Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology.

Held at the University of Glasgow,

30 July 2005.

Second Edition

Edited by
Andrew Trotman,
Mounia Lalmas
Norbert Fuhr

Proceedings of the
INEX 2005 Workshop on Element Retrieval Methodology,
held at the University of Glasgow,
30 July 2005.

Second Edition

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

Editors:
Andrew Trotman,
Mounia Lalmas,
Norbert Fuhr

ISBN 0-473-10228-5

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

UNIVERSITY *of* GLASGOW

Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology, held at the University of Glasgow, 30 July 2005. Second Edition.

Preface

These proceedings contain the papers of the INEX 2005 Workshop on Element Retrieval Methodology held at the University of Glasgow on 30th July 2005. Ten papers were selected by the program committee from eleven submissions. Each paper was reviewed by at least two members of the committee – however reviewing was not comprehensive due to the nature of the papers and workshop (they are opinion papers).

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 University of Glasgow 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.

INEX is an activity of the DELOS network of excellence in digital libraries.

Andrew Trotman
Mounia Lalmas
Norbert Fuhr

Organizers

Norbert Fuhr	University of Duisburg-Essen (Germany)
Mounia Lalmas	Queen Mary University of London (UK)
Andrew Trotman	University of Otago (New Zealand)

Program Committee

Norbert Fuhr	University of Duisburg-Essen (Germany)
Shlomo Geva	Queensland University of Technology (Australia)
Mounia Lalmas	Queen Mary University of London (UK)
Birger Larsen	Royal School of Library and Information Science (Denmark)
Yosi Mass	IBM Research Lab in Haifa (Israel)
Benjamin Piwowarski	University of Chile (Chile)
Andrew Trotman	University of Otago (New Zealand)
Arjen de Vries	Centrum voor Wiskunde en Informatica (The Netherlands)

Contents

Introduction to the INEX 2005 Workshop on Element Retrieval Methodology

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

Mounia Lalmas
Department of Computer Science
Queen Mary University of London
London, UK
mounia@dcs.qmul.ac.uk

1. INTRODUCTION

With a wealth of documents originating in markup languages such as XML, it is appropriate to ask how this markup might be used in information retrieval. One answer is to change the focus of retrieval from whole documents to document elements.

In document-centric IR the user searches whole documents and is returned a ranked list of documents that match their queries. By contrast, in element retrieval document elements are returned – perhaps a chapter of a book, or a section of an academic paper.

Since 2002 the annual INEX workshop [2] has been examining element ranking algorithms for XML documents. Most specifically, the IEEE collection of 12,107 documents. Arguably progress has been made.

It is this "arguably" that has become the center of attention. On the outset it would appear as though element retrieval is a simple derivation of document retrieval – but experience at INEX has shown this to be far from the truth.

A document centric search engine makes a binary decision about the relevance of a given document – either it will appear in a result list or it will not. It cannot "partly appear".

An element centric search engine having decided a piece of text is relevant is faced with how to return that information. Perhaps only a paragraph is relevant, or perhaps the sub-section, or the section, or it may be the entire document. The same piece of text can be returned in many different ways.

When humans are making judgment decisions, they too, are faced with similar problems. If a given paragraph is relevant, then surely a containing section is also relevant. How much more so, or less so?

Combining these, how can the performance of a search engine be measured?

There are clearly methodological issues in element retrieval, and these need addressing. It is these issues that are of interest at this workshop.

For many the most pressing issues is this: when there is no community accepted methodology it is not possible to claim any one system is better than any other.

2. FOCUS OF THE WORKSHOP

The workshop was organized to address some of the methodological issues in element retrieval. Specifically six areas requiring attention were identified: theory, application, measurement, judgment, experience, and other. Two areas were excluded: ranking algorithms and existing software.

2.1 Theory

A sound theoretical basis for element retrieval is yet to be established, both in terms of the document collection, and the interaction model.

It is not clear what properties of an XML document collection make it more suitable for element retrieval than for document retrieval. It is also not clear what properties make that collection either "heterogeneous" or "multimedia".

As yet there is no established theoretic basis of interaction with element retrieval – although the INEX interactive track is investigating this [14]. It is not clear when an element is a better answer than a document, or if elements must be bound by context when returned to the user.

2.2 Application

It is entirely possible that many of the methodological issues can be resolved if there existed an application of element retrieval (outside the research community). It is not clear where to look for such an application, or if such an application will ever exist.

2.3 Measurement

One of the methodological issues addressed from the outset is that of performance measures. Kazai [7] identifies five different metrics that have been proposed, and there are more besides. It is clear that these metrics measure different things, however what is not clear is what should be measured – or how to measure it. With no single community accepted performance metric, it is impossible to identify one algorithm as any better than any other. Consequently, progress on ranking algorithms is impossible to make.

2.4 Judgment

At present INEX judgments are made on two separate dimensions, one is a measure of how specific the element is, and the other how exhaustive the element is. Each is on a four point scale (not, marginally, fairly, highly), giving a total of ten grades (if an element is not specific it cannot be exhaustive, and *vice versa*).

Prior investigations into the judgments (such as that of Pehcevski [11]) have raised questions as to whether or not the assessors understand this scale – and it is not clear. It is entirely possible that the complexities of grading a "near miss" element (that encloses relevant information but is not itself entirely relevant) are beyond the capabilities of a subjective assessor.

What is clear is that different assessors have different marking conventions. Some will mark references as valid, where others

may not. Under investigation is the judgment process and the judgments. Are they, or are they not sound?

2.5 Experience

Drawing on the experience of other evaluation workshops (including TREC [3], NTCIR [6], and CLEF [13]) may provide answers to some of the methodological issues facing element retrieval. Parallels, for example, can be drawn between element retrieval and passage retrieval.

2.6 Other

By including an "other" category the workshop remained open to discussion of any additional issued not discussed above.

2.7 Exclusions

Ranking algorithms were specifically excluded from focus primarily to ensure the workshop would not act as a "half-INEX". That is., by allowing submissions on the topic of relevance ranking there was a perceived danger that the workshop would turn into an evaluation forum. The end of year INEX workshop fulfills this purpose admirably so accepting contributions on this topic would only blur the boundaries between the two workshops.

The existing software was excluded for two reasons. First, the mammoth efforts of those who build it should not go unnoticed, and attracting criticism of this effort was perceived as departmental to both the individuals involved and to the community as a whole. Second, the software should not dictate methodology, but should reflect methodology – as such the focus of the workshop was shifted from what the community currently does to what it should do.

3. CONTRIBUTIONS

A general call for papers was widely distributed. Interested parties were asked to contribute opinion papers for the purpose of promoting discussion. A total of eleven contributions were received, of which ten were accepted. Originally only four were to be accepted; however the papers were unexpectedly broad and workshop was reorganized to accommodate this.

3.1 Short Review of Submissions

Clarke [1] attacks individual elements as a suitable search engine result. He provides evidence that relevant information lies in sequences of tags (e.g. two consecutive paragraphs) and identifies a mismatch between returned results and relevant information. He suggests results should be returned as element ranges and provides a syntax for doing so. He suggests judgments should be done in the same manner and proposes using text-highlighting as a method of achieving this.

Hiemstra and Mihajlovic [4], apply the "simplest possible" approach to evaluation metrics and argue that precision-at-n elements reported with overlap scores provides a wealth of information for comparing two systems. They provide scores for several runs from INEX 2004 and explain how to read their scores and what, exactly, the scores mean.

Kamps *et al*. [5] examine what can (in principle) be expressed in a query language, then examine how users actually use such languages. From this they suggest formulating a set of topics with CO, CAS, and NLP expressions of the same information need (i.e. sharing a narrative). Judging against the narrative makes it possible to compare the performance of each of the queries and to directly compare each type of query. This will provide evidence of the superiority (or not) of using structural hints in a query.

Kazai and Lalmas [8] examine the requirements for an element retrieval precision metric. They classify each of the existing metrics against a list showing that they all fall short on some account.

Larsen *et al*. [9] identify the obtrusiveness of the relevance scale in user interaction experiments. By removing this imposition a true investigation of the element-centric searching behavior of users could be conducted. They provide several suggestions of non-obtrusive ways to examine user interaction.

Larson [10] focuses on heterogeneous searching. Identifying with the user, he notes that as the number of document collections increases, the cognitive load of the user increases. Whereas a user might have the ability to intimately know one DTD, there is little chance they will intimately know hundreds of DTDs. He identifies content and structural heterogeneous search as a possibly impossible user task. He suggests the issues might be addressed with reference to prior work in IR including embracing the principles of the Dublin Core.

Pehcevski *et al*. [12] examine the different judging behaviors between topic assessors and users (from the INEX interactive track). They identify patterns in judging behavior which demonstrate that the 10 point relevance scale is not well understood. They recommend changing the judgment scale.

Trotman [15] claims that the methodological issues in element retrieval stem from a lack of user grounding. If an application existed it could be examined and issues resolved with respect to the application. Identifying the IEEE collection as not suitable for element retrieval he calls for a shift to an audio or video collection, and metrics that do not reward element milking.

Woodley and Geva [18], frustrated at the judgment process, investigate ways to generate a more reliable set of judgments, while at the same requiring less work on the part of the judge. They provide evidence that to remain stable the judgment pool must be made from all retrieval runs and that the judgments must continue to be graded. However, they also identify out-of-pool judgments (those not in the pool, but forced by element context) as unnecessary. Secondly, they discuss ways to annotate the document collection. Finally they propose several possible future tracks.

Van Zwol *et al*. [17] suggest the complex structures of NEXI [16] are beyond the abilities of end users. They propose a visual query language called Bricks. This method of searching, they suggest, is more successful at completing the end user task than keyword search, while being faster (for the same purpose) then NEXI.

4. CONCLUSIONS

The INEX 2005 Workshop on Element Retrieval Methodology aims to provide a forum for discussion of element retrieval issues (other than relevance ranking).

Collected in this volume are papers on a broad set of issues ranging from user interaction through to performance metrics. These opinion papers were solicited with the aim of promoting discussion, and they no doubt will. The collection forms a discussion document for the workshop.

It is the combination of the discussion document and the face to face debate at the workshop that will enable progress on the many raised issues. When reading these papers, remember the object was to raise issues for discussion, not to solve the problems.

5. ACKNOWLEDGEMENTS

The organizers would like to thank the University of Glasgow for hosing the workshop. INEX is an activity of the DELOS network of excellence in digital libraries.

Without the discussions of the element retrieval community, including INEX and the various discussion lists, element retrieval would never have developed to where it is – it is the work of these others that makes the work of us possible.

It is always necessary to thank the program committee, the paper authors, and the participants, for without these people there would be no workshop.

6. REFERENCES

[1] Clarke, C. (2005). Range results in XML retrieval. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*.

[2] Fuhr, N., Gövert, N., Kazai, G., & Lalmas, M. (2002). INEX: Initiative for the evaluation of XML retrieval. In *Proceedings of the ACM SIGIR 2000 Workshop on XML and Information Retrieval*.

[3] Harman, D. (1993). Overview of the first TREC conference. In *Proceedings of the 16th ACM SIGIR Conference on Information Retrieval*, (pp. 36-47).

[4] Hiemstra, D., & Mihajlovic, V. (2005). The simplest evaluation measures for XML information retrieval that could possibly work. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*.

[5] Kamps, J., Marx, M., Rijke, M. d., & Sigurbjörnsson, B. (2005). Understanding content-and-structure. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*.

[6] Kando, N. (2001). Overview of the second NTCIR workshop. In *Proceedings of the 2nd NTCIR Workshop*.

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

[8] Kazai, G., & Lalmas, M. (2005). Notes on what to measure in INEX. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*.

[9] Larsen, B., Tombros, A., & Malik, S. (2005). Obtrusiveness and relevance assessment in interactive XML IR experiments. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*.

[10] Larson, R. (2005). XML element retrieval and heterogeneous retrieval: In pursuit of the impossible? In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*.

[11] Pehcevski, J., Thom, J. A., Tahaghoghi, S. M. M., & Vercoustre, A.-M. (2004). Hybrid XML retrieval revisited. In *Proceedings of the INEX 2004 Workshop*, (pp. 153-167).

[12] 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*.

[13] Peters, C. (2004). What happened in CLEF 2004? Introduction to the working notes. In *Proceedings of the CLEF 2004*.

[14] Tombros, A., Larsen, B., & Malik, S. (2004). The interactive track at INEX 2004. In *Proceedings of the INEX 2004 Workshop*, (pp. 410-423).

[15] Trotman, A. (2005). Wanted: Element retrieval users. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*.

[16] Trotman, A., & Sigurbjörnsson, B. (2004). Narrowed Extended XPath I (NEXI). In *Proceedings of the INEX 2004 Workshop*, (pp. 16-40).

[17] van Zwol, R., Baas, J., van Oostendorp, H., & Wiering, F. (2005). Query formulation for XML retrieval with bricks. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*.

[18] Woodley, A., & Geva, S. (2005). Fine tuning INEX. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*.

Range Results in XML Retrieval

Charles L. A. Clarke
School of Computer Science, University of Waterloo, Canada
claclark@plg.uwaterloo.ca

1. INTRODUCTION

To date, retrieval results for the INEX adhoc task have been restricted to simple XPath location paths with positional predicates, such as

```
/article[1]/bdy[1]/sec[5]/ss1[1]/p[4]
```

which specifies the fourth paragraph in the first subsection of the fifth section of the first body of the first article. This element of the document "`ex/2001/x3047.xml`" is an example of a high exhaustivity and specificity (3,3) element for INEX adhoc topic 165. Since elements are the standard unit of retrieval at INEX, the retrieval system must choose between this paragraph and the subsection that contains it. Unfortunately, the subsection may be too broad, but the paragraph may be too narrow.

XPath is considerably more expressive, and it may be fruitful to enlarge the class of possible results to include more of its features. Furthermore, I believe users would be better served by increasing the the set of potential results beyond single document elements. In particular, I believe it would be beneficial to express retrieval results as ranges of elements or text, for example as the "first three paragraphs in section 8," an approach which better reflects the way in which people informally describe portions of books and other documents. This short opinion paper marshalls a modest amount of evidence in support of range results, and briefly examines the availability of appropriate facilities in XPath to support these ranges.

2. RANGE RESULTS AT INEX

The potential benefits of range queries may be seen in the INEX 2004 adhoc relevance judgments. Of the 5229 elements judged as highly exhaustive and specific, at least 1700 (32%) are part of larger range of elements with identical tag names.

