 Workshop Pre-Proceedings, Dagstuhl, Germany, November 28–30, 2005*, pages 391–400, 2005.

[6] B. Larsen, S. Malik, and A. Tombros. The Interactive track at INEX 2005. In Fuhr et al. [1], pages 398–410.

[7] B. Larsen, A. Tombros, and S. Malik. Obtrusiveness and relevance assessment in interactive XML IR experiments. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, pages 39–42, Glasgow, UK, 2005.

[8] S. Mizzaro. Relevance: The whole history. *Journal of the American Society for Information Science and Technology*, 48(9):810–832, 1997.

[9] R. A. O'Keefe. If INEX is the answer, what is the question? In Fuhr et al. [2], pages 54–59.

[10] J. Pehcevski and J. A. Thom. HiXEval: Highlighting XML retrieval evaluation. In Fuhr et al. [1], pages 43–57.

[11] J. Pehcevski, J. A. Thom, and A.-M. Vercoustre. Users and assessors in the context of INEX: Are relevance dimensions relevant? In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, pages 47–62, Glasgow, UK, 30 July 2005.

[12] N. Pharo and R. Nordlie. Context matters – an analysis of assessments of XML documents. In *Proceedings of 5th International Conference on Conceptions of Library and Information Sciences, CoLIS 2005*, pages 238–248, Glasgow, UK, 2005.

[13] B. Piwowarski and M. Lalmas. Providing consistent and exhaustive relevance assessments for XML retrieval evaluation. In *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management (CIKM '04)*, pages 361–370, Washington DC, USA, 2004.

[14] T. Saracevic. Relevance reconsidered. In *Proceedings of 2nd International Conference on Conceptions of Library and Information Sciences, CoLIS 1996*, pages 201–218, Copenhagen, Denmark, 1996.

[15] A. Tombros, B. Larsen, and S. Malik. The Interactive track at INEX 2004. In Fuhr et al. [2], pages 410–423.

[16] A. Trotman. Wanted: Element retrieval users. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, pages 63–69, Glasgow, UK, 2005.

[17] R. van Zwol, G. Kazai, and M. Lalmas. INEX 2005 multimedia track. In Fuhr et al. [1], pages 497–510.

[18] E. M. Voorhees. Evaluation by highly relevant documents. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 74–82, New Orleans, USA, 2001.

Passage Retrieval and other XML-Retrieval Tasks

Andrew Trotman
Department of Computer Science
University of Otago
Dunedin, New Zealand
andrew@cs.otago.ac.nz

Shlomo Geva
Faculty of Information Technology
Queensland University of Technology
Brisbane, Australia
s.geva@qut.edu.au

ABSTRACT

At INEX there is an underlying assumption that XML-retrieval and element retrieval are one and the same. This is, in fact, not the case. The hypothesis at INEX is that XML markup is useful for information retrieval. We firmly believe this, but no longer in element retrieval. In this contribution we examine in detail the evidence collected in support of element retrieval and suggest that, contrary to expectation, it in fact supports passage retrieval and not element retrieval. Particularly, we draw on other studies that collectively show that INEX assessors are identifying relevant passages (not elements), they agree on where in a document those passages lie, that there already exists suitable metrics in the XML-retrieval community for evaluating passage retrieval algorithms, and that the tasks make more sense as passage retrieval tasks. Finally we show that future tasks of XML-retrieval also fit well with passage retrieval.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Retrieval models, Search process*.

General Terms

Human Factors, Theory

Keywords

Element retrieval, XML-retrieval, passage retrieval

1. INTRODUCTION

The IEEE document collection used at INEX [5] between 2002 and 2005 has been replaced in 2006 by the Wikipedia collection. On initial inspection, structurally this new collection does not appear to be as versatile as the previous, the DTD does not appear to be semantically as rich, and the applicability of the content itself to element retrieval does not appear to be strong.

These "weaknesses" are only of concern if the underlying assumption is that element retrieval is the most appropriate way to search the collection – and this does not appear to be the case.

In this investigation we examine the methodological evidence for passage retrieval as a replacement for element retrieval in XML-retrieval. What we find is that assessors are highlighting passages; these highlighted passages are not typically elements; and that methodology is already in place for measuring the performance of passage retrieval within INEX.

After presenting the evidence for passage retrieval, we show that some of the problems facing element retrieval do not exist if passages are used. The problems associated with identifying focused results are problems of elements, and not problems of XML-retrieval. The problem of "too small" elements does not exist if the natural relevant unit is a passage and not an element.

Information retrieval is user-centered task; the purpose is to identify relevant information and to present it to a user. We show that, in fact, some of the current element retrieval tasks are a consequence of elements and not users – specifically we ask: what are the natural tasks for a passage-retrieval system? We show that *focused* retrieval and *thorough* retrieval are equivalent under passage retrieval.

Finally we examine some possible future tasks for XML-retrieval and show that passages are the natural unit in which to specify them.

We do not suggest the XML markup is of no benefit – such markup might be used for identifying good passages. Elements might also be good answers to question answering topics.

In conclusion we propose parallel element retrieval and passage retrieval tasks at INEX 2007 with the possibility of passage only tasks at INEX 2008 and onwards.