For example, the paragraph given above is part of the larger range of (3,3) elements

```
/article[1]/bdy[1]/sec[5]/ss1[1]/p[3]
/article[1]/bdy[1]/sec[5]/ss1[1]/p[4]
/article[1]/bdy[1]/sec[5]/ss1[1]/p[5]
/article[1]/bdy[1]/sec[5]/ss1[1]/p[6]
```


INEX Workshop on Element Retrieval Methodology, July 2005

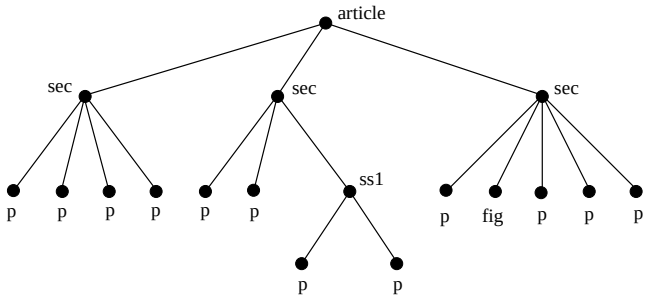


Figure 1: ***Example XML tree.***

By taking some liberties with XPath, this range of elements might be better expressed as

```
/article[1]/bdy[1]/sec[5]/ss1[1]/p[3 to 6]
```

It may be that this range is a more appropriate result than the individual paragraphs or the entire subsection — which is also a (3,3) element.

The inclusion of ranges in INEX retrieval results would necessitate changes to evaluation metrics and methodology, and the technique of assessing individual elements may have to be abandoned. Nonetheless, I believe it is feasible to extend the current INEX approach in a reasonable fashion, without introducing additional complexity. I outline one proposal next.

While struggling with the relevance assessment tools for INEX, I have often wished for a yellow highlighter that would allow me to directly select a section of a document for judging. Imagine a document marked up with a highlighter to indicate relevant regions. Potentially, each highlighted region could be labeled with exhaustivity and specificity attributes, and the relevance of an element (or a range of elements) could be determined from the attributes and proportion of highlighted text it contains. Moreover, if highlighting is permitted at the level of individual sentences and words, explicit labeling of specificity becomes unnecessary, since we may assume that only highly specific regions will be highlighted.

3. RANGE RESULTS IN XPATH

While we might informally speak of "the first three paragraphs" of a section, this statement can have many interpretations in the formal context of XML trees and XPath expressions. Consider the document in figure 1. While the

meaning of the statement is clear with respect to the first section, the same is not true of the second and third sections, since the third paragraph of second section is contained within a subsection and the first three paragraphs of the third section include a figure.

Straightforward attempts to specify "the first three paragraphs" in XPath further illustrate the ambiguities. For example, the expression

```
/article[1]/section[2]/p[position() <= 3]
```

will not include the first paragraph of the subsection. If it is correct to include this paragraph, an expression such as

```
/article[1]/section[2]//p[position() <= 3]
```

would be necessary. Similarly, the expression

```
/article[1]/section[3]/p[position() <= 3]
```

includes the first three paragraphs of the third section, but excludes the figure. Depending on the topic and the document, the inclusion of the figure may or may not be desired.

Other aspects of XML and XPath further complicate the specification of range results. Elements with the same logical type may receive different tag names. For example, in the current INEX adhoc test collection the tags "`p`" and "`ip1`" both indicate paragraphs. If ranges are to be accepted as retrieval results, then our method of expressing these ranges must accommodate this difference. On the other hand, a general ability to accept any XPath expression as a retrieval result is probably unneeded. A retrieval result that represents the second paragraph in every section

```
//sec/p[2]
```

is likely to have little value in a document-oriented context.

In XPath 2.0, it is relatively simple to express a range in a document as a pair of endpoints. Given two location paths, X and Y, the expression

X`/following::*[. <<` Y`]`

includes all the elements between them. In order to accept ranges as INEX retrieval results it may be sufficient to represent them as an (X, Y) pair. Using this representation, it would not be possible to exclude undesirable elements, such as the figure in third section of our example, but most of the potential benefits of range results could be realized.

4. SUMMARY

Constraining INEX results to single elements unnecessarily eliminates some of the potential benefits of XML retrieval, possibly forcing retrieval systems to return inappropriately narrow or broad results. Ranges are a natural means of specifying portions of documents and should be supported at INEX.

5. ACKNOWLEDGMENT

Thanks to Frank Tompa for his comments on this topic and his assistance with XPath 2.0.

The simplest evaluation measures for XML information retrieval that could possibly work

Djoerd Hiemstra and Vojkan Mihajlović
University of Twente
Centre for Telematics and Information Technology
P.O. Box 217, 7500 AE Enschede, The Netherlands
{d.hiemstra, v.mihajlovic}@utwente.nl

ABSTRACT

This paper reviews several evaluation measures developed for evaluating XML information retrieval (IR) systems. We argue that these measures, some of which are currently in use by the INitiative for the Evaluation of XML Retrieval (INEX), are complicated, hard to understand, and hard to explain to users of XML IR systems. To show the value of keeping things simple, we report alternative evaluation results of official evaluation runs submitted to INEX 2004 using simple metrics, and show its value for INEX.

1. INTRODUCTION

The INitiative for the Evaluation of XML Retrieval (INEX) is a yearly evaluation effort aimed at providing an infrastructure and a framework for evaluating the performance of retrieval systems that offer effective access to content that is structured using extensible markup language (XML). As such, INEX provides a large XML test collection and appropriate scoring methods for the evaluation of content-oriented XML retrieval systems [6]. INEX was inspired largely by ground-breaking work on laboratory-style evaluation of information retrieval (IR) systems developed in the Cranfield experiments [17] and later in the Text REtrieval Conferences (TREC) [18].

1.1 Measuring IR performance

Following the TREC paradigm, the effectiveness of information retrieval systems is usually measured by the combination of *precision* and *recall*. Precision is defined by the fraction of the retrieved items that is actually relevant. Recall is defined by the fraction of the relevant items that is actually retrieved.

$$\text{precision} = \frac{r}{n} \qquad r\text{: number of relevant items retrieved}$$

n: number of items retrieved

$$\text{recall} = \frac{r}{R} \qquad R\text{: total number of relevant items}$$

Although precision and recall are defined for sets of items, they are in practice used on ranked lists of documents. One approach that is used in TREC is to report the precision of documents at several document cut-off points, that is, the precision at 10 documents retrieved, at 20 documents, etc. These measures are easy to understand by the user of an IR system. Furthermore, it makes good sense to average the precision at 10 documents retrieved of a number of queries, to arrive at an average precision at 10 documents over, say, 50 queries. Averaging over queries is essential, since we cannot possibly draw conclusions on the performance of the system on one query only. A second approach that is often used is to report precision at several recall points, so the precision when the system retrieved 10% of the relevant documents, precision when the system retrieved 20%, etc. Usually a fixed number of recall points is used: 10%, 20%, $\cdots$, 100%. Often, there is also a need to arrive at a single effectiveness measure averaged over both the ranked list and the queries. One might for instance calculate the precision at R (total number of known relevant documents for a query) and average those measures over the queries (for different values of R). This is called R-precision. One might also calculate precision at each natural recall level for a query, average those measures, and average the resulting measure over all queries, so-called mean average precision [8]. These approaches are implemented in an evaluation programme for TREC [1].

1.2 Robertson's compatibility argument

There has been a lot of debate in the past on evaluation metrics, and there are various problems with precision and recall [2, 9]: For instance, if there are only 10 known relevant document for a topic, is it useful then to report the precision at 20 documents retrieved which never exceeds 0.5? Or, if there are 7 known relevant documents for a topic, what would be the precision at 10% recall level? – the natural levels of recall are in this case: $1/7$, $2/7$, $\cdots$, $7/7$, so we need some form of interpolation. Or, does it make sense, once we use interpolation, to average precision at 10% recall level over, say, 50 queries if those queries have a widely varying number of known relevant documents? etc.

When choosing an evaluation measure for a task, one might take these problems and arguments into consideration and make a personal decision. However, Robertson [15] raises a convincing reason for researchers to *not* make such personal decisions unless there is a very good reason for them to do so:

> (...) there is a strong compatibility argument for researchers to use the same methods as each other unless there is very good reason to depart from the norm.

This raises the following question: Are there reasons for INEX to depart from the norm? If so, what are those reasons, and, are they good enough to make different decisions

than the researchers that paved the way of laboratory-style IR system evaluation?

1.3 Is XML IR more complex for evaluation?

When using precision and recall, one at least has to make the following two assumptions.

- relevance is a binary property (items are relevant or not)
- the relevance of one item is independent of other items in the collection.

Additionally, when using the methods described above for measuring precision (and recall) for ranked lists, the following assumptions are made.

- a user spends approximately the same constant time on each retrieved element
- a user looks at one retrieved element after another from the ranked list and stops at some (arbitrary) point.

These assumptions might not be true for XML IR: We might be interested in more than just binary relevance (i.e., we are interested in specificity and exhaustiveness). The relevance of an element cannot possibly be independent of, for instance, its parent: XML elements overlap and are not separate units. Furthermore, the size of the retrieved elements vary, so the time spent on each document is not a constant value. A linear ordering of results might not be realistic as the user would like to see all parts of the context document and not jump from one document to the other.

Recent papers have proposed several new evaluation metrics that address the issues listed above. These metrics incorporate the size of XML elements [7], the time for reading an XML element [4], user browsing behavior when searching XML [13], take overlap of elements and the so-called overpopulated recall base into account [11, 12]. In this paper we like to contribute to the evaluation metrics discussion of the INEX methodology workshop by supporting the following statement: "There already exists a plethora of metrics so new metrics are not of interest, what is of interest is the identification of what should be measured."[1] More specifically, we emphasise the value of Robertson's compatibility argument in the discussion.

2. EVALUATION METRICS IN INEX

In this section we give an overview of the metrics used for INEX 2002 – 2005, and depict some of the metrics proposed for future usage. We start with relevance dimensions used for the relevance assessments and in the specification of quantisation functions used in these metrics. Readers familiar with INEX might skip this section to go directly to Section 3.

2.1 Relevance dimensions

In INEX relevance assessments, two relevance dimensions are used for evaluating XML elements: exhaustivity and specificity. Exhaustivity (E) is the extent to which the document component discusses the topic of request (more-or-less similar to traditional topical relevance as measured in TREC). Specificity (S) is the extent to which the document component *focuses* on the topic of request (i.e., if the component also contains a lot of irrelevant information specificity goes down). For most of the metrics, to produce the final evaluation result, e.g., recall-precision graph, the two dimensional relevance assessments are mapped to one dimensional relevance scale by employing a quantisation function, $f_{quant}(e,s) : ES \rightarrow [0,1]$, where ES denotes the set of possible assessment pairs $(e,s) : ES = \{(0,0),(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3)\}$[2]. Each XML element can be marginally (1), fairly (2), or highly (3) exhaustive or specific, or not relevant (denoted with pair (0,0)).

2.2 INEX 2002 metric: inex_eval

The INEX 2002 metric (also called inex_eval) computes the so-called *precall* measure, proposed by Raghavan et al. [14], on returned XML elements using the probability that the element viewed by the user is relevant ($P(rel|retr)$):

$$P(rel|retr)(x) = \frac{x \cdot n}{x \cdot n + esl_{x \cdot n}} \tag{1}$$

where $esl_{x \cdot n}$ denotes the expected search length [3], i.e. the expected number of non-relevant elements retrieved until an arbitrary recall point x is reached, and n is the total number of relevant elements with respect to a given topic. The expected search length is specified using the following formula:

$$esl_{x \cdot n} = j + \frac{s \cdot i}{r+1} \tag{2}$$

where j is the total number of non-relevant elements in all levels preceding the final level, s is the number of relevant elements required from the final level to satisfy the recall point, i is the number of non-relevant elements in the final level, and r is the number of relevant elements in the final level. The term level is used here to denote the set of elements that have the same rank in the retrieval process (see weak ordering in [3]).

Two quantisation functions are used for mapping relevance dimensions: f_{strict} (Equation 3) and $f_{general}$ (Equation 4). Strict quantisation function is used to evaluate retrieval methods with respect to their capability of retrieving highly exhaustive and highly specific XML elements, while general quantisation rewards methods that retrieve XML elements according to their degree of relevance.

$$f_{strict}(s,e) = \begin{cases} 1 & \text{if } e = 3 \text{ and } s = 3, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$f_{general}(s,e) = \begin{cases} 1 & \text{if } (e,s) = (3,3), \\ 0.75 & \text{if } (e,s) \in \{(2,3),(3,\{2,1\})\}, \\ 0.5 & \text{if } (e,s) \in \{(1,3),(2,\{2,1\})\}, \\ 0.25 & \text{if } (e,s) \in \{(1,2),(1,1)\}, \\ 0 & \text{if } (e,s) = (0,0) \end{cases} \tag{4}$$

[1] From the INEX Methodology Workshop call for papers.

[2] Note that in INEX 2002 exhaustivity was termed relevance, and instead of specificity a slightly different relevance dimension was used, termed coverage.

As can be seen in the definition of the generalized quantisation function, this function favors exhaustivity over specificity. The question is: Does this follow the user request as well as the assessment process on hierarchically structured XML documents? We can ask ourselves among fairly and marginally exhaustive and specific elements, which dimension is more important for the system's effectiveness?

The features of the INEX 2002 metric is that it calculates recall based on the full recall-base that contains large amounts of overlapping elements. Additionally, INEX 2002 metrics ignore possible overlap between result elements and rewards the retrieval of a relevant component regardless if part of it has been seen already. To resolve these problems numerous metrics are proposed as we can see below.

2.3 INEX 2003 metric: inex_eval_ng

The INEX 2003 metrics (also called inex_eval_ng) tries to overcome the overlapping problem of 2002 metrics by incorporating component size and overlap within the definition of recall and precision [7]. However it does not address the problem of overlapping XML elements in the assessments results, i.e., overpopulated recall-base [12]. Overlap is surpassed by considering only the increment in text size of the elements that are already seen. The metric assumes that the relevant information is distributed uniformly through a component which is a strong assumption that is not proven correct in practice.

Recall and precision for inex_eval_ng measure are computed as follows:

$$recall_o = \frac{\sum_{i=1}^{k} e(c_i) \cdot \frac{|c'_i|}{|c_i|}}{\sum_{i=1}^{N} e(c_i)} \tag{5}$$

$$precision_o = \frac{\sum_{i=1}^{k} s(c_i) \cdot |c'_i|}{\sum_{i=1}^{k} |c'_i|} \tag{6}$$

where elements $c_1, c_2, ..., c_n$ represent a ranked result list, N is the total number of elements in the collection, $e(c_i)$ and $s(c_i)$ denote the quantised assessment values of element c_i according to the exhaustivity and specificity dimensions respectively, $|c_i|$ denotes the size of the element, and $|c'_i|$ is the size of the element that has not been seen by the user previously. $|c'_i|$ can be computed as:

$$|c'_i| = |c_i - \bigcup_{c \in C[1,n-1]} (c)| \tag{7}$$

where n is the rank position of $|c_i|$ and $C[1, n-1]$ is the set of elements retrieved between the ranks $[1, n-1]$.

Quantisation functions are defined in such a way that they provide separate mapping for exhaustivity and specificity: $f'_{quant}(e) : E \rightarrow [0, 1]$ and $f'_{quant}(s) : S \rightarrow [0, 1]$. For the strict case the result of the quantisation functions is one if $e = 3$ or $s = 3$, respectively. For the generalized case quantisation functions are defined as: $f'_{general}(e) = e/3$ and $f'_{general}(s) = s/3$.

The problem of INEX 2003 metric is because relevance dimensions are treated in isolation while they both are required in order to identify the most appropriate unit of retrieval according to the retrieval task definition [11].