2. Element Retrieval and Passage Retrieval

In this section we examine element retrieval and passage retrieval, then put the case that evidence collected to support element retrieval in fact supports passage retrieval.

2.1 Element Retrieval

If a document is marked up in a semantic mark-up language such as XML, it is possible for a search engine to take advantage of the structure. It could, for example, return a more focused result than a whole document. In element retrieval the search engine is tasked to identify not only which documents are relevant, but also which semantic structures (or elements) within those documents are relevant to an information need.

On initial inspection element retrieval appears to be a reasonable technology. Considering the INEX IEEE document collection, instead of returning a whole (say 10-page) document, the search engine might return a document section, subsection, or just a paragraph to the user. This far more focused result is clearly of benefit to our user. Several algorithms have been proposed and tested within [12; 29] (and without [6]) INEX

The benefit to the user is obvious. Whereas a document-centric search engine would return 10 pages, the element-centric search engine returns, perhaps, a single relevant page filtered from, perhaps, 9 other pages of irrelevant content. This machine filtering reduces the cognitive load on the users by increasing the ratio of relevant to irrelevant content presented to them.

2.2 Passage Retrieval

An alternative (and earlier) technology exists for identifying relevant parts of documents – passage retrieval. Should a

SIGIR 2006 Workshop on XML Element Retrieval Methodology
August 10, 2006, Seattle, Washington, USA.

document be long, say 10 pages, but not contain semantic markup, then element retrieval is inappropriate. Considering the same IEEE collection, but this time as a collection of PDF formatted documents, the search engine is again tasked to identify the relevant parts of the document but has no semantic markup to use. This time it must use the document content itself and not rely on explicit markup.

Several approaches have been suggested. Harper and Lee [8], for example, suggest sliding a fixed sized window over the text and computing a window score for each and every word – resulting in a relevance profile for a document. Such approaches are generally based on one variation or another of the proximity heuristic and are hence language model free. More sophisticated approaches such as natural language processing (NLP) have also been used in passage retrieval. NLP techniques appear to be successful at question answering but not yet at *ad hoc* retrieval where other than very simple techniques have yet to succeed. In question answering a more refined context analysis approach, beyond simplistic proximity heuristics, is advantageous [1; 13].

As with element retrieval the aim of passage retrieval is to reduce the cognitive load on the user. This, again, is by filtering relevant from irrelevant content within a document. Both technologies aim to increase precision.

2.3 Element Assessments

Along with the increased understanding of element retrieval came changes to the assessment methodology. At INEX 2004 assessors were presented with documents and asked to judge (pooled) elements from those. At INEX 2005 the assessors were presented with documents and asked to first identify relevant passages, then to apply exhaustivity values to elements within those passages [19]. Critically, this change allowed the analysis of passages in relation to XML documents. The evidence is in favor of passages.

2.4 Applicability

If we assume that XML markup adequately takes care of fine grained semantics, it is then a reasonable hypothesis that element retrieval is the most appropriate technology for XML and that passage retrieval is not necessary for XML documents. This, however, does not appear to be the case.

Extensive analysis of the judgments collected at INEX 2004 was done by Trotman [27] and by Pehcevski *et al*. [17]. Trotman focused his discussion on the agreement levels between judges on 12 topics assessed by two independent judges. He presents the binary document-centric agreement level as 0.27 which is low by comparison to TREC (between 0.33 and 0.49), but in line. Exact 10 relevance-point agreement of elements was 0.16, very low. Pehcevski *et al.* examined the agreement levels between the judges and participants in interactive experiments. They show agreement only at the extreme ends of the relevance scale, that is, E3S3 and E0S0 only. This end-only agreement is also seen in the cystic fibrosis collection [22]. In an effort to increase cross judge agreement the assessment method was changed from judging elements to highlighting passages – on the hypothesis that this might reduce the cognitive load on the judge resulting in an increase in agreement levels.

There has also been extensive analysis of the INEX 2005 passage and element results.

Trotman and Lalmas [28] examine which elements were identified as relevant. They found that regardless of the query specific target element there were more relevant paragraph elements than any other element. Even when the judgments were filtered for focused retrieval (with the exception of queries targeting whole articles), paragraphs prevailed in the judgments. They suggest that this might be because the assessors are identifying relevant and consecutive passages of text, and not elements, when identifying relevant content in a document.

Piwowarski *et al*. [19] examine the average specificity of paragraph elements and report a value of 0.94. For comparison, the average specificity of a section element is 0.51. They conclude that paragraphs are, in general, either completely relevant to an information need, or not at all relevant.

Piwowarski *et al*. go on to examine the correlation between passages and elements in the judgments. They define two types of passages: elemental passages and non-elemental passages. An elemental passage is a passage that is also a whole element whereas a non-elemental passage is a subset of the content of the smallest fully encompassing element. They report that only 36% of passages are elemental (therefore 64% are not). The conclusion is that assessors are not, in general, highlighting relevant elements, but are identifying relevant passages.

Ogilvie and Lalmas [14] examine the stability of the metrics under different conditions. They conclude that the exhaustivity dimension can be dropped from the assessment procedure without unduly affecting the relative performance of search engines. They suggest assessment by specificity only, or in other words highlighting passages of text and performing element retrieval based solely on these highlighted passages (as do Pehcevski and Thom [16]).

Finally, Pehcevski and Thom [16] examined the agreement levels between judges at INEX 2005 (using highlighting). They report a non-zero document level agreement of 0.39 and an exact element agreement of 0.24. Piwowarski *et al*. measured the agreement level of whole passages and report a value of 0.23. Although only 5 topics were used in this comparison, a large improvement is seen. An improvement indicating that a passage is a more natural unit than an element.

In summary, assessors are highlighting passages of text and not elements, these passages consist mostly of whole paragraphs. The judges agree not only on which documents are relevant, but on the passages within those documents. The obvious conclusion is that passage retrieval is a more appropriate technology for the INEX IEEE document collection than element retrieval.

2.5 The Case For Passage Retrieval

The INEX focused retrieval task aims to identify document elements of just the right size, however *right size* is not a well defined concept. There is scope for disagreement between assessors, and they do disagree. Furthermore, while systems are required to return XML elements of optimal granularity, the assessors as asked perform relevant passage identification. This discrepancy means that the elements of the optimal granularity (in the judgments) must somehow be derived from the relevant passages identified by the judges.

Several ways to do this have been proposed and opinions on effectiveness differ. There was, for example, much discussion and disagreement at INEX 2005 about the automatic derivation of "too small" elements. A too small element is part of a relevant passage, while at the same time insufficient in itself at fulfilling any of the information need. Such an element might

be a citation number in flowing text – relevant in context but on its own just a number.

There are two ways such difficulties might be overcome. Either ask systems to return passages instead of elements, or ask assessors to identify focused elements and too small elements and not passages. In either case there must be a direct correspondence between the retrieval task and the assessment task. It seems that passage retrieval is the obvious option from the assessment point of view, and hence probably the more reasonable approach – particularly if it more accurately matches the user needs.

But does moving to passage retrieval mean that element retrieval is unnecessary? The hypothesis being tested at INEX is that XML markup is useful in retrieval. INEX is not an element retrieval evaluation forum; it is an XML-retrieval evaluation forum. In past workshops the hypothesis was tested by comparing results that were obtained by content only (CO) queries and content and structure (CAS) queries. For some systems the hypothesis holds and for other it does not [28], but it is still an open question whether *markup* is useful. The nature of the broad concept of *ad hoc* querying, and the semantically weak markup of the INEX IEEE collection did not allow this hypothesis to be vigorously tested. By moving to passage retrieval (and perhaps with it also moving to more focused tasks such as question answering) the usefulness of exploiting XML *markup* may come to the fore. We believe this is a compelling argument for moving to passage retrieval and to more sophisticated tasks and challenges.

Can passage retrieval be assisted by XML markup? In the context of question answering, summarization, or even known entity searching it is reasonable to believe so, especially in the case of a collection with semantically strong markup and strongly typed elements. Therefore, it is necessary not only to move to passage retrieval, but to also change the kind of tasks under study and the type of collections that we use. Some of these issues are addressed in the later part of this paper, where we discuss potential future tasks for XML-retrieval systems.

2.6 Transition

Passage retrieval and element retrieval are not mutually exclusive technologies and a transition from one to the other is possible. Specifically, the transition from elements to passages is of interest for two reasons. First, this is the transition which INEX is facing. Second, it is likely to result in an increase in precision as further irrelevant content can be removed from a user's result list (that content in an element, but at the same time not relevant to the user's information need).

2.6.1 From Elements to Passages

Given a ranked set of elements from an element retrieval search engine, it is trivially possible to convert these into a set of passages. The start and end of an element become the start and end points of a passage. Additionally, immediately adjacent passages may need to be merged into a single passage.

2.6.2 From Passages to Elements

The conversion from passages to a thorough set of elements is straightforward; all elements containing any part of a passage are relevant.

The conversion to focused elements is not trivial. A passage could start mid-way through an element, cross several element boundaries and finish midway through another element. Conversion to a single element is straightforward; the smallest element fully enclosing the passage would be selected. Unfortunately it is not clear that this element is the best focused result as such an element may not be fully specific. An alternative approach might be to identify the largest elements fully enclosed by the passage. These elements would be fully specific; however there remains the potential for some relevant content to be lost, that content jutting-in to an adjacent element.

2.6.3 Passage Specification

Several methods for specifying passages have already been proposed. Previous investigations into passage retrieval such as TREC HARD have used byte offset into document and length in characters. Such a method is not suitable for XML-retrieval as mid-way through a tag might be specified.

Clarke [3] suggests element range results at INEX and recommends an XPath syntax for doing so. We note that the INEX 2005 judgments already specify passages and suggest this convention also be used for specifying passages in runs.

2.7 Passage Assessments

The transition to assessing passages has already started, albeit not for passage retrieval purposes. At INEX 2005 the assessors first identified relevant passages, then exhaustivity values were assigned to any element intersecting the passage [19]. The extensions necessary to change to passage retrieval could be done in one of two possible ways. Either the assignment of exhaustivity would be to a passage and not an element, or alternatively the assessment of exhaustivity could be dropped. The latter has been suggested already by Ogilvie and Lalmas [14] and is already under consideration for INEX 2006. Should this be adopted then everything, except the task definitions, are in place for passage retrieval.