2.4 INEX 2004 metric: specificity-oriented and exhaustivity-oriented quantisation

Based on the discussion during INEX 2003 [11] on quantisation functions and drawbacks of INEX 2003 metrics, for strict quantisation two additional classes of exhaustivity-oriented and specificity-oriented quantisation functions are defined. Exhaustivity-oriented functions apply strict quantisation with respect to the exhaustivity dimension, allowing different degrees of specificity (Equation 8) or only fairly and highly specific elements (Equation 9).

$$f_{e3\_s321}(s, e) = \begin{cases} 1 & \text{if } s \in \{3, 2, 1\} \text{ and } e = 3, \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$f_{e3\_s32}(s, e) = \begin{cases} 1 & \text{if } s \in \{3, 2\} \text{ and } e = 3, \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Similarly, specificity-oriented functions apply strict quantisation with respect to the specificity dimension, allowing different degrees of exhaustivity (Equation 10) or only fairly and highly exhaustive elements (Equation 11). However, both quantisation function classes suffer from overlap problem.

$$f_{s3\_e321}(s, e) = \begin{cases} 1 & \text{if } e \in \{3, 2, 1\} \text{ and } s = 3, \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$f_{s3\_e32}(s, e) = \begin{cases} 1 & \text{if } e \in \{3, 2\} \text{ and } s = 3, \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

2.5 XCG: Extended Cumulative Gain

Criticizing the INEX 2002 generalized quantisation function, which is exhaustivity oriented, Kazai et al. [12] defined a specificity-oriented quantisation function to address the focused retrieval. This quantisation function should better reflect the user behavior and evaluation criterion for XML retrieval as defined in INEX [5]. It assumes that the specificity plays more dominant role than exhaustivity:

$$f'_{general}(s, e) = \begin{cases} 1 & \text{if } (e, s) = (3, 3), \\ 0.9 & \text{if } (e, s) = (2, 3), \\ 0.75 & \text{if } (e, s) = \{(1, 3), (3, 2)\}, \\ 0.5 & \text{if } (e, s) = (2, 2), \\ 0.25 & \text{if } (e, s) = \{(1, 2), (3, 1)\}, \\ 0.1 & \text{if } (e, s) = \{(2, 1), (1, 1)\}, \\ 0 & \text{if } (e, s) = (0, 0) \end{cases} \tag{12}$$

The extended cumulative gain (XCG) measure is based on cumulative gain (CG) measure [10]. The cumulative gain at the rank i, $CG[i]$, is computed as the sum of the relevance scores, $G[j]$, up to that rank:

$$CG[i] = \sum_{j=1}^{i} G[j] \quad (13)$$

An ideal gain vector, I, is than computed by summing rank values of all elements in the recall-base in decreasing-order of their degree of relevance. By dividing the CG vectors with the ideal vector I we obtain the normalized, nCG, relevance measure. The area between the normalized actual and ideal curves represents the quality of a retrieval approach.

Ideal recall base in extended cumulative gain metrics (XCG) is formed by selecting result elements from the full recall-base based on a given quantisation function and assuming that the component that has the highest score on the relevant XML path is chosen. In case two components on the same path have the same score, the one deeper in the XML tree is chosen (following the focused retrieval approach). XCG then uses uses full recall-base to enable scoring of near misses.

To define the relevance score of an element using XCG a result-list dependent relevance-value function is used:

$$rv(c_i) = f(quant(assess(c_i))) \quad (14)$$

where $assess(c_i)$ is a function that returns the assessment value pair for the element c_i, and $quant(assess(c_i))$ is a chosen quantisation function. Function f has three different variants. In case current element has not been evaluated before $f(x) = x$, where $x = quant(assess(c_i))$. In case an element has been seen before $f(x) = (1 - \alpha) \cdot x$. Here α is a factor that simulates user behavior with respect to the already seen elements. Finally, in case c_i has been seen in part then $f(x) = \alpha \cdot \frac{\sum_{j=1}^{m}(rv(c_j)\cdot|c_j|)}{|c_j|} + (1 - \alpha) \cdot x$, where m is the number of c_i's relevant child nodes. Additional normalization function is needed to disable that the total score of any group of descendant nodes of an ideal result element exceed the score achieved by retrieving the ideal element.

Therefore, in the extended cumulative gain (XCG) [12] the authors separated the model of user behavior from the actual metric employed via the definition of a set of relevance value (RV) functions, implementing scoring mechanisms based on parameters including e.g., the relevance degree of a retrieved element, the ratio of already viewed parts. Each RV function should model different user behaviors when searching for information. However, the weakness of the XCG metric is that the proper relevance-value function is still an open issue, and in handling the situation when the actual and ideal CG curves meet, as the interpretation of the curves after this point requires further studies [12].

2.6 Discussion and some more metrics

The INEX metrics briefly explained in this section raise some interesting issues. There might be some "very good reasons" to use these measures if traditional measures do not apply. Clearly, the section demonstrates that there is a lot of debate on evaluation metrics for XML IR. In fact, there are alternative proposals that are worth mentioning as well.

Tolerance to Irrelevance

The main idea is that the retrieval system needs to provide the user with an entry-point into the document that is close to the relevant information [4]. Thus, the system should produce the ranked list of entry points. The user reads the (part of a) document starting from the entry-point until his tolerance to irrelevance has been reached (specified using tolerance to irrelevance parameter), and then continue with the next ranked result. This measure aims at focused retrieval as it favors the systems that bring the user closer to the relevant information and avoid returning too large fragments. The drawback of this measure is that tolerance to irrelevance parameter has to be calibrated based on experimental studies.

Expected Ratio of Relevant Documents

The expected ratio of relevant documents (ERR) measure provides an estimate of the expectation of the number of relevant elements a user sees when looking at the list of the first k returned elements, divided by the expectation of the number of relevant elements a user would see when looking at all elements in the collection [13]. The value of ERR for each k between 1 and the total number of retrieved elements is given by:

$$ERR = \frac{\mathbb{E}[N_R|N = k]}{\mathbb{E}[N_R|N = E]} \quad (15)$$

where $N_R|N = k$ represents the total number of relevant elements the user has access to within the first k elements in the result list, and $N_R|N = E$ represents the total number of relevant elements within the whole collection. This computation is based on hypothetical user behavior assumption used in traditional IR: (1) the user browse through the retrieved document's structure, jumping with a specific probability to other elements in the structure, and (2) this browsing is influenced by the specificity of the returned elements. The drawback of this metric is the number of parameters that need to be estimated, simulating user's browsing behavior, for relevance computation.

In the next section, we explore the usefulness of simple evaluation metrics based on cut-offs in the ranked list.

3. ANALYSING INEX RUNS WITH SIMPLE METRICS

In this section we will report simple evaluation results of the official INEX 2004 runs using simple evaluation measures. We will take the following decisions.

- Our quantisation functions will map exhaustivity and specificity to a binary measure: relevant or not relevant. We do not use generalised quantisation measures.
- We will only report average precision at fixed cut-off values. This way, at least for small cut-off values, our measures do not depend on the total number of relevant items, thereby partly avoiding the "overpopulated recall base" problem.
- We will report set-based overlap for (the same) fixed cut-off values, not only for the total retrieved list (usually 1500 elements) as was done for INEX 2004. This way, we are able to distinguish a system that tries to identify elements from different articles from one that retrieves many from a single article.

The following quantisation functions were used: *strict* (Equation 3), *exhaustive* (Equation 8), *specific* (Equation 10), and finally *liberal* (Equation 16).

$$f_{liberal}(s,e) = \begin{cases} 1 & \text{if } s \in \{3,2\} \text{ or } e \in \{3,2\} \\ \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

Set-based overlap is defined as in INEX 2004 [5]:

$$\frac{|\{e_1 \in R | \exists e_2 \in R \wedge e_1 \neq e_2 \wedge overlap(e_1, e_2)\}|}{|R|} \quad (17)$$

where R is a result list, $overlap(e_1, e_2)$ is true if these two elements, e_1 and e_2, are overlapping one another, i.e., if they are nested.

The measures reported are easy to explain. For instance, if for strict quantisation and cut-off value 10 we report precision 0.25 and overlap 0.6; then this would be communicated to a user or potential customer as: "Of the first ten retrieved elements, our system produces on average two-and-a-half relevant element. On average, six out of ten elements overlap with another element in the first ten."

3.1 Content-only (CO) runs

The INEX content-only task provides queries without any structural constraints. In this task, the system needs to identify the most appropriate XML element for retrieval. The task resembles that of users that want to search XML data without knowing the schema or DTD. In this section, we select the evaluation of some runs which we believe show quite different behaviour when compared to each other.

	average precision				
cut-off	strict	liberal	exhaust.	specific	overlap
5	0.200	0.577	0.359	0.329	0.682
10	0.162	0.547	0.297	0.329	0.768
20	0.146	0.506	0.266	0.306	0.799
30	0.134	0.477	0.226	0.313	0.847
100	0.087	0.337	0.142	0.239	0.894
200	0.062	0.244	0.099	0.175	0.908
1500	0.016	0.073	0.027	0.051	0.906

Table 1: Precision and overlap of CO run `ibmhaifa3`

Table 1 shows average precision values per cut-off value for each quantisation function, as well as the overlap per cut-off value of the best (best according to the official INEX measures, but also the best according to the measures reported in this section) INEX 2004 content-only (CO) run (`ibmhaifa3`, `CO-0.5-LAREFIENMENT`). The evaluation shows that among the first 5 elements retrieved there is at least 1 relevant element (strict quantisation) up to almost 3 relevant elements (liberal quantisation). Interestingly, the overlap is quite high for all cut-off values. Overlap goes up steadily for this run from 68% for cut-off 5 to more than 90% for the whole list of 1500 documents. All runs with high precision values have quite some overlap.[3] Also interestingly, when focussing on specificity (Equation 10), the precision values do not change a lot for cut-offs 5, 10, 20 and 30 elements retrieved; however, precision goes down for exhaustiveness-oriented quantisation.

[3]Our overlap for cut-off 1500 differ considerably from the ones reported by INEX, maybe because we ignored results for which no assessments were done, i.e., precision values and overlap are calculated on the same set of 34 CO topics.

	average precision				
cut-off	strict	liberal	exhaust.	specific	overlap
5	0.112	0.341	0.194	0.200	0.059
10	0.085	0.306	0.159	0.168	0.094
20	0.063	0.246	0.115	0.138	0.125
30	0.055	0.230	0.102	0.131	0.170
100	0.041	0.164	0.073	0.102	0.364
200	0.028	0.127	0.055	0.077	0.509
1500	0.011	0.045	0.023	0.027	0.868

Table 2: Precision and overlap of CO run `lip63`

Table 2 reports a run with different behaviour. This run, `lip63`, `bn-m1-eqt-porder-eul-o.df.t-parameters-00700)`, performs worse than the previous run. INEX reported a similar amount of overlap in the retrieved list (1500 elements) of this run, however, the run does not show a lot of overlap for the initial cut-off values.

For all CO runs we investigated, overlap was either relatively constant, or going up quickly when approaching the 1500 elements that could be submitted. Some runs (e.g. `ucalif0`, (`CO-3`) did not submit 1500 elements for each topic. For those runs, precision and overlap at 1500 were calculated by assuming that the elements that could have been submitted, but were not submitted are not relevant and do not overlap with another element in the retrieved list. This leads to low overlap values at cut-off 1500 as shown in Table 3. One might argue that if the precision at 1500 is identical for two systems, the one that has stopped retrieving when it expects no more relevant elements (and therefore has low overlap at 1500) should be preferred over one that filled all slots with overlapping elements (resulting in high overlap at 1500). Interestingly, this run initially performs better on exhaustivity-oriented quantisation than on specificity-oriented quantisation.

	average precision				
cut-off	strict	liberal	exhaust.	specific	overlap
5	0.172	0.382	0.300	0.218	0.700
10	0.135	0.318	0.235	0.185	0.656
20	0.094	0.262	0.175	0.150	0.707
30	0.072	0.222	0.138	0.130	0.714
100	0.034	0.134	0.068	0.083	0.711
200	0.019	0.085	0.040	0.053	0.592
1500	0.004	0.016	0.008	0.010	0.199

Table 3: Precision and overlap of run `ucalif0`

	average precision				
cut-off	strict	liberal	exhaust.	specific	overlap
5	0.082	0.329	0.153	0.194	0.000
10	0.085	0.285	0.144	0.179	0.000
20	0.062	0.224	0.110	0.141	0.000
30	0.053	0.202	0.096	0.131	0.000
100	0.029	0.114	0.048	0.078	0.000
200	0.017	0.074	0.028	0.052	0.000
1500	0.004	0.019	0.007	0.013	0.000

Table 4: Precision and overlap of run `utampere0`

Finally, Table 4 shows run `utampere0` (`UTampere_CO_average`), the best run according to the XCG evaluation measure. This run does not show any overlap at all. Interestingly, this run performs better on specificity-oriented quantisation than on exhaustivity-oriented quantisation.

	cut-off at 10			cut-off at 100		
run id	precision	overlap	rank	precision	overlap	rank
ibmhaifa3	0.334	0.768	1	0.201	0.894	1
ibmhaifa0	0.323	0.718	2	0.195	0.881	2
uwaterloo0	0.300	0.806	3	0.133	0.899	9
uamsterdam1	0.288	0.935	4	0.158	0.956	3
ibmhaifa4	0.285	0.665	5	0.153	0.853	4
cmu0	0.214	0.618	17	0.149	0.814	5
uwaterloo1	0.273	0.785	6	0.107	0.904	16
uamsterdam0	0.266	0.882	7	0.139	0.929	6
qutau0	0.263	0.888	8	0.126	0.942	11
cmu2	0.184	0.621	23	0.137	0.851	7

Table 5: Well-performing INEX 2004 CO runs: average precision at cut-off 10 and 100 averaged over 4 quantisations

Table 5 shows the best-performing runs according to precision at 10 and precision at 100 averaged over all four quantisations. The top 4 runs correspond with the top 4 as presented by the official INEX measures. All runs have a relatively high number of overlap at cut-off 10 and 100. It seems to be impossible to achieve high precision without a considerable amount of overlap in the retrieved elements. It is therefore questionable if these top runs are also the most useful from a user-perspective. A measure that somehow combines precision and overlap in a single measure, for instance the XCG measure, might be desirable.

3.2 Vague content-and-structure (VCAS) runs

The vague content-and-structure task (VCAS) provides queries that besides query terms also contain structural constraints. This task resembles that of users or applications that do know the schema or DTD, and want to search some particular XML elements while formulating restrictions on some (other) elements.

	average precision				
cut-off	strict	liberal	exhaust.	specific	overlap
5	0.239	0.500	0.354	0.346	0.554
10	0.204	0.458	0.304	0.319	0.677
20	0.165	0.431	0.290	0.273	0.769
30	0.142	0.409	0.271	0.254	0.767
100	0.100	0.309	0.180	0.201	0.836
200	0.079	0.237	0.134	0.159	0.900
1500	0.030	0.087	0.047	0.060	0.830

Table 6: Precision and overlap of run `qutau4`

Table 6 shows average precision values per quantisation function and cut-off value, and the overlap per cut-off value of the best (best according to the official INEX measures) INEX 2004 vague content-and-structure (VCAS) run (`qutau4`, `VCAS_PS_stop50K_049025`). On all cuf-off points, the measured overlap is quite high, going from initially 55% to 90 % overlap. The run shows almost equal performance of the specificity-oriented quantisation and the exhaustiveness-oriented quantisation methods.

The run in Table 7 (`utwente2`, `LMM-VCAS-Relax-0.35`) shows different behaviour. First, the overlap does not exceed 30% for most document cut-offs. Second, the run seems to do somewhat better on the specificity-oriented quantisation method than on the exhaustiveness-oriented quantisation method. The run has higher precision at the early cut-offs than the run from the previous example, but lower precision at later cut-offs.

	average precision				
cut-off	strict	liberal	exhaust.	specific	overlap
5	0.246	0.515	0.339	0.377	0.177
10	0.223	0.496	0.300	0.365	0.215
20	0.190	0.444	0.250	0.325	0.240
30	0.146	0.383	0.201	0.269	0.242
100	0.080	0.240	0.107	0.162	0.280
200	0.059	0.162	0.074	0.117	0.297
1500	0.021	0.048	0.026	0.038	0.316

Table 7: Precision and overlap of run `utwente2`

Interestingly, the best VCAS runs show similar absolute performance figures as the best CO runs. Appearently, the CO task is not inherently more difficult than the VCAS task. However, whereas all good CO runs have high overlap, some good VCAS runs actually have low overlap. This leads us to the following hypothesis: Structured queries can be used as a means to remove overlap (redundancy) from the result list without loosing much precision.

Like the University of Tampere in the previous section, the University of Amsterdam explicitly experimented with systems that do not produce any overlap at all. The run in Table 8 (`uamsterdam4`, `UAms-CAS-T-FBack-NoOverl`) has zero overlap at all cut-off points. Interestingly, the same group also produced a run with some small overlap and a run with relatively high overlap that obtain higher precision than this run. Removing all overlap seems to result in lower precision, even at small element cut-off values.

	average precision				
cut-off	strict	liberal	exhaust.	specific	overlap
5	0.115	0.400	0.239	0.262	0.000
10	0.096	0.335	0.192	0.204	0.000
20	0.106	0.281	0.169	0.196	0.000
30	0.100	0.263	0.155	0.186	0.000
100	0.066	0.171	0.095	0.126	0.000
200	0.047	0.124	0.067	0.093	0.000
1500	0.017	0.036	0.021	0.030	0.000

Table 8: Precision and overlap of run `uamsterdam4`

Table 9 shows the best-performing runs according to precision at 10 and precision at 100 averaged over all four quantisations. For VCAS runs, there is quite some difference between the top precision at 10 runs and the top precision at 100 runs. The top 4 runs for precision at 100 correspond with the top 4 as presented by the official INEX measures. Interestingly, the runs show quite some variation in overlap. Some runs have an overlap of about 90 % (e.g. `qutau4`, `VCAS_PS_stop50K_049025`), whereas others have an overlap of no more than 30 % (e.g. `utwente1`, `LMM-VCAS-Strict-0.35`).

run id	cut-off at 10 precision	overlap	rank	cut-off at 100 precision	overlap	rank
utwente2	0.346	0.215	1	0.147	0.280	7
qutau3	0.338	0.915	2	0.180	0.924	3
uamsterdam5	0.332	0.239	3	0.146	0.283	9
qutau5	0.332	0.877	4	0.190	0.949	2
qutau4	0.321	0.677	5	0.196	0.836	1
utwente1	0.318	0.150	6	0.127	0.254	12
ibmhaifa1	0.316	0.465	7	0.150	0.539	6
uamsterdam3	0.296	0.877	9	0.172	0.918	4
cmu5	0.205	0.581	21	0.150	0.770	5

Table 9: Well-performing INEX 2004 VCAS runs: average precision at cut-off 10 and 100 averaged over 4 quantisations

4. PROPOSALS FOR DISCUSSION

In this paper we showed some examples of how simple evaluations measures can give insight in XML IR. We believe that precision at document cut-offs – which has been part of the standard TREC evaluation metrics repertoire since the very start of TREC in 1992 – is an elegant simple measure, that is easily explained. Following Robertson's compatibility argument [15], there is no good reason to *not* report this measure in the official INEX evaluation reports. Since it is part of standard practice in IR system evaluation, this measure should be reported by INEX as well. Note that precision at cut-offs suffers less from the "overpopulated recall base" problem since it does not use the total number of relevant elements in its calculation.

In analogy to reporting the precision at cut-offs, we also reported the overlap at cut-offs. Here, Robertson's argument does not fully apply: overlap is a problem that is relatively new to IR. Simply reporting overlap for the same cut-offs as precision seems to be "closest" to the norm. In future studies, we plan to investigate overlap further. For instance, the current overlap definition seems, at least in theory, somewhat unstable. Suppose a run retrieves 1499 non-overlapping elements and as its first element the collection root (let's assume that would be possible) than the measured overlap would be 100 % at each cut-off point. Maybe a probabilistic overlap version can be adopted such as the probability that two elements in the list overlap.

Precision and overlap at cut-off points give some interesting insights. Overlap varies a lot over different cut-off points for some runs. It seems that overlap plays a different role in the CO task than in the VCAS task. However, overlap is not exclusively a problem in the CO task. In fact, some interesting observations can be made on the relation between overlap and precision in the VCAS task. All of this is, fortunately, in line with the official results as reported by INEX.

So, what about the existing INEX measures? We feel that XML IR does not give a "very good reason" to prefer Raghavan et al.'s [14] precall measure over the more standard precision at fixed recall points measures. Following Robertson's compatibility argument, choosing this measure as the basis of `inex_eval` seems an odd decision at the time first INEX workshop, but one might argue now that the measure is retrospectively the norm for XML IR because of INEX. Furthermore, Raghavan's version of mean average precision (using strict quantisation) is only a slight deviation of the TREC version of mean average precision. We feel that the alternatives briefly explained in Sections 2.3 and 2.5, that is, the `inex_eval_ng` and XCG measures, are interesting for XML IR. There might be some "very good reasons" to use these new measures. However, in our (non-scientific) opinion these measures are hard to grasp for IR system users, and even so for IR system researchers. In fact, computer science researchers do not have much more skills than ordinary users as nicely pointed out by Trotman and O'Keefe [16] who showed that many researchers that participate in INEX make errors in specifying their queries in XPath. Similar to Trotman and O'Keefe's query language problem, we should ask ourselves: "What would be the simplest approach that could possibly work?"

Acknowledgements

Many thanks to anonymous reviewers for given useful feedback. The research presented in this paper was funded by the Netherlands Organisation for Scientific Research (NWO) and the Dutch BSIK program MultimediaN.

5. REFERENCES

[1] C. Buckley. The trec_eval evaluation program. *Available for TREC participants*. http://trec.nist.gov.

[2] C. Buckley and E.M. Voorhees. Evaluating evaluation measure stability. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 33–40, 2000.

[3] W. Cooper. Expected search length; a single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation*, 19(1):30–41, 1968.

[4] A.P. de Vries, G. Kazai, and M. Lalmas. Tolerance to irrelevance: A user-effort oriented evaluation of retrieval systems without predefined retrieval unit. In *RIAO Conference Proceedings*, pages 463–473, 2004.

[5] A.P. de Vries, G. Kazai, and M. Lalmas. Evaluation metrics 2004. In *Proceedings of the 3rd INEX Workshop, LNCS 3493, Springer*, 2005.

[6] N. Fuhr and M. Lalmas. Report on the INEX 2003 workshop. *SIGIR Forum*, 38(1), 2004.

[7] N. Gövert, G. Kazai, N. Fuhr, and M. Lalmas. Evaluating the effectiveness of content-oriented XML retrieval. Technical Report Computer Science 6, Technischer bericht, University of Dortmund, 2003.

[8] D.K. Harman. Appendix b: Common evaluation measures. In *Proceedings of the 13th Text Retrieval Conference (TREC)*, 2005.

[9] D. Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th ACM Conference on Research and Development in Information Retrieval (SIGIR'93)*, pages 329–338, 1993.

[10] K. Järvelin and J. Kakäläinen. Cumulated gain-based evaluation of IR techiques. *ACM Transactions on Information Systems*, 20(4):551–556, 2002.

[11] G. Kazai. Report of the INEX 2003 metrics working group. In *Proceedings of the 2nd INEX Workshop*, ERCIM Publications, 2004.

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

[13] B. Piwowarski and P. Gallinari. Expected ration of relevant units: A measure for structured information retrieval. In *Proceedings of the 2nd INEX Workshop*, ERCIM Publications, 2004.

[14] V. Raghavan, P. Bollmann, and G. Jung. A critical investigation of recall and precision. *ACM Transactions on Information Systems*, 7(3):205–229, 1989.

[15] S.E. Robertson. Evaluation in information retrieval. In M. Agosti, F. Crestani, and G. Pasi, editors, *European Summer School on Information Retrieval (ESSIR)*, number 1980 in Lecture Notes in Computer Science, pages 81–92. Springer-Verlag, 2000.

[16] A. Trotman and R.A. O'Keefe. The simplest query language that could possibly work. In *Proceedings of the 2nd Workshop of the INitiative for the Evaluation of XML retrieval (INEX)*, 2004.

[17] B.C. Vickery. *Techniques of Information Retrieval*. Butterworths, 1970.

[18] E.M. Voorhees and D.K. Harman, editors. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, 2005.

Understanding Content-and-Structure

Jaap Kamps[1,2] Maarten Marx[2] Maarten de Rijke[2] Börkur Sigurbjörnsson[2]
[1] Archives and Information Studies, University of Amsterdam, Amsterdam, The Netherlands
[2] Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands
{kamps,marx,mdr,borkur}@science.uva.nl

ABSTRACT

Document-centric XML is a mixture of text and structure. With the increased availability of document-centric XML content comes a need for query facilities in which both structural constraints and constraints on the content of the documents can be expressed. This has generated considerable interest in both the IR and DB communities, and has lead to the launch of evaluation efforts tailored for XML documents. One of the driving and long-standing research questions here is: How does the increased expressiveness of languages for querying XML documents help users to better, and more effectively, express their information needs? And closely related to this: How should we evaluate systems that enable users to express their information needs using both content and structural constraints?

In this paper we address these research questions. Our analysis follows two lines: What requirements can *in principle* be expressed in query languages for document-centric XML documents? And: How do users *actually* use such languages? For the former, we provide mathematical characterizations of two query languages, one for users with next to no knowledge of the document structure (ignorant users), and one for users that have some, but not complete, knowledge of the document structure (semi-ignorant users). To address the latter issue, we examine the topics formulated in the second query language as part of the 2004 edition of the INEX XML retrieval initiative. Our main findings are as follows: First, while structure is used in varying degrees of complexity, over half of the queries can be expressed in the very restrictive ignorant user language. Second, structure is used as a search hint, and not a search requirement, when judged against the underlying information need. Third, the use of structure in queries functions as a precision device. Fourth, the underlying retrieval task of content-and-structure querying is no different from the ordinary natural language query retrieval task. From those findings we derive a number of recommendations for the evaluation of systems that cater for content-and-structure queries.

1. INTRODUCTION

Increasingly, users, both expert and non-expert, have access to text documents, equipped with some semantic hints through XML-markup. How can we query such data? We could adopt a standard IR approach: perform best match querying using plain text queries. But this would not allow users to specify constraints on the document structure. Alternatively, we could query the documents using a database approach: perform exact-match using XPath queries. But here, effective query formulation is non-trivial and recall is often too low.

Within the INitiative for the Evaluation of XML Retrieval (INEX) [14], the two approaches are combined. Free text search functionality is added to XPath, in the form of a new `about` function. With the same (standard) syntax as the standard `contains` function, the `about` function has two main features; it allows us to (1) express information needs with a mixture of content and structure requirements; and (2) use best-match querying of document-centric XML.

How do users exploit the expressive power offered by such languages? User-oriented studies from INEX have shown that full XPath is too complex for querying document-centric XML documents, even for experts. Moreover, it is unrealistic to assume that casual users have full knowledge of the structure of the documents they want to query. We discuss several XPath fragments (extended with `about`) that are simpler and, we believe, more effective for querying document-centric XML for users.

Our aim in this paper is not to complement the proposed languages with an algebra and implementations—such issues are being addressed elsewhere, see e.g., [1, 9, 10, 23]. Instead, our main aim is to understand how the increased expressiveness of query languages tailor-made for querying XML documents can help users to better, and more effectively, express their information needs. We pursue this aim from two perspectives, subjecting these new query languages to a number of sanity checks: we need to understand their *expressive power*, and we need to assess the types of *information needs* they are meant to address. To deal with the former issue we relate the query languages to logical languages, for which expressiveness results are well-known. To deal with the latter issue we analyze a set of user queries formulated in the INEX query language, and the sets of elements judged relevant by these users, both made available through INEX. Our main findings are as follows: First, while structure is used in varying degrees of complexity, over half of the queries can be expressed in the very restrictive *ignorant user* language. Second, structure is used as a search hint, and not a search requirement, when judged against the underlying information need. Third, the use of structure in queries functions as a precision device. Fourth, the underlying retrieval task of content-and-structure querying is no different from the ordinary natural language query retrieval task. Building on our user-oriented findings, we address the second aim of this paper: understanding how we should eval-

uate systems that cater for content-and-structure queries; we derive recommendations concerning topics, metrics, and assessments.

In Section 2 we provide background on querying document-centric XML. In Section 3, we discuss content-oriented flavors of XPath and provide semantic characterizations of their expressive power. Section 4 describes the content-and-structure language used at INEX 2004, and analyzes the resulting queries and assessments. In Section 5, we describe how structure helps users improve the quality of retrieval results. In Section 6 we change tack and derive implications of our user-oriented findings for the evaluation of content-and-structure retrieval engines. We conclude in Section 7.

2. QUERYING XML

XML can be used to mark up content in various ways. Based on the content, XML documents are often categorized into two groups: *data-centric* and *document-centric*. The former contain highly structured data marked up with XML tags, an example being geographic data in XML [20]. Document-centric documents are loosely structured documents (often text) marked-up with XML, with electronic journals in XML providing important examples. For our experiments we use the document-centric XML collection that comes with the INEX test suite [14]. It contains over 12,000 articles from 21 IEEE Computer Society journals, marked up with XML tags. On average an article contains 1532 elements and the average element depth is 6.9. About 170 tag names are used, such as articles ⟨article⟩, sections ⟨sec⟩, author names ⟨au⟩, affiliations ⟨aff⟩, etc.

Whereas emerging standards for querying XML, such as XPath and XQuery, can be very effective for querying data-centric XML, another approach seems to be needed for querying document-centric XML. The latter task is a natural meeting point of two disciplines: the XML nature of the documents calls for methods from the database field for querying structure, and the textual nature of the documents calls for approaches from the field of information retrieval (IR) (cf. [31, Section 5]). It is interesting to contrast the two subtasks. As to querying structure, XML query languages such as XPath have a definite semantics. Judging whether an element satisfies an XPath query can be done by a machine (XPath processor), based on the pattern appearing in the XML document, using an *exact match* approach. It is clearly defined which nodes or elements match a given query. An XPath processor will return precisely these elements with no inherent ranking of results. In contrast, for querying text IR uses free text queries. These can be keywords or full sentences describing an information need. An IR system uses a *best match* approach: it attempts to rank results by their topical relevance to the user's query.