3. Passage Retrieval Tasks

Passage retrieval is well suited to XML documents. Additionally, passages can be more accurate as there is no requirement for a passage to start (or end) on a tag boundary. But what of the element retrieval tasks currently under investigation? It is important to look at user needs before task definition, but it turns out there are direct analogies between the existing element retrieval tasks and those one might expect for passage retrieval.

We initially envisage three tasks: the first is it the identification of relevant passages of text which are presented to the user in passage-relative order of relevance – this turns out to be a combination of the existing *focused* task and *thorough* task. The second is the identification of relevant passages of text which are presented to the user in document-relative order of relevance – essentially the *relevant in context* task. Finally, the identification of relevant documents presented to the user with and entry point identified – the *best in context* task essentially unchanged.

The retrieval task specification for INEX 2006 [4] discusses these 4 tasks with respect to element retrieval. In this section we discuss transitioning them to passage retrieval.

3.1 Focused Retrieval

In the existing *focused* task, a search engine must identify only those relevant elements that are most focused on the information need. A list of focused results may not contain any overlapping elements. For the search engine there are two problems at hand: the first is the identification of a relevant piece of text (where); and the second is the identification of the appropriate size of the text.

This task would change only subtly. Whereas at present the task is to identify non-overlapping elements (essentially passages), it would be changed to the identification of non-overlapping passages. A transitional requirement might be that passages must start and end on a tag boundary. This transition would allow the continued use of the current metrics. Alternatively, the introduction of a metric such as HiXEval [16] would alleviate this transitional need.

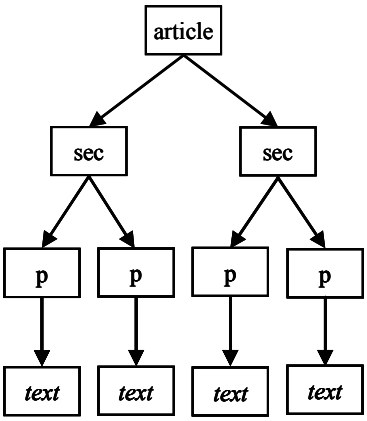


Figure 1: A simple document tree with text at the leaves

3.2 Thorough Retrieval

In the existing *thorough* task, a search engine must identify each and every relevant element in the document collection, and it must rank these relative to each other. This task is the only task that has continued in INEX since the first workshop.

This task has been criticized as it, by its very definition, requires the search engine to return overlapping elements in the results list [27]. Examining the document tree in Figure 1, and relevant text in a <p> element, and that inside a <sec> element, and that inside an <article> element. A thorough retrieving search engine will identify all three, and rank them relative to each other.

A natural consequence of this task is that the same text could be identified multiple times. To be *thorough* the search engine must identify *all* overlapping elements. In an interactive environment in which these overlapping results are displayed on-screen for a user, that user could potentially be presented with the very same element of text, and only that element of text, for the entire first page of results. Experiments conducted as part of the interactive track at INEX 2004 show that users do not want overlapping elements in results lists [11; 24]. This makes it a target for criticism on the basis of having no user-model, and it has been criticized for this [27].

We believe these criticisms are short-sighted, not because they are wrong but because the conversion to thorough results list from a passage is straightforward. This task could, therefore, act as a sanity check during the transition from elements to passages.

Of course, under the definition of passages, the thorough and focused task are equivalent[1] – the identification of documents, start points and the end points of all passages of text that satisfy the user's information need. These passages are sorted relative to each other.

[1] Until the use-case, we avoid discussing tasks with overlapping passages

3.3 Relevant In Context

In the existing *relevance in context* task, a search engine must first identify which documents are relevant, and then identify which elements within those documents are relevant. Results are grouped first by document, and then presented in document order. Overlapping elements are forbidden. This task is based on the experimental *Fetch Browse* task of INEX 2005 but the older task was thorough. This tasks is already (essentially) a passage retrieval tasks.

The change to passage retrieval is just a change in granularity. Whereas an element retrieval search engine is restricted to identifying elements, a passage retrieval search engine might identify passages that do not start or end on element boundaries (perhaps sentences).

By switching this task to a passage retrieval task, it is brought inline with the focused task. The difference between them being the order passages are returned. Relevance in context results lists would be in document order whereas focused results lists would be relative to other passages.

3.4 Best In Context

In the existing *best in context* task, a search engine must first identify relevant documents and then a single best point (BEP). The BEP is used to direct the user to relevant content within the document. At present this entry point is specified as an element start point. Only one best entry point into a document may be given and results are ranked on document topical relevance.

There may not be one best entry point in a document. Piwowarski *et al*. [19] examine the number of relevant passages per relevant document in the INEX 2005 judgments. They report that fewer than 50% of relevant documents contain only one relevant passage, while over 85% of relevant documents contain 5 or fewer relevant passages. As many as 49 passages are seen in one relevant document. When there are multiple passages in a single document it is not clear that one particular passage must necessarily be any better than all the others. This leads to questions about cross-judge agreement levels – which remain to be computed (this task is new for INEX 2006).