As pointed out above, several studies are dedicated to understanding the formal and/or computational properties of hybrid content and structure query languages. Our focus is different: on query languages as a means for users to express their information needs more precisely (as opposed to queries with no structural constraints).

3. THE EXPRESSIVE POWER OF CONTENT ORIENTED XPATH

To query document-centric XML documents we need a hybrid query language, in which content and structural requirements can be expressed and mixed. At INEX, an XPath-like query language has been proposed for this purpose. The syntax of the language looks like XPath, but does not have the same strict semantics. It can be seen as an *extension* of a *subset* of XPath.

In this section, we will first motivate why XPath needs to be *restricted* and examine some fragments of XPath (Section 3.1). We will then motivate why those fragments need to be *extended* with the `about` function (Section 3.2).

3.1 Restricting XPath

Experience from INEX has shown that people—in this case, academics familiar with query languages—have great difficulties in using (the navigational part of) XPath to formulate queries that combine content and structural aspects [24]. The restriction to navigational XPath was originally motivated by the fact that it is a widely used technology, whence it was assumed that it would be easily learnable. This assumption proved to be wrong.

Based on the extensive data described in [24], we argue that the cause of users' difficulties in writing content-and-structure queries can be traced back to a combination of two related items: (1) Users have no, or at best incomplete, knowledge of the structure of documents, i.e., of the DTD.[1] (2) Users have problems handling the expressive power of full XPath. In particular, the fact that the same query can be expressed in several fundamentally different ways proved problematic for users. These observations give rise to two constraints on XPath fragments: it should be possible to formulate information needs even with limited knowledge of the DTD, and the expressive power should be restricted.

A user's knowledge about a set of documents can be naturally formalized in terms of an *indiscernibility relation* over the elements selected by an XPath query: a binary relation that identifies elements in a document. What does such a relation have to do with query languages? We say that a language is *safe* or *well-designed* if indiscernible elements cannot be distinguished by an expression in the query language. This design criterion will help us single out natural XPath fragments. In fact, the fragments discussed below have a perfect fit with two user profiles formalized by an indiscernibility relation: not only are they safe, they are also complete in the sense that every first-order definable set of indiscernible elements can be defined in the language.

Below, we define two user profiles, both capturing users with limited knowledge of the DTD. First, we consider, what we call, *ignorant users* who only know the tag names. Second, we consider *semi-ignorant users*, who know the tag names and have some clue about the hierarchal structure of the elements, without knowing the full details. For both profiles we will design fragments that are *safe* for the sketched user profiles; we interpret this as saying that the chance that a user makes a semantic mistake when describing her information need in terms of XPath is minimal. For clarity, in this subsection we only consider the *navigational* part of XPath. The next subsection deals with the `about` function.

Ignorant Users. Users formulating queries at INEX did not have a clear idea of the DTD of the collection [24]. Typically, they browsed the documents and picked up some

[1]The DTD of the INEX XML document collection was extremely complex. There were 192 different content types, including 11 different tag names for representing paragraphs.

knowledge about the available tags in this manner. This can be viewed as an XML version of fielded search. For users who know (a subset of) the tag names, but do not (want to) know the structure of the documents, we create an XPath fragment which exactly fits their knowledge. Specifically, our ignorant user is able to ask questions like: "Give me sections about weather forecasting where an author is affiliated in California". In a hybrid XPath-like language this could be written as:

```
//sec[about(.,'weather forecasting') and
                          //aff[about(.,'California')]
```

More generally, the user can express her information need as a conjunction of two boolean formulas: one restricting the element of interest, and the other restricting the surrounding document. The following syntax, which we call *non-structure aware XPath* allows this. A query is of the form `//::tag[P]`, where `tag` is either the wild card `*` or a tag name, and `P` is a predicate created using '`and`,' '`or`,' and '`not`' from location paths of the form `//::tag`. Note that when // :: `t` is used in a filter it means "there exists a descendant of the root with tag `t`". I.e., `//::t` simply says that somewhere in the document there is a `t` element.

We turn to a semantic characterization of this fragment. In social network theory [32] several indiscernibility relations have been proposed, including the useful and robust notion of *bisimulation* (a.k.a. 'regular equivalence'). We need the following special "structurally unaware" version.

Definition 1. Let D, D' be documents and B a non-empty binary relation between the elements of D and D'. We call B a *structure unaware bisimulation* if, whenever xBy holds for two elements x, y in D, then

1. x and y have the same tag name;
2. if there exists an $x' \in D$, then there exists a $y' \in D'$ such that $x'By'$; and
3. conversely for $y' \in D'$.

Let $\phi(x)$ be a first-order formula (in one free variable) in a suitable vocabulary; $\phi(x)$ is *invariant under bisimulations* whenever the following holds: for any a, b and bisimulation B, if $\phi(a)$ and aBb hold, then $\phi(b)$ holds as well.

A few comments. First, since we are usually comparing elements within a single document, our notion of indiscernibility relation is an *auto*-bisimulation, where D and D' in Definition 1 are the same document. Secondly, in the usual definition of bisimulation, the clauses in items 2 and 3 above are conditioned on x' (and y') being "structurally" related to x (and y, respectively); but our ignorant user is not aware of the structure, hence we omitted these conditions.

Theorem 2. *1. Elements that are related by a structure unaware bisimulation cannot be distinguished by a non-structure aware XPath expression.*

2. Every first-order formula that is invariant under structure unaware bisimulations is definable by a non-structure aware XPath expression.

We can conclude that this language fits perfectly to the sketched user profile: the first part of the theorem states that it is *safe*, the second that it is *complete*.

Semi-Ignorant Users. For semi-ignorant users, we will define two equivalent XPath fragments. One coincides with the fragment proposed in [24] and is supported by the query working group at INEX 2003 [29]. We will show that these fragments have a meaningful semantic characterization. The fact that these fragments fits a common user profile is strong evidence for its naturalness.

Semi-ignorant users have some ideas about the hierarchical structure of the documents. E.g., they know that paragraphs are below sections but, as pointed out in [24], they need not know that there *can* be elements in between. For this reason, [24] proposes Positive Descendant XPath: the fragment of XPath in which only the descendant axis may be used and the booleans in the predicates are restricted to "`and`" and "`or`".

We sketch two possible ways in which semi-ignorant users might pose queries. Suppose a user is interested in 'bisimulation' theorems which appear in sections about 'XPath.' He knows about the theorem tag ⟨theorem⟩ and the section tag ⟨sec⟩; he also knows that theorems can be nested somewhere inside sections. This user might ask:

```
//sec[about(.,'XPath')]//theorem[about(.,          (1)
                                   'bisimulation')]
```

Another user might formulate the same need as:

```
//theorem[about(.,'bisimulation') and              (2)
          ancestor::sec[about(.,'XPath')]
```

The two users seem to engage in different mental processes when formulating their queries. The first thinks top-down: zoom in on a relevant section and then specify what sort of information should be retrieved from that section. The second approaches the problem bottom-up: determine a segment of interest and then think about sections that might contain the segment. The authors of this paper disagree on which scenario is more natural. Both scenarios can be captured in an XPath fragment, and we will show that the two fragments are equivalent.

To admit formulation (2) above, we need to allow both descendant and ancestor relations. We provide O'Keefe and Trotman's fragment [24] with a double characterization: a semantic one in terms of simulations, and a syntactic one, as a fragment of a well-known language in computer science, the temporal logic CTL. First, we need some definitions.

Definition 3. *Positive Temporal XPath* consists of queries of the form //`tag`[`P`], where `P` is in the following restriction of navigational XPath:

- the only axis relations are `descendant` and `ancestor`;
- only boolean `and` and `or` can be used in filters.

As none of the above two XPath fragments contains negations, bisimulation is too strong a notion [18]. As a general fact, positive fragments correspond to simulations, which are bisimulations from which one of the directions is dropped. We use $<$ to denote the descendant relation between elements; i.e., $x < y$ means that y is a descendant of x.

Definition 4. Let D, D' be documents and B a non-empty binary relation between the elements of D and D'. We call B a *temporal simulation* if, whenever xBy, then

1. x and y have the same tag names;

2. if there exists an $x' \in D$ such that $x < x'$, then there exists a $y' \in D'$ such that $y < y'$ and $x'By'$; and
3. similarly when $x' < x$.

Temporal simulations correspond to users that know the element hierarchy: note that both elements below *and* above have to be simulated. The next theorem is an analogue of Theorem 2 for Positive Descendant XPath: it is both safe and complete for semi-ignorant users.

THEOREM 5. *Let X be a set of nodes. The following are equivalent on trees.*

1. *X is definable by a first-order formula in one free variable in the signature with $<$ and unary predicates which is preserved under temporal simulations.*
2. *X is definable as the answer set of a Positive Descendant XPath formula.*
3. *X is definable as the answer set of a Positive Temporal XPath query.*

The proof uses ideas from modal logic [3, Theorem 2.78] together with ideas from [2, Theorem 3.2]. We conjecture that the language in item 3 of Theorem 5 is exponentially more succinct than the language in item 2.

3.2 Extending XPath

Now that we have looked at restrictions of the navigational part of XPath to "manageable" fragments, we look at extensions with the `about` function. Although `about` has the same syntax as the XPath function `contains`, their semantics are radically different. Because of its strict, boolean character, `contains` is not suitable for text rich documents. The semantics of `about` is meant to be very liberal. Consider the element `<aff>'Stanford University'</aff>`. A human assessor will likely decide that `about(.//aff,'California')` returns true if that element is below the node of evaluation; but an XPath processor equipped only with `contains` would have difficulties trying to do the same. As a more elaborate example, look at the following query (against a collection containing several articles):

> Find articles where the author is affiliated in California. From those articles return sections about weather forecasting systems.

In a hybrid syntax, mixing content and structure, this would be something like

```
//article[about(.//au//aff,'California')]//sec[
            about(.,'weather forecasting systems')]
```

This query has two content-based restrictions, linked by a structural constraint. The semantics of this query is not strict. In the spirit of information retrieval, the ultimate decision of relevance is in the hands of a human assessor, who may bring lots of context and world knowledge to her judgment. E.g., a human assessor is likely to judge a section about 'storm prediction systems' to be relevant to the information need expressed above.

4. EXPRESSING INFORMATION NEEDS WITH CONTENT-AND-STRUCTURE

Now that we have seen what properties can *in principle* be expressed by ignorant and semi-ignorant users in their respective hybrid query languages, we take a closer look at what users *actually* express in the semi-ignorant query language used at INEX 2004. We will see that many of the queries submitted can actually be expressed in the ignorant sublanguage. But let's not run ahead of ourselves.

4.1 The INEX Query Format

At INEX, two types of topics are used. Content-Only (CO) topics and Content-And-Structure (CAS) topics. All topics contain the same three fields as traditional IR topics [6, 11]: title, description and narrative. The description and narrative describe the information need in natural language. The difference between the CO and CAS topics lies in the topic title. In the case of the CO topics, the title describes the information need using a small list of keywords. In the case of CAS topics, the title describes the information need using (a flavor of) XPath extended with the `about` function. At INEX 2003, full XPath was allowed, and at INEX 2004 descendant positive XPath (i.e., the restricted fragment for semi-ignorant users) is used [29, 30]. Below we analyze the title part of the CAS topics.

4.2 INEX 2004 Queries

The specific instructions for topic development at INEX 2004 [28] stated that CAS queries

- should use only descendant axis (i.e., `//`),
- should use only boolean `and` and `or`,
- should contain at least one `about` statement, and
- the rightmost filter should be an `about` statement.

The resulting language is called NEXI (Narrowed Extended XPath I) [30]. We consider the set of 34 CAS topics (version 2004-7) with topic numbers 127–147, and 149–161.

4.2.1 Knowledge of the document structure

Because of the restrictions just listed, the NEXI language is a proper subset of the *semi-ignorant user* language discussed in Section 3. We can break down the 34 NEXI topics based on the two types of users identified in Subsection 3.1.

Ignorant Users. They know only (some of) the tag names in the collection, but are ignorant of the structure of the documents. In total there are 11 topics that reflect this type of user.[2] The topic numbers are 128, 134, 136, 141–143, 145, 151, 152, 159, and 160.

Semi-Ignorant Users. These users have some idea of the hierarchical structure of the documents. I.e., they know (some of) the tag names in the collection, and (some of) the legitimate nesting of tags. There are 23 topics that reflect this type of user. The topic numbers are: 127, 129–133, 135, 137–139, 140, 144, 146, 147, 149–150, 153–158, 161.

[2]We ignore the restriction to only returning XML elements within articles [19]. I.e., most queries start with `//article` to reflect this constraint. Only three queries do not start with this prefix, however these queries are prefixed with either ⟨sec⟩ or ⟨abs⟩ tags that only occur in the context of an ⟨article⟩ tag anyway.

Element	Frequency	Percentage
sec	16	47.06%
article	5	14.71%
p	4	11.76%
*	2	5.88%
abs	2	5.88%
bb	1	2.94%
bdy	1	2.94%
bib	1	2.94%
fig	1	2.94%
vt	1	2.94%

Table 1: Frequency of requested elements in the 34 CAS topics of INEX 2004.

Even with the explicit instructions to construct content-and-structure queries, no less than one-third of the resulting queries can be expressed in the very restricted *ignorant user language* introduced in Section 3.

4.2.2 Requested elements

One of the main advantages of using CAS queries is that they allow the user to specify the types of elements that should be returned. Table 1 lists the elements resulting from the granularity constraints in the 34 CAS topics.

4.3 The INEX 2004 Assessments

At INEX, relevance is assessed on the basis of the narrative describing the underlying information need. As Kazai et al. [16, p.237] put it:

> CAS queries are topic statements, which contain explicit references to the XML structure, and explicitly specify the contexts of the user's interest (e.g. target elements) and/or the contexts of certain search concepts (e.g. containment conditions). [...] Although users may think they have a clear idea of the structural properties of the collection, there are likely to be aspects to which they are unaware. The idea [...] is to allow the evaluation of XML retrieval systems [...] where not only the content conditions within a user query are treated with uncertainty but also the expressed structural conditions. [...] The path specifications should therefore be considered hints as to where to look.

In the spirit of textual IR, the granularity constraint is not strictly enforced, but merely regarded as a retrieval hint. Hence, it is of interest to look at the tag-names of elements that are judged relevant for the respective topics.

4.3.1 Elements judged relevant

We use version 3.0 of the assessments, containing judgments for the 26 topics numbered 127–137, 139–145, 149–153, and 155–157. Moreover, we focus on elements rated as highly exhaustive and highly specific—also called strict or (3,3) assessments. For the 4 topics numbered 133, 140, 143, and 144, there are no elements judged as highly exhaustive and highly specific. Table 2 lists the frequencies of element types judged relevant for the remaining 22 CAS topics. We collapse the tag equivalences for sections and paragraphs, as defined in [28].

Element	Frequency	Percentage
p+	854	31.41%
vt	747	27.47%
sec+	262	9.64%
au	110	4.05%
bb	104	3.82%
fnm	104	3.82%
st	90	3.31%
article	73	2.68%
fig	53	1.95%
it	37	1.36%
bdy	36	1.32%
ref	34	1.25%
scp	32	1.18%
atl	23	0.85%
abs	13	0.48%
fm	11	0.40%
b	10	0.37%

Table 2: Frequency of elements judged relevant for all assess CAS topics at INEX 2004. We only show tag names occurring at least 10 times.

	article	sec+	p+	abs	vt
article (2)	10.8%	1.3%	1.6%	–	–
sec (10)	3.3%	27.7%	24.7%	0.9%	0.4%
p (4)	4.0%	26.0%	48.0%	–	–
abs (2)	16.0%	–	24.0%	24.0%	–
vt (1)	–	–	44.0%	–	52.0%

Table 3: Frequency of relevant elements (columns) for topics with particular granularity constraint (rows). The number of aggregated queries is indicated between brackets.

4.3.2 Cross product

We also investigate how often the element that is judged relevant actually has the tagname specified by the granularity constraint. Consider Table 3; the rows show the tag-names of elements resulting from the granularity constraints, and the columns show the tag-names of elements judged relevant.[3] It is clear from the table that the granularity constraint can indeed be considered as a retrieval *hint*. Although it is far from strictly enforced, there seems to be a preference for the type of XML elements satisfying it.

4.4 How Structure is Used

To further our understanding of the role of structure in content-and-structure topics, we break down the set of topics by increasing complexity. We define four categories:

Restricted Search. This category has topics in which structure is only used as a granularity constraint. The topic is an ordinary content-only topic, where the search is restricted to particular XML elements. There is a filter on the target element having no nested path constraint. A typical example of such a topic is to restrict the search to sections, this may look like:

```
//sec[about(., ‘‘xxx’’)].
```

[3]Note, especially for granularity constraints abs and vt, that we do not distinguish between paragraphs appearing within or outside the element matching the granularity constraint.

This category has 5 topics: 127, 136, 142, 143, and 152.

Contextual Content Information. This category is similar to the *Restricted Search* category, but now there may be a content restriction on the environment in which the requested element occurs. A typical example of such a topic is one asking for sections from articles with a content restriction on the abstract, this may look like:

```
//article[about(.//abs, ‘‘xxx’’)]//sec[about(., ‘‘yyy’’)].
```

The category contains 16 topics: 128, 129, 130, 131, 132, 134, 135, 137, 138, 141, 144, 145, 151, 158, 159, and 160.

Search Hints. This category contains topics with a complex filter in which a nested path occurs, but the element targeted by the nested paths resides inside the requested element. I.e., the user provides a particular retrieval cue to the system. An example of such a topic may be, when interested in sections on a topic, to tell the system to look for certain terms to appear in a theorem environment, this may look like:

```
//sec[about(., ‘‘xxx’’) and about(.//thm, ‘‘yyy’’)].
```

There are 2 topics in this category, numbered 147, and 153.

Search Hints in Context. The fourth and last category deals with topics with a nested path that targets elements that are disjoint from the requested element. Here, the user is really exploiting her knowledge of the structure of the documents, and conditions the retrieval of elements on the content found along other paths in the document tree. I.e., the condition is evaluated against parts of the text that are not being returned to the user as a result. E.g., one might be looking for sections, in papers authored by someone. This may look like:

```
//article[about(.//fm//au, ‘‘xxx’’)]//bdy//sec[about(.,
    ‘‘yyy’’)].
```

There are 11 topics in this category, numbered 133, 139, 140, 146, 149, 150, 154, 155, 156, 157, and 161.

Carmel et al. [4, 5] proposed XML fragments as another, simple alternative to XPath for content and structure queries. Using the intuitive query-by-example underlying XML Fragments, only the *Restricted Search* and *Search Hint* categories can be expressed. For capturing queries in the other categories, a syntactic device is introduced [4].

5. HOW DOES STRUCTURE HELP?

If users are aware of the structure of documents in a collection, they can query the collection by means of constraints on both the content and the structure of desired XML elements, giving rise to the following

> **CAS Hypothesis** Hybrid content-and-structure queries are more expressive than ordinary natural language queries, and this will lead to better retrieval performance.

While the CAS hypothesis has great intuitive appeal, it is also rather vague and underspecified. In what sense can the structural part help to improve retrieval performance?

The use of structure in queries has been studied extensively in the literature; prominent examples include booleans, proximity and phrase operators. In early publications, the usage of phrases and proximity operators—as well as a careful usage of boolean operators—showed improved retrieval results [7, 8, 12, 13, 17], but rarely anything substantial. As retrieval models became more advanced, the usage of query operators was questioned. E.g., Mitra et al. [22] conclude that when using a good ranking algorithm, phrases have no effect on high precision retrieval (and sometimes a negative effect due to topic drift). Rasolofo and Savoy [25] combine term-proximity scoring heuristics with an Okapi model, obtaining 3%–8% improvements for Precision@5/10/20, with hardly observable impact on the MAP scores. Mishne and de Rijke [21] found that even on top of a good basic ranking scheme for web retrieval, phrases and proximity terms may bring improvements in retrieval effectiveness, both for MAP and high precision measures.

Where does this leave us with our content-and-structure queries? First, it is interesting to analyze how the expressiveness of content-and-structure queries is put to use, and, in particular, in what sense this may lead to better retrieval performance. To address the issue, we return to the four topic categories from Section 4. For each category, one would expect the structural constraints to have a precision-enhancing effect, either by specifying or constraining the granularity of the elements being sought (as in the *Restricted Search* category), or by constraining the environment in which the results being sought appear (as in the *Contextual Content Information* and *Search Hints* categories, or by imposing “non-local” structural and content constraints (as with the *Search Hints in Context* category). Overall, then, our expectation is that structural aspects of the query improve precision. Although this may reduce fall-out, it will also reduce recall. So, it is not clear what we may expect for CAS queries in terms of mean average precision. If we expect CAS queries to function as a precision device, then we should put more emphasis on measures that reflect this.

There is some experimental evidence that confirms the precision enhancing nature of structural constraints. As explained above, at INEX 2004 topics were assessed while treating the structural constraints as hints. Sigurbjörnsson et al. [27] experimented with a content-only based approach [15] vs. a content-and-structure based approach [26]: whereas the CO based approach resulted in superior MAP, the strict CAS-based approach resulted in improved early precision scores (MRR, Precision@10, etc). These findings suggest that the retrieval task underlying content-and-structure querying is no different from the ordinary natural language query retrieval task: they may be used as different ways of articulating the same information need.

6. EVALUATION LESSONS

The user-oriented upshot of the previous sections is that users use CAS in different, and often shallow or restricted ways, most likely as a precision enhancing device. How do these findings inform us about the *evaluation* of systems that handle content-and-structure queries? We discuss several aspects. We will first describe at a high level how we expect users to interact with a retrieval engine that supports content-and-structure querying.

Let’s imagine the mental process the user goes through when interacting with a retrieval engine. First, she starts with an abstract, informal information need. Then she needs to articulate her information need more formally. Naturally she would first use natural language and formulate her in-

formation need as a short list of keywords. If our user has additional knowledge about the types of documents or elements that would satisfy her information need, in particular about the document structure, she may consider rendering her information need in a structured language. The resulting query is, presumably, a more precise description of the original information need. However, the underlying information need has not changed. If our user doesn't have such additional knowledge, she simply won't add structure to her query. Either way, she will take her formulation of the information need, structured or not, to the retrieval engine. Independent of whether the query has structure, the task of the retrieval engine is to answer the user's information need. In the end, the success of the search process depends solely on whether the user's information need was satisfied.

Against this background, our findings from earlier sections have some clear implications for the evaluation of content-and-structure querying. Let's consider the various stages of the evaluation process, starting with topic development process. The topic formulation process could start with writing a detailed natural language description of the information need. This will imitate the formation of a mental image of the information need. The next step would be to formulate the need as a list of keywords. Then, in case the user/topic creator has the appropriate knowledge, and the collection supports it, the user/topic creator can also formulate her information need using a mixture of content and structure requirements. What we end up with, then, is one set of topics, all of which have a natural language title. However, a subset of the topics will also have a content-and-structure title. Of course, all topics have a narrative which verbosely describes the information need.[4]

The assessments will not differ from traditional IR assessments. The narrative will be authoritative when judging results. For topics whose information need is expressed in different query formats (i.e., with and without structural constraints), we get one query assessed for free.

In the evaluation phase, the added value of having information needs both expressed in natural language and in a structured language allows us to directly (and only for appropriate information needs) measure whether structural constraints can indeed be used to enhance keyword based queries. Systematic experiments with multiple ways of expressing the same information need, would help make progress on research questions such as our *CAS Hypothesis* in Section 5."

Turning to metrics now, as we pointed out, we expect structural hints to be a precision enhancing device. The evaluation should try to answer whether that is actually the case by comparing systems not solely based on MAP but also on initial/early precision metrics such as MRR, Precision@10, etc.

It is important to realize that a content-and-structure query depends on both the information need and the structure of the document collection. Whereas natural language queries usually depend solely on the information need, structured queries also crucially depend on knowledge of the types of elements that are relevant, i.e., what tags-names they have, or how these are nested. This implies that, in a sense, structure is never an inherent part of an information need itself; at most, it is part of a formal query that offers one (out of a many) way of expressing the information need. This does not mean that the notions of structure and information need are independent. The structured part of the query may capture a part of the information almost literally. As an example of a situation where a structured expression can be very helpful, consider a user who wants to look at vitae of machine learning students. The natural language expression of this information need may look like

```
vitae machine learning student.
```

In terms of the markup of the INEX collection, which has a special tag for vitae, and the NEXI query language, this query may be better expressed as

```
//vt[about(., machine learning student)].
```

Note, however, that even if we have multiple expressions of an information need, the need itself stays the same. Structural constraints do not alter the original information need: they merely express the need differently, more formally. And it is important to understand how useful or helpful they are, and to set up the appropriate experiments that will tell us exactly that.

7. CONCLUSIONS

Document-centric XML is a mixture of text and structure. With the increased availability of document-centric XML content, we require query facilities in which both structural constraints and constraints on the free text of the documents can be expressed. This has generated considerable interest in the IR community, and has lead to the launch of evaluation efforts tailored for XML documents. One of the driving and long-standing research questions is: How does the increased expressiveness of query languages tailor-made for querying XML documents help users to better, and more effectively, express their information needs? And closely related to this: How should we evaluate systems that enable users to express their information needs using both content and structural constraints?

We addressed these research questions from two angles: what requirements can *in principle* be expressed in query languages for document-centric XML documents? And, how do users *actually* use such languages? For the former, we gave mathematical characterizations of two query languages, in terms of suitable variations on the notion of bisimulation. To address the latter, we provided a detailed examination of the topics formulated in the NEXI query language as part of the 2004 edition of the INEX XML retrieval initiative. Our main findings are as follows. First, while structure is used in varying degrees of complexity, over half of the queries can be expressed in the very restrictive *ignorant user* language. Second, structure is used as a search hint, and not a search requirement, when judged against the underlying information need. Third, the use of structure in queries functions as a precision device. Fourth, the underlying retrieval task of content-and-structure querying is no different from the ordinary natural language query retrieval task. From those findings we derive a number of recommendations for the evaluation of systems that cater for content-and-structure queries: First, if we expect content-and-structure queries to function as a precision device, we should also look at measures that reflect this. Second, if the underlying retrieval

[4]As an aside, this procedure should have a positive effect on the pool quality, as we can pool together retrieval results that are derived from essentially different representations of the same information need.

task is the same for content-only and content-and-structure topics, we could use a single topic set with title fields for both a natural language query as well as a structured query. Third, if we introduce an additional title field for a content-and-structure query, it should be optional, so that users will formulate a structured query only in case the underlying information need naturally gives rise to it.

8. ACKNOWLEDGMENTS

Thank you to all the participants of the Topic Format Working Groups at INEX 2002–2004. This research was supported, in part, by the Netherlands Organization for Scientific Research (NWO) under project numbers 017.001.-190, 220-80-001, 264-70-050, 354-20-005, 612-13-001, 612.-000.106, 612.000.207, 612.066.302, and 612.069.006.

REFERENCES

[1] S. Amer-Yahia, L. V. S. Lakshmanan, and S. Pandit. FleXPath: flexible structure and full-text querying for XML. In *SIGMOD'04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 83–94. ACM Press, 2004.

[2] M. Benedikt, W. Fan, and G. Kuper. Structural properties of XPath fragments. In *Proc. ICDT*, 2003.

[3] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[4] 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.

[5] D. Carmel, Y. S. Maarek, Y. Mass, N. Efraty, and G. M. Landau. An extension of the vector space model for querying XML documents via XML fragments. In R. Baeza-Yates, N. Fuhr, and Y. S. Maarek, editors, *Proceedings SIGIR 2002 Workshop on XML and Information Retrieval*, pages 14–25, 2002.

[6] CLEF. Cross-Language Evaluation Forum, 2004. `http://www.clef-campaign.org`.

[7] W. Croft, H. Turtle, and D. Lewis. The use of phrases and structured queries in information retrieval. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 32–45. ACM Press, 1991. ISBN 0-89791-448-1.

[8] J. Fagan. Experiments in automatic phrase indexing for document retrieval: A comparison of syntactic and non-syntactic methods. Technical report, Cornell University, 1987.

[9] 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.

[10] N. Fuhr and K. Großjohann. XIRQL: An XML query language based on information retrieval concepts. *ACM Transactions on Information Systems*, 22:313–356, 2004.

[11] D. Harman. Overview of the first Text REtrieval Conference (TREC-1). In *Proc. TREC-1*, 1993.

[12] D. Hawking and P. Thistlewaite. Relevance weighting using distance between term occurrences. Technical Report TR-CS-96-08, Department of Computer Science, Australian National University, 1996.

[13] D. Hull, G. Grefenstette, B. Schultze, E. Gaussier, H. Schutze, and J. O. Pedersen. Xerox TREC-5 Site Report: Routing, Filtering, NLP, and Spanish Tracks. In *Proceedings TREC-5*, pages 167–180, 1997.

[14] INEX. INitiative for the Evaluation of XML Retrieval, 2004. `http://inex.is.informatik.uni-duisburg.de:2004/`.

[15] 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.

[16] G. Kazai, M. Lalmas, and B. Piwowarski. INEX 2004 relevance assessment guide. In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors, *INEX 2004 Workshop Pre-Proceedings*, pages 241–248, 2004.

[17] E. Keen. Term position ranking: some new test results. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 66–76. ACM Press, 1992. ISBN 0-89791-523-2.

[18] N. Kurtonina and M. de Rijke. Expressiveness of concept expressions in first-order description logics. *Artificial Intelligence*, 107(2):303–333, 1999.

[19] M. Lalmas and G. Kazai. INEX 2004 retrieval task and result submission specification. In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors, *INEX 2004 Workshop Pre-Proceedings*, pages 237–240, 2004.