Conversion of this task to passage retrieval requires one subtle change; the entry point would no longer be required to lie on a tag boundary.

With passage retrieval this task is very close in definition to both focused and relevant in context. In relevant in context, documents are sorted relative to each other. Focused results are sorted relative to each other. Best in context results are first sorted on documents and then within document they are sorted relative to each other.

3.5 Passage Retrieval At TREC

The TREC HARD track [25] examined passage retrieval in 2003 and 2004. There the granularity of a query result was specified in metadata attached to the query. A query could target a document, passage, sentence or phrase sized units. Passages were specified in submissions as byte offset into a document, and length.

The TREC Genomics track is using a collection of scientific articles marked up in HTML for question answering. Results to queries are passages, identified by document identifier, passage offset, and passage length. Several TREC Genomics participants pushed for the collection in XML, including some also active in INEX.

We believe INEX should be looking at passage retrieval in semi-structured (XML) documents. TREC Genomics is already looking at passage retrieval in semi-structured (HTML) documents. This is an ideal opportunity to share results – and document collections.

By sharing document collection the algorithms from INEX and TREC Genomics could be compared head to head, this would imply also sharing metrics.

4. The Performance Task

Thorough retrieval is the only retrieval task that has been at INEX since the start. It could be used to measure the annual performance increase seen in ranking algorithms (as could other tasks, but this task has existed from the start).

A mapping from a passage to a thorough list is mechanical. All elements fully contained by the passage are fully and equally relevant. All those not intersecting with a passage are not relevant. For all others the relevance can be computed in the manner in which specificity is currently computed in the judgments: the ratio of thought-relevant text to the size of the element. A relevance value for elements in all documents can be computed and these ranked relative to each other.

With thorough rankings for search engines from the start of INEX, and a single (appropriately chosen) metric, the performance of the best submitted runs can be computed for each year and the result graphed since the beginning of INEX. Care must be taken when interpreting such a result as differences could reflect the hardness of the topic set and not improvements in search engine performance.

Alternatively, a set of unchanging benchmark topics could be used. These topics would remain the same from year to year and would not form part of evaluation – only new topics would be used for that. However, by analyzing the global performance on benchmark topics we would be able to say with confidence whether, or not, performance across the board was improving. There is still the risk that over-fitting will occur if INEX participants use these benchmark topics to train their systems – as they will no-doubt attempt to do. This might be overcome if neither the topics nor the judgments were released. Only performance statistics would be given.

Introduction of the Wikipedia collection is opportune. 125 topics have already been published for INEX 2006. From those, some suitably large number (say 25), might be used as benchmark topics and the other (say 100) for standard evaluation purposes. The judgments for the benchmarks would be withheld whereas the other judgments would be published.

Informal discussions, currently centered on an efficiency track, have suggested participants should submit their search engines and not runs. Should INEX adopt such an approach then performance changes from year to year could be measured on these submitted search engines. Care must also be taken with this approach as each year some participants re-train their search engines using the results from previous years. Re-running queries on these re-trained search engines is equivalent to measuring the performance of the training set – which should be optimal.

None the less, with INEX in its 5th year it is still not clear that any one relevance ranking algorithm is superior to any other. There are no standard benchmarks to which new algorithms are compared, and no clear evidence that improvements are being made from year to year. The purpose of this track would be to identify the state of the art and to introduce a standard methodology for experimentation.

In whole document retrieval the performance of a new ranking algorithm is compared to that of BM25 [20], pivoted length normalized retrieval [23], or language models [33]. Any differences are checked for statistical significance using either the *t*-test or Wilcoxon test [21]. No such standard methodology exists for XML-retrieval – because it is not clear which algorithms are state of the art.

Part of the cause of this problem has been the shifting metrics. An effective metric should be both stable, and say something useful. For XML-retrieval, something useful has been the cause of much debate. Generalized Precision Recall (inex_2002) [9] was criticized because it rewarded search engines for returning overlapping elements [10] – something shown to be a cause of frustration to users in the interactive experiments [11; 24].

The first alternative, NG [7] was criticized because it treated precision and recall separately and did not combine them into a single metric [32]. Because it assumed relevant content was uniformly distributed in an element, and because it did not address the overpopulated recall base problem [32].

There was very much a need for an appropriate metric when XCG [10] was introduced. Variants of this metric were used at INEX 2005, however there was debate. Woodley and Geva [32] showed that this metric is overlap negative that is, runs including overlapping elements were penalized. Piwowarski and Dupret [18] criticized it for having no user model.

Further metrics have been proposed: PRUM and EPRUM [18] model the behavior of a user in a hypertext environment. Such a user might click on a result in a results list, and then navigate from there to a relevant document through a hypertext link. This metric stochastically models this behavior. The versatility of this metric makes it appropriate for XML-retrieval – however we await the investigation into the behavioral parameters needed before it could be applied without controversy.

If passage retrieval is to take the place of element retrieval then metrics specifically designed for measuring passage-based performance are needed.