[20] W. May. Information extraction and integration with FLORID: The MONDIAL case study. Technical report, Universität Freiburg, Institut für Informatik, 1999.

[21] G. Mishne and M. de Rijke. Boosting web retrieval through query operations. In *Proceedings ECIR 2005*, 2005.

[22] M. Mitra, C. Buckley, A. Singhal, and C. Cardie. An analysis of statistical and syntactic phrases. In *Proceedings of RIAO-97*, 1997.

[23] G. Navarro and R. Baeza-Yates. A language for queries on structure and contents of textual databases. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 93–101. ACM Press, New York NY, USA, 1995.

[24] R. A. O'Keefe and A. Trotman. The simplest query language that could possibly work. In *Proceedings of the 2nd INEX Workshop*, 2004.

[25] Y. Rasolofo and J. Savoy. Term proximity scoring for keyword-based retrieval systems. In *Proceedings 25th European Conference on IR Research (ECIR 2003)*, pages 207–218, 2003.

[26] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. Processing content-oriented XPath queries. In *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management (CIKM 2004)*, pages 371–380, 2004.

[27] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. The University of Amsterdam at INEX 2004. In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors, *INEX 2004 Workshop Proceedings*, pages 104–109, 2004.

[28] B. Sigurbjörnsson, B. Larsen, M. Lalmas, and S. Maalik. INEX04 guidelines for topic development. In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors, *INEX 2004 Workshop Pre-Proceedings*, pages 212–218, 2004.

[29] B. Sigurbjörnsson and A. Trotman. Queries, INEX 2003 working group report. In *Proceedings of the 2nd INEX Workshop*, 2004.

[30] A. Trotman and B. Sigurbjörnsson. Narrowed Extended XPath I (NEXI). In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors, *INEX 2004 Workshop Pre-Proceedings*, pages 219–236, 2004.

[31] V. Vianu. A Web odyssey: from Codd to XML. In *Proc. PODS*, pages 1–15. ACM Press, 2001. ISBN 1-58113-361-8.

[32] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, 1994.

Notes on what to measure in INEX

Gabriella Kazai
Queen Mary University of London
Mile End Road
London, UK
gabs@dcs.qmul.ac.uk

Mounia Lalmas
Queen Mary University of London
Mile End Road
London, UK
mounia@dcs.qmul.ac.uk

ABSTRACT
This paper looks at a number of issues regarding the evaluation of XML retrieval. It aims to identify what the requirements on a measure of XML retrieval effectiveness are and how the actual evaluation methodology and aspects such as the relevance dimensions and the assessment procedure affect the evaluation. We examine various current and proposed metrics, how they fit the requirements and aim to give an explanation of what exactly they measure. A question we are attempting to address is: "Is there a single good measure of retrieval effectiveness for XML retrieval?".

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

General Terms
Performance, Measurement

Keywords
XML retrieval, INEX, evaluation metrics

1. INTRODUCTION
Since its launch in 2002, INEX (INitiative for the Evaluation of XML Retrieval) has been challenged by the issue of how to measure an XML information retrieval (IR) system's effectiveness. Due to the fact that most underlying assumptions that traditional IR metrics are based upon no longer hold in an XML IR setting [6], INEX has been investigating various adaptations of established measures as well as newly proposed metrics.

Currently there are five metrics under consideration to be used as the official metric of INEX 2005. One issue with having a range of available metrics is that unless we are clear about what exactly they measure, their incorrect use can lead to confusion regarding the result of the evaluation. Although pair-wise comparisons of some of the metrics now exist in the literature [10, 17], we are still largely in the dark as to how these different measures relate to each other or how they differ from each other, or, in fact, how well they suit the evaluation task.

INEX Workshop on Element Retrieval Methodology, Glasgow, UK, July 30, 2005

In this paper, we look at various issues regarding the evaluation: what should we expect from a measure, how do the relevance dimensions and the assessment procedure affect the evaluation and what the current metrics measure.

2. WHAT TO MEASURE
The main criterion of any evaluation measure is that it should be able to rank systems according to how well they satisfy a user's information need, given a retrieval task and a model of user behaviour.

2.1 Retrieval task
In INEX, the retrieval task is given as the ad-hoc retrieval of XML documents. As in traditional IR, the INEX task of ad-hoc retrieval is considered as a simulation of how a library might be used: a static set of documents being searched using a new set of topics. However, the similarity ends there.

In traditional IR, the library consists of documents, representing well-defined units of retrieval, where the relevance of one document to the query is (considered) independent from the relevance of other documents to the query. The user's information need is typically expressed in the form of a natural language statement or simply as a set of keywords. Given this, the task of an IR search engine is to return to the user, in response to his/her query, as many relevant documents and as few irrelevant documents as possible. The output is usually presented to the user as a ranked list of documents, ordered by presumed relevance to the query.

Established measures, such as recall and precision graphs, provide suitable and intuitive mechanisms for evaluating the effectiveness of IR systems based on the above retrieval task and model of user interaction. The atomic retrieval unit of a document and the binary relevance assumption allows for the simple counting of the number of relevant and the number of retrieved documents, which forms the basis of recall/precision measures. The ranking is considered by taking counts at various recall levels.

In an XML IR setting, the library consists of XML documents composed of different granularity nested XML elements, each of which represents a valid unit of retrieval, where the relevance of one component may be dependent on the relevance of other structurally related components. Furthermore, the user's query may contain structural constraints in addition to the typical content conditions. These structural constraints may then be interpreted by an XML

IR system as strict conditions that must be met by relevant elements or as vague conditions that can be considered only as hints or clues as to where relevant information may be found. The decision really comes down to the question of how much we trust users' understanding of the searched collection's structure as well as their abilities in expressing complex queries.

The general task of an XML IR engine has been defined in INEX as the task of returning, instead of whole documents, those document components (XML elements) that are most specific and exhaustive to the user's information need [15, 14]. This general task definition is then extended for the various CAS sub-tasks to take into account that user satisfaction is also dependent on the structural conditions being met (strictly or vaguely). To simplify our study, for the rest of the paper we concentrate on the CO tasks.

An issue with the above general task definition is that it leaves the concept of "most specific and exhaustive" somewhat unspecified (or rather specified only within the quantisation functions, which currently do not form part of the task description). In addition, a common misinterpretation of "most specific and exhaustive" is to equate it to "highly specific and highly exhaustive" (i.e. $(e, s) = (3, 3)$). However, the term "most" is understood to refer to the highest available combined exhaustivity and specificity score of nodes in a given XML tree. For example, it may be that amongst all possible retrievable components in an XML document (article), even the most exhaustive node is only marginally exhaustive ($e = 1$) if the topic of request is only mentioned. Similarly, it may well be that even the most specific node contains some irrelevant information (e.g. $s = 2$). When the combination of the two dimensions is considered, additional criteria is required to decide if the most specific and exhaustive elements of a tree should be, for example, two $(e, s) = (2, 3)$ paragraph elements or their container $(e, s) = (3, 2)$ section element.

The output of XML IR systems, up until the time of writing this, has been assumed to be a ranked list of XML elements, ordered by their presumed relevance to the query. Other forms of non-linear result presentations, e.g. where related results are clustered based on structural relationships, have so far been ignored due to the added complexity in their evaluation. The INEX 2005 task guidelines [14], developed in the meantime, provide a welcomed step in this direction, although - in the opinion of the first author - they require further development (more on this in Section 2.3).

2.2 User behaviour

The definition of the general ad-hoc retrieval task in the previous section is still rather vague, and one that requires further clarification. For example, what exactly is meant by returning XML elements to users? Will users have access to the full text of a returned element and its sub-nodes? What about access to the element's context? Will users need to browse in order to access related components or will the system show result elements, for example, as highlighted text fragments within their larger context element? What can be assumed of users' interaction with the system? All these factors affect the assumed user model which then impacts on the evaluation.

A user of a traditional IR system is typically associated with a simple model for interacting with the system. He/she is assumed to examine the returned ranked list in a linear fashion, moving from the top of the list down, either until the end of the list is reached, or until the point where his/her information need has been satisfied or where the user gives up. Each examined document is assumed to require approximately the same amount of effort from the user.

The user of an XML IR system is currently assumed to follow the same routine and work through the returned ranked list from top to bottom, with similar stopping options. The required effort to consult a result element, however, can no longer be assumed to be equal, but should rather be given as some function of element size or required reading time.

When users of an XML IR system access a result element, they may then have access, in one form or another (e.g. browsing or scrolling), to the element's structurally related nodes and/or context (where, depending on the user interface, the cost of this access may differ in different situations). This motivates the need to consider so-called near-misses, elements from where users can access desired relevant content, within the evaluation. For example, a section containing the sought-after relevant paragraph, a list item within the paragraph, or a neighbouring paragraph or section may all be considered as near-misses. A near-miss may itself be relevant or irrelevant to the user's query. Assuming that such near-misses may be useful for a user, as it gives him/her access to otherwise lost relevant information, the idea is then to allow systems to pick up partial scores for finding such elements.

In addition, due to the possible overlap of result elements (e.g. returned nested elements), it is argued that a further assumption is needed in INEX, according to which redundant relevant fragments are to be considered of no further value to the user. For example, once seen, a relevant paragraph may be of no interest to the user if it is again returned as part of its container section. The need for making this behaviour an explicit assumption has only been highlighted recently in [3, 10]. In [10], it was shown that unless an evaluation metric that explicitly addresses this issue is employed, unfair advantage can be gained by systems that exploit this phenomena of the INEX recall-base over systems that actually put effort into not to inundate users with such redundancy. This process of deliberately returning overlapping elements to increase effectiveness results has since been popularly named as "milking" and has been the centre of some debate in INEX.

Most of us agree that returning overlapping results contradicts the intuition about the retrieval task, which aims to decrease the user effort required in finding relevant information, and that it can lead to user disorientation when such related redundant components are dotted around at different ranks in the output list. This was also indicated in the experiments conducted by the INEX 2004 interactive track [22]. However, we may equally argue that overlap - from a system evaluation point of view - should not be seen as an issue since systems can be assumed to be able to deal with it when presenting their results to users. For example, systems may remove overlapping nodes via some filtering strategy or

cluster them together, and so on. Therefore, overlap should be allowed in result lists when a system-oriented evaluation is applied. This said, a crucial (implicit) assumption of this argument is that overlap, while allowed, should not represent a potential gain factor to be exploited. This means that systems should not be penalised for not retrieving overlapping nodes! For example, a system that retrieves all relevant nodes on a path (e.g. `article[1]`, `bdy[1]`, `sec[6]` and `p[1]` in Figure 1), ranking the highest scoring first (e.g. `sec[6]`), should not be ranked better by the evaluation than another system that only returns the highest scoring node from the same path (e.g. `sec[6]`). In conclusion, *given a suitable metric*, systems would be free to follow a retrieval strategy based on "milking", but such a strategy would not present an advantage, but may in fact prove unbeneficial as it may result in the output list being filled with elements of no further value while pushing other non-overlapping relevant nodes down the list.

As a result, if returning overlapping results does not lead to a sensible retrieval strategy, then systems will be forced to make decisions as to which element(s) to retrieve from an arbitrary tree of XML elements, which is arguably the presumed aim of XML IR. As mentioned before, the retrieval task implicitly relies on a set of user preferences, modeled within the quantisation functions. These preferences dictate which elements systems should return to the user from a given XML tree. For example, the generalised quantisation function describes a user who would prefer more exhaustive components despite the additional effort needed to be spent on locating the relevant information within. Based on these set of preferences, a system would need to locate those elements in an XML tree that are more exhaustive than any of their structurally related nodes, where from two nodes with the same exhaustivity the more specific one is preferred.

2.3 Matching user types to tasks

A problem with the current setup of the general INEX retrieval task and the various user models represented by the quantisation functions is that in the first instance the task is not explicitly motivated by a given user model and secondly that different systems may have been tuned to different user models, but were all evaluated under the general CO task umbrella and using all quantisations. While this may provide an indication of how well systems do in general (in trying to satisfy all types of users), an appropriate matching of evaluation criteria and tasks is still needed. To this end, what is required is to define specific retrieval tasks that motivate certain user behaviours. For example, the task of highlighting highly specific relevant text fragments may reflect a user who prefers more specific elements and who may have access to the context of the highlighted text fragments. In line with this, during the INEX 2004 workshop (see http://inex.is.informatik.uni-duisburg.de:2004/presentations/metrics-wg.ppt), the following system task has been put forward for INEX 2005:

- Find the most specific elements (in each path), i.e., those elements with the highest ratio of relevant to irrelevant information. These elements are considered independent (i.e., non-overlapping), of equal quality, and it does not matter if they are from the same or different documents.

An argument that supports the selection of this task as "the" main task in INEX is that it requires search engines to pinpoint the exact location of relevant texts (and hence it is not enough to just go for a 'safe' option and return large container units). We see this as one of the main driving forces for XML IR in the first place: XML IR systems should aim to present users with more focused material, and thus reduce users' efforts in locating sought-after information. In other words, systems should return components that contain as much relevant information and as little irrelevant information as possible.

Given this retrieval task, a suitable evaluation measure should be able to rank systems according to how well they are able to locate XML elements that contain as much relevant information and as little irrelevant information as possible.

In addition to the above system task, a number of user tasks have been outlined at the workshop:

- Find the most specific elements in a path
- Find as much relevant content as possible
- Find as many relevant elements as possible

Here, the first task may be considered as an extension of the system task, where additional aspects of the user's interaction with the retrieval system may be included, e.g. browsing to structurally related elements. The second and third tasks are a bit harder to interpret and seem to be more motivated from a system-oriented point of view, whereby systems are required to return all reference elements that form the full recall-base (including all overlapping nodes).

Based (loosely) on the above task proposals, INEX 2005 defined a number of specific retrieval strategies to be investigated: "focused", "thorough" and "fetch and browse" strategies. These strategies build on assumed user behaviours that take into account how the results may actually be presented and provide explicit guidelines for search engines on how to deal with issues such as overlap. For example, the focused strategy aims to remove overlap and can be associated with a user interface where most specific elements may be highlighted for the user.

Although these sub-tasks go some way to clarify what actual output is expected of an XML IR system, they still leave a lot of questions open mainly due to the problem that we are unsure about what real users of an XML digital library would want returned to them. As a result, in the opinion of the first author, the sub-task definitions still remain open to individual interpretation, which is bound to lead to confusion and later on to questions regarding the appropriateness of the adopted evaluation metrics.

In an effort to correct this, the following modifications are suggested with respect to the focused task: "This strategy should return to the user those largest XML elements that contain *only relevant* (or minimal irrelevant[1]) information. A reason to specify 'largest' in the definition is that in case of

[1]E.g. if the most specific node on a path is $s = 2$ or $s = 1$.

a completely relevant section (e.g. $s = 3$), the section should be returned instead of its individual paragraphs (which will also have $s = 3$ since no irrelevant information is contained in the section element and consequently in any of its sub-nodes). More formally, the task is to return, given an arbitrary tree of relevant XML elements, the most specific non-overlapping relevant elements, where relevant simply means having any level of exhaustivity ($e > 0$). From two nodes with the same specificity the one with higher exhaustivity should be retrieved. In the case where two nodes on the same path are equally specific and exhaustive, the ascendant element should be returned. The output should be presented to the user as a ranked list of XML elements, ranked by specificity first and then by exhaustivity."