Two such metrics have been proposed for the TREC HARD track [25]. The first is the R-Precision of the F measure of individual passage precision and recall scores (passage precision and recall were computed on a character by character basis). This measure was shown to prefer a large number of short and contiguous passages over a small number of non-contiguous passages, that is, it encouraged identifying passages and then splitting them. The second was the bpref [2] of the top 12,000 characters.

TREC 2006 Genomics track [26] is proposing to use mean average passage precision (MAPP) where passage precision is computed as character overlap with relevant passages.

For XML-retrieval, Pehcevski and Thom suggest HiXEval [16], the F measure of the passage precision and passage recall, where passage precision and passage recall are defined with a tuning parameter to compensate for overlapping passages.

In summary, elements can be converted into passages. The performance of each of the runs thus-far submitted to INEX could be computed using a metric such as HiXEval, and the top performing algorithms identified. The performance of these could be graphed identifying if, or not, progress is being made

at XML-retrieval. A standard methodology could be put in place by which new algorithms are compared to old and statistical tests could be used to show the significance of any reported improvements.

5. Multiple Document Formats

XML is one of many semi-structured formats; SGML and HTML are two others. Or a document might be stored in plain unstructured text. The premise of XML-retrieval is that the structure, necessarily present in an XML document, can be used to improve performance. It might be used by a user to state, more specifically, where in a document relevant content might be found (a CAS query). Or it might be used by a search engine to increase the precision by returning only relevant elements (in a CO query). But does this structure help?

Trotman and Lalmas [28] compare the performance of a set of content only (CO) queries to their counterpart with structure added (CO+S queries). They found no statistical difference in performance of the best runs (submitted to INEX 2005) for the two types of queries on the same document collection.

The document collection they used was highly marked up. For both kinds of query (CO and CO+S) the search engines were able to, and did, take advantage of the structure. It is not at all obvious that the result would be the same if the same queries were run on documents not so strongly marked up. For the collection they used (INEX IEEE), such a derivative collection could be constructed by removing XML tags from each document leaving just the pain text. For the INEX Wikipedia collection, HTML, XML and plain text versions could be made available.

It is reasonable to assume that a search engine working without structured documents would not perform as well as one working with structured documents – but there are reasons to believe it might. Without structure the search engine is forced to identify relevant passages; and passages are more likely to be a better fit to the user's information need than are elements. It is reasonable to assume the precision might increase as a result. On the other hand, the element boundaries might help with the identification of passages so precision on the XML collection might be better.

Either way, it is reasonable to assume some queries will be better serviced by XML documents, some by HTML, and others by plain text. Knowing which will help identify the circumstances under which markup is of benefit, of how much benefit, and how much markup is needed for that benefit.

Opening up XML-retrieval to include HTML, plain text, and passage techniques will bring with it techniques from other information retrieval domains. This will provide an opportunity for understanding semi-structured document retrieval without being tied to XML.

6. Related Articles (Mini-Web)

Web retrieval differs considerably from other forms of information retrieval. The web is a dynamic hyperlinked environment where all pages are current and two pages can link to each other. In an academic document collection (such as the INEX IEEE collection) links can only point backwards in time – an academic article cannot be changed (after it has appeared in print) to cite papers published *post facto*.

Wikipedia articles are more like the web than like academic articles (the IEEE collection) in this regard. All articles are current and articles can cite each other, thus forming a mini-web. This leads to two problems: First the maintenance problem of keeping all cross links up-to-date. Second the selection of the mini-web when a new article is added.

In a dynamic environment new articles are constantly being added and old articles deleted, in both cases links must be maintained. Examining article 5001 on "Bathyscaphe Trieste", there is a section entitled "See also" that contains links to three other articles in the collection as well as one yet to be written. But there is no "See also" link to the vehicle's successor, the "Bathyscaphe Trieste II". The person who created (or maintains) the article also had to make the connection – this is tedious and requires extensive knowledge they may not have. Incoming links should also have been added to the collection, but from where? This task is even more tedious, perhaps prohibitively so as it requires updating many documents. The added value of a related articles task is clear.

An automated system would take a written article, find others like it (using XML-retrieval techniques) and suggest a mini-web of bidirectional links that a user may then (fetch) browse, filter, clean, and adopt as a desirable set of mini-web links. This process would both significantly enhance the collection and facilitate an activity that is highly unlikely to occur otherwise.

Creating cross-document links is a document similarity problem. This has already been examined in many domains (such as medicine [31]). But the Wikipedia offers a unique opportunity to examine document similarity in XML-retrieval. This is for one important reason – human generated links between documents are already in the collection. An almost cost-free evaluation method presents itself.

We expect a good concept formation system to return a set of links that is at least partially overlaps those that are already defined by the original contributors to the article.

The links between articles in the collection could be removed. Several articles from the collection could be selected as a test set, and a search engine would be tasked to insert links to relevant articles from the collection. The submitted runs would be compared to the ground truth – the links that were removed from the article in the first place.

If resources are available manual relevance judgments may be performed on those links identified by a search engine, but not already known to be appropriate. This would not be too onerous as it is a simple yes / no question – either the articles are related or they are not.

The performance of a search engine could also be computed in a straightforward manner. The precision with respect to a single article could be measured with mean average precision, and the mean of this might be used over a collection of query articles.

A clear task with a real need has presented itself. Topics already exist and evaluation is inexpensive. Best of all, the task only makes sense in a semi-structured hyperlinked environment – it is an ideal XML-retrieval task.

The task has an analogue for passages of text. In this case the need is not for "See also" links, but for links from the paragraph text to other articles. In this case a test set might be created by removing the links from pre-existing paragraphs. Natural language processing techniques might be used by a search engine to re-insert them. This task might be treated as a known entity searching problem and performance might be measured using mean reciprocal rank (MRR).

7. Question Answering

O'Keefe [15] examined the queries submitted to INEX 2003 and noted the high proportions that did not target elements as

return results. Trotman and Lalmas [28] identify only 13 (68%) of the 19 assessed CAS topics at INEX 2005 targeting elements. In the words of O'Keefe "If INEX is the answer, what is the question?".

Piwowarski *et al*. [19] observed that paragraphs are almost exclusively either fully specific or not specific to an information need. By comparison, only half of a relevant section element was specific on average. It is reasonable to conclude from their investigation that if elements are the right granularity of answer then the queries should be targeting paragraphs, or perhaps paragraphs and elements smaller than paragraphs: sentences, phrases, or single words.

Queries targeting words, sentences and paragraphs are not the usual domain of the *ad hoc* query. They usually target whole documents (or, of course, passages from documents). Words, sentences, or sometimes paragraphs are the granularity of answer expected of a question answering system.

INEX does not, at present, have a question answering track, but it is an obvious extension to both the NLP track and the Entity Ranking track. Questions would be asked in natural language and information (entity) extraction techniques would be used to identify answers. Standard methods such as those used at TREC Question Answering [30] would be used to evaluate performance.

It is reasonable to believe the markup present in an XML document will be of help in this task. The templates present in the INEX Wikipedia collection are of particular interest. One might ask "When was Edmund Burke first made Paymaster of the Forces?" to which the answer (1782) is held in a single template tag of the document on Edmund Burke (document 10030).

8. Conclusions

In this contribution we have examined evidence collected (by others) in favor of element retrieval with XML documents and shown that, in fact, it supports passage retrieval.

Prior studies into the agreement levels between judges show that that when judges are asked to identify relevant passages, and not elements, that the agreement level is very much higher than when asked to identify relevant elements.

Studies into which elements are most likely to be relevant show that paragraphs are essentially an atomic unit of relevance. Studies correlating passages and elements show that relevant passages in the text are not usually elements, but rather collections of consecutive elements (or, indeed, passages).

We discussed some of the problems facing element retrieval. Specifically we note that the problem of automatically identifying "too small" elements does not exist with passage retrieval. The problem of deriving the "ideal recall base" for focused retrieval disappears. We also drew evidence from a study that showed that the two dimensional relevance, itself problematic, is unnecessary if assessors judge passages and not elements. Methods for search engine evaluation, we believe, are already in place if metrics like HiXEval are used.

We examined possible user tasks for passage retrieval and showed that the existing XML-retrieval tasks *focused* and *thorough* are analogous under passage retrieval. We examined the *relevant in context* task, and the *best in context* track and showed they not only do they exist essentially unchanged with passages, but that the differences between all these tasks is easily explained.

Finally we examined possible future XML-retrieval tasks and showed that a paradigm shift to passage retrieval not only has no negative impact on these tasks, but is likely to enhance them.

The future of XML-retrieval is, we believe, with passage retrieval and not element retrieval. We showed that the transition from element to passages can be smooth, and that methods are already in place to make the transition. We now propose that INEX 2007 fully embrace passage retrieval and run parallel passage and element tasks with the intent of moving solely to passages for 2008.

9. References

[1] Bernardi, R., Jijkoun, V., Mishne, G., & de Rijke, M. (2003). Selectively using linguistic resources throughout the question answering pipeline. In *Proceedings of the 2nd CoLogNET-ElsNET Symposium*.

[2] Buckley, C., & Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. In *Proceedings of the 27th ACM SIGIR Conference on Information Retrieval*, (pp. 25-32).

[3] Clarke, C. (2005). Range results in XML retrieval. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology, Second Edition*, (pp. 4-5).

[4] Clarke, C., Kamps, J., & Lalmas, M. (2006, to appear). INEX 2006 retrieval task and result submission specification. In *Proceedings of the INEX 2006 Workshop*.

[5] Fuhr, N., Gövert, N., Kazai, G., & Lalmas, M. (2002). INEX: Initiative for the evaluation of XML retrieval. In *Proceedings of the ACM SIGIR 2002 Workshop on XML and Information Retrieval*.

[6] Fuhr, N., & Großjohann, K. (2000). XIRQL an extension of XQL for information retrieval. In *Proceedings of the ACM SIGIR 2000 Workshop on XML and Information Retrieval*.

[7] Gövert, N., Kazai, G., Fuhr, N., & Lalmas, M. (2003). *Evaluating the effectiveness of content-oriented XML retrieval*: University of Dortmund, Computer Science 6.

[8] Harper, D. J., & Lee, D. (2004). On the effectiveness of relevance profiling. In *Proceedings of the 9th Australasian Document Com-puting Symposium*, (pp. 10-16).

[9] Kazai, G. (2003). Report of the INEX 2003 metrics working group. In *Proceedings of the INEX 2003 Workshop*.

[10] Kazai, G., Lalmas, M., & de Vries, A. P. (2004). The overlap problem in content-oriented XML retrieval evaluation. In *Proceedings of the 27th ACM SIGIR Conference on Information Retrieval*, (pp. 72-79).

[11] Kim, H., & Son, H. (2004). Interactive searching behavior with structured XML documents. In *Proceedings of the INEX 2004 Workshop*, (pp. 424-436).

[12] Mass, Y., & Mandelbrod, M. (2004). Component ranking and automatic query refinement for XML retrieval. In *Proceedings of the INEX 2004 Workshop*, (pp. 73-84).

[13] Moldovan, D., Harabagiu, S., Girju, R., Morarescu, P., Lacatusu, F., Novischi, A., Badulescu, A., & Bolohan., O. (2002). LCC tools for question answering. In *Proceedings of the 12th Text REtrieval Conference (TREC-11)*.

[14] Ogilvie, P., & Lalmas, M. (2006 (submitted)). Investigating the exhaustivity dimension in contentoriented XML element retrieval evaluation.

[15] O'Keefe, R. A. (2004). If INEX is the answer, what is the question? In *Proceedings of the INEX 2004 Workshop*, (pp. 54-59).
[16] Pehcevski, J., & Thom, J. A. (2005). HiXEval: Highlighting XML retrieval evaluation. In *Proceedings of the INEX 2005 Workshop*.
[17] Pehcevski, J., Thom, J. A., & Vercoustre, A.-M. (2005). Users and assessors in the context of INEX: Are relevance dimensions relevant? In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology, Second Edition*, (pp. 47-62).
[18] Piwowarski, B., & Dupret, G. (2006). Evaluation in (XML) information retrieval: Expected precision recall with user modelling (EPRUM). In *Proceedings of the 29th ACM SIGIR Conference on Information Retrieval*.
[19] Piwowarski, B., Trotman, A., & Lalmas, M. (2006 (submitted)). Sound and complete relevance assessments for XML retrieval.
[20] Robertson, S. E., Walker, S., Beaulieu, M. M., Gatford, M., & Payne, A. (1995). Okapi at TREC-4. In *Proceedings of the 4th Text REtrieval Conference (TREC-4)*, (pp. 73-96).
[21] Sanderson, M., & Zobel, J. (2005). Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proceedings of the 28th ACM SIGIR Conference on Information Retrieval*, (pp. 162-169).
[22] Shaw, W. M., Wood, J. B., Wood, R. E., & Tibbo, H. R. (1991). The cystic fibrosis database: Content and research opportunities. *Library and Information Science Research,* 13, 347-366.
[23] Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. In *Proceedings of the 19th ACM SIGIR Conference on Information Retrieval*, (pp. 21-29).
[24] Tombros, A., Larsen, B., & Malik, S. (2004). The interactive track at INEX 2004. In *Proceedings of the INEX 2004 Workshop*, (pp. 410-423).
[25] TREC. (2003). Hard, high accuracy retrieval from documents TREC 2003 track guidelines. TREC. Available: http://ciir.cs.umass.edu/research/hard/guidelines2004.html [2006, 16 June].
[26] TREC. (2006). TREC 2006 genomics track draft protocol. TREC. Available: http://ir.ohsu.edu/genomics/2006protocol.html [2006, 16 June].
[27] Trotman, A. (2005). Wanted: Element retrieval users. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology, Second Edition*, (pp. 63-69).
[28] Trotman, A., & Lalmas, M. (2006). Why structural hints in queries do not help XML retrieval. In *Proceedings of the 29th ACM SIGIR Conference on Information Retrieval*.
[29] Trotman, A., & O'Keefe, R. A. (2003). Identifying and ranking relevant document elements. In *Proceedings of the 2nd workshop of the initiative for the evaluation of XML retrieval (INEX)*.
[30] Voorhees, E. M., & Tice, D. M. (2000). Building a question answering test collection. In *Proceedings of the 23rd ACM SIGIR Conference on Information Retrieval*, (pp. 200-207).
[31] Wilbur, W. J., & Coffee, L. (1994). The effectiveness of document neighboring in search enhancement. *Information Processing & Management,* 30(2), 253-266.
[32] Woodley, A., & Geva, S. (2005). XCG overlap at INEX 2004. In *Proceedings of the INEX 2005 Workshop*, (pp. 25-39, pre-proceedings).
[33] Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *Transactions on Information Systems,* 22(2), 179-214.

Author Index

Ashoori, Elham 20

Dopichaj, Philipp 1

Geva, Shlomo 5,43

Hassler, Marcus 5

Kamps, Jaap 13

Kazai, Gabriella 20

Larsen, Birger 13

Lehtonen, Miro 28

Pehcevski, Jovan 35

Tannier, Xavier 5

Trotman, Andrew 43