The thorough strategy, which may be motivated by the idea of using it as a catch-all for possible different retrieval strategies, may be defined as a task to "find all relevant elements, where a relevant element is one with $e > 0$. The output is assumed to be a ranked list of XML elements, ranked by combined exhaustivity and specificity according to a chosen quantisation function."[2]

The fetch and browse strategy is also felt to be rather vague. While the basic idea of the fetch phase is clear, the browse phase will need further clarification. Although the authors do not actually agree on this point, we would like to suggest as discussion point the following redefinition of this task into two separate tasks: a fetch and highlight strategy and a fetch and browse strategy.

The aim of the former strategy would be to first identify relevant articles (the fetching phase), and then to identify the most specific relevant elements within the fetched articles (the highlighting phase). In the fetching phase, articles should be ranked according to how exhaustive and specific they are, where the relative value of the combined exhaustivity and specificity would be given by a chosen quantisation function. For the highlighting phase, the ranking of XML elements within an article should be done according to the focused retrieval strategy. The assumed output is a ranked list of articles, which are then viewed by the user as flat text files, where the most specific relevant elements are highlighted.

Within the fetch and browse strategy, as with the fetch and highlight strategy, the aim of the fetching phase is to retrieve relevant articles, ranked by exhaustivity and specificity (based on a chosen quantisation function). For the browsing phase, the ranking of XML elements within an article should be done according to the thorough retrieval strategy. The assumed output is again a ranked list of articles, but on viewing the user is assumed to interact with a ranked list of XML elements from the article.

Alternative tasks may also consider the retrieval of "best elements", which involves finding the preferable units of retrieval (given a specific user interface). We, however, believe that such a task requires, as its precondition, knowledge of the locations of the most specific elements. Strategies for deciding which elements would be best to return to the user will then further depend on assumptions about the user's preferences and browsing behaviour as well as assumption about how the results are presented to the user. For example, best elements may be best hub nodes for a user of a hyperlinked environment who is happy to browse in search for relevant content. However, if the results are presented to the user as highlighted text fragments within a document unit (e.g. article), then best elements may well be the same nodes as the most specific elements.

A further issue with the "finding the best elements" task is that it may require additional assessments, whereby given a set of relevant nodes in an XML tree and a specific user interface, users need to identify which elements they would want to be returned by a search system [12]. Note that we would not recommended to try to obtain assessments - directly within the relevance assessment procedure - with the "best elements" task in mind as it is ultimately a much more complex notion than relevance. Different people will have widely varied ideas as to what should be a best element to return (even if the user interface is fixed), which is likely to have an impact on the quality of the assessments.

Nevertheless, the best element task is one that is of particular interest to us and we would be keen to support its integration into INEX. We envision the interactive track as probably the best venue for starting experiments to investigate this task and how best to derive assessments for it. Some initial results on a different test data can be found in [19, 20].

3. RELEVANCE

3.1 Multiple dimensions and degrees

As mentioned before, the ordering of the results in the output list is according to presumed relevance. In traditional IR experiments, this output is then compared against the set of relevant documents identified by human assessors (or its subsets at different recalls). Since relevance assessments are typically given in the form of binary decisions, e.g. relevant or not, simple counting mechanisms can be employed by the evaluation measures (i.e. precision and recall).

In INEX, relevance represents a more complex notion with two separate identified aspects: exhaustivity and specificity. Both these aspects influence the overall relevance of an XML element: the more exhaustive and more specific an element, the more it is desired by the user. Exhaustivity reflects how exhaustively a document component discusses the topic of request (and hence relates to the amount of relevant information contained within the element), while specificity reflects how focused the component is on the topic of request, i.e. discusses no other, irrelevant topics (and hence relates to the amount of irrelevant information contained within the element). These two aspects have been separated into two relevance dimensions for better control. Although there have been arguments against this separation, it was decided that this solution would provide a more stable measure of relevance than if assessors were asked to rate elements on a single scale. This is because on a single scale an element may be judged, for example, marginally relevant if it contained only relevant information, but this information was not very exhaustive; and also if it was exhaustive, but the element also contained a lot of irrelevant information. Judges

[2]The current definition is already along these lines, but the phrasing of the task may be slightly misleading.

are also likely to place varying emphasis on these two aspects when assign a single relevance value.

This argument is supported by our findings from building a small test collection from Shakespeare plays marked up in XML. There, we employed binary relevance assessments, which were derived using a highlighting procedure (assessors marked relevant text fragments with a yellow marker), where each topic was assessed by multiple assessors. We found that different people highlighted widely different sized text fragments as being relevant to the same query. Some of the assessors highlighted very specific relevant sentences only, while others highlighted complete sections [12]. This suggests that, when considering relevance, different judges placed varying degrees of importance on the exhaustivity or specificity aspects and highlighted text segments according to a relative rating that they felt was appropriate in a given situation and at a given time. In addition, text fragments that were not strictly relevant, but provided contextual information may have also been highlighted by some of the judges.

One advantage of a single scale relevance, however, is that it implicitly combines exhaustivity and specificity (and probably other aspects too), which closer reflects the user's true preferences, rather than being modeled afterwards using quantisation functions.

In INEX, in addition to the two dimensions, it was felt that multiple grades were necessary in order to be able to reflect the relative relevance of a component with respect to its sub-components. For example, a document component may be *more* exhaustive than any of its sub-components alone given that it covers *all* (i.e. the union of) the aspects discussed in each of the sub-components. Similarly, sub-components may be *more* specific than their parent components, given that the parent components may cover multiple topics, including irrelevant ones.

The relevance degree of an assessed component, given by the combined values of exhaustivity and specificity, is denoted as $(e, s) \in ES$, where $ES = \{(0,0), (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)\}$.

A consequence of separating the two dimensions is that evaluation measures need to be able to either handle the dimensions separately or be able to combine them in a way that reflects appropriate user expectations. As mentioned before, the quantisation functions aim to do just that. They provide a relative ordering of the various combinations of (e, s) values and a mapping of these to a single relevance scale: $\mathbf{f}_{quant}(e, s)\colon ES \rightarrow [0, 1]$[3].

A number of relevance value functions have been in use throughout the years reflecting various user preferences. Some of these functions, e.g. strict quantisations, result in binary relevance values, while others, e.g. generalised or SOG (see Equation 2), result in multiple degree relevance scales having a range of values in $[0, 1]$. While strict quantisations lend their results suitable for an evaluation measure based on counting mechanisms, others that produce non-binary relevance values require alternative measures like generalised precision and recall or cumulated gain [13, 8, 1].

One question that remains open, even with the defined "most specific" task, is the question of how to decide about the exact mapping to be employed (within a given quantisation function). How much would a $(2, 3)$ element be worth to the user, or how much more could it be worth than a $(1, 1)$ element?

3.2 Binary relevance

With all the additional effort involved in producing relevance assessments according to the two dimensions and along the multiple grades, arguments have been raised time and time again for the use of a simple binary relevance measure. The report of the INEX 2003 workshop [9] reports on a similar discussion, where the benefits of graded relevance assessments have again been pointed out (see [13, 7, 21]). An additional problem with binary relevance assessments is that it becomes no longer possible to reason about relative preferences among related relevant elements (i.e. component vs. its sub-components).

3.3 Continuous scale

In order to decrease assessment effort, a highlighting procedure is being considered for INEX 2005 (INEX organizers mailing list), and may even have been put in place by the time of this workshop. A proposed process for assessment is as follows:

- In the first pass, assessors highlight text fragments that contain only relevant information
- In the second pass, assessors judge the exhaustivity level of any elements that have highlighted parts.

As a result of this process, any elements that have been fully highlighted will be automatically labeled as fully specific. For example, if the last paragraph of a section (say `sec1`) and the first two paragraphs of the next section (`sec2`) have been highlighted, then the three paragraphs and any of their descendants will be marked as fully specific (e.g. $s = 1 = 100\%$). The specificity of any other (partially highlighted) elements will be calculated automatically as some function of the contained relevant and irrelevant content (e.g. in the simplest case as the ratio of relevant content to all content, measured in number of words or characters). The two sections in our example may then get a specificity score of $s = 10/100 = 10\%$ and $s = 20/100 = 20\%$, respectively, assuming that each paragraph consists of 10 words and each section has 10 paragraphs. The same procedures can be applied when highlighting is done at the sentence or word level.

The main advantage of this highlighting approach is that assessors will now only have to judge the exhaustivity level of the elements that have highlighted parts (in the second phase). A vital consideration, however, is that the highlighting must be based solely on the specificity dimension (e.g. ignoring exhaustivity in the first phase). Assessors should

[3]Note that the quantisation functions used within the inex-2003 metric provide a separate mapping for exhaustivity, $\mathbf{f}'_{quant}(e)\colon E \rightarrow [0, 1]$ and specificity, $\mathbf{f}'_{quant}(s)\colon S \rightarrow [0, 1]$, where $E = \{0, 1, 2, 3\}$ and $S = \{0, 1, 2, 3\}$

be made aware not to highlight larger contexts because these are more exhaustive, if at the same time they are less specific (i.e. contain irrelevant fragments). It is important that only purely relevant information fragments get highlighted.

Although, with this semi-automated method, specificity will be measured on a continuous scale, with a simple quantisation method, it can be mapped onto the already established 4 point specificity scale, if desired. However, the use of a continuous scale for specificity may also simplify the evaluation as it will no longer require a relative ordering of (e, s) pairs, but allows for a more natural combination of the two dimensions.

Although there have been suggestions for also employing a continuous scale for the exhaustivity dimension, this option has not yet been explored. It is not yet clear to us what benefits this may have and if it could lead to a reliable measure.

4. WHY DO WE NEED AN IDEAL RECALL-BASE?

In INEX, the recall-base consists of sets of overlapping elements (which will remain the case even with the proposed new assessment procedure). For example, from the XML article of co/2001/r7022.xml, all elements shown in Figure 1 form part of the recall-base for INEX 2004. As detailed in [10], this so-called *overpopulated recall-base* can lead to skewed and misleading effectiveness results if it is ignored by the employed evaluation metric. The root of this problem lies in the fact that the recall-base contains more reference elements than an ideal system should in fact retrieve. In fact, if the problem is ignored by the metric then perfect recall can only be reached by systems that return all the relevant reference components of the recall-base, including all the overlapping elements [16, 10, 3, 17]. Such retrieval behaviour, however, contradicts the definition of an effective XML retrieval system.

Following on from the focused task definition in section 2.3, systems should return only the most specific non-overlapping elements from an XML tree of relevant nodes. Based on the thorough task, ideal elements are those that score highest along a path of the XML tree according to a chosen quantisation function. Elements that correspond to such ideal nodes must also be selected from the recall-base. Given a suitable procedure, we can define an 'ideal recall-base' as a collection of ideal nodes, where overlap between reference elements is completely removed. All remaining components of the original recall-base may then be considered as near-misses.

The constructed ideal recall-base could be used (by itself) for evaluating XML retrieval systems using traditional metrics (i.e. recall and precision). In such an evaluation setting, however, systems would be measured against a rather strict ideal scenario, where only exact matches between retrieved elements and ideal reference elements are considered a hit. However, given the possibly fine graded structure of an XML document, the judgement to only credit systems that are able to return exactly the ideal components may seem too harsh, especially since the retrieval of near-misses may still be considered useful for a user when the ideal component is not found.

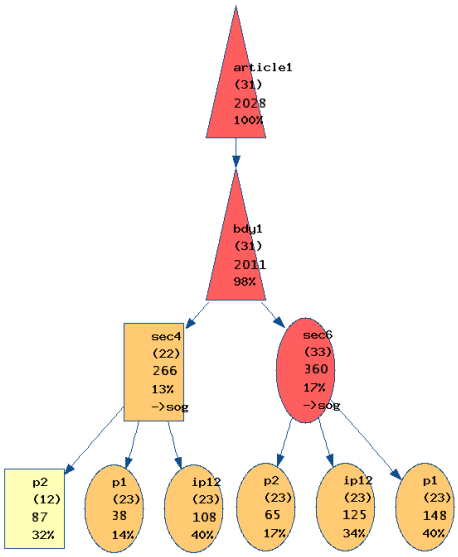


Figure 1: Sample assessments showing only relevant nodes (i.e. $e > 0$ and $s > 0$) for topic 163 in the article file co/2001/r7022.xml. For each node, the node name, the assessment value pair (e, s), the size in number of words and the size ratio to its parent node is shown. Nodes marked as "→ sog" are the selected ideal nodes based on the SOG quantisation function.

A better solution can be reached by the combined usage of the full recall-base and the derived ideal recall-base: elements in the ideal recall-base represent the desired target components that should be retrieved, while all other elements in the full recall-base (or even in the full collection) may be rewarded partial score. The main significance of the definition of an ideal recall-base is that it supports the evaluation viewpoint whereby components in the ideal recall-base *should* be retrieved, while the retrieval of near-misses *could* be rewarded as partial successes, but other systems *need not* be penalised for not retrieving such near-misses.

4.1 How to build an ideal recall-base

An ideal recall-base is a set of ideal result nodes selected from the full recall-base, where the selection process must follow assumptions regarding the given retrieval task and user behaviour. In [10], a proposed selection process was based on a chosen quantisation function, representing a user model, and the following methodology. Given any two components on a relevant path[4], the component with the higher quantised score is selected. In case two components' scores are equal, the one deeper in the tree is chosen. The procedure is applied recursively to all overlapping pairs of components along a relevant path until one element remains.

[4]A relevant path is defined as a path in an article file's XML tree, whose root node is the `article` element and whose leaf node is a relevant component (i.e. $(e>0, s>0)$) that has no or only irrelevant descendants. E.g. in Figure 1 there are 2 relevant paths.

After all relevant paths have been processed, a final filtering is applied to eliminate any possible overlap among ideal components, keeping from two overlapping ideal paths the shortest one. The resulting ideal recall-base contains the best elements to return to a user based on the assumptions that overlap between result nodes should be avoided and that the user's preferences are reflected within the employed quantisation function.

For example, using the SOG quantisation function (Equation 2), the ideal nodes selected from the XML tree shown in Figure 1 are `sec[6]` and `sec[4]`.

Based on the proposed new highlighting procedure and the focused task, the ideal elements will be the largest fully specific (or most specific[5]) elements that directly contain the highlighted relevant information.

An alternative method is proposed by Benjamin Piwowarski (in PRUM's implementation in EvalJ), where a node x is selected as ideal if:
1.) $f_{quant}(x) > 0$ AND
2.) for any descendant z of x $f_{quant}(z) < f_{quant}(x)$ AND
3.) for any ancestor y of x $f_{quant}(x) \geq f_{quant}(y)$ OR there exists a descendant z of y for which $f_{quant}(z) \geq f_{quant}(y)$.

For example, using the SOG quantisation function, the ideal nodes selected from the XML tree shown in Figure 1 are `sec[6]`, `sec[4]/ip1[2]`, `sec[4]/p[1]` and `sec[4]/p[2]`.

The difference between the two methods is that the latter places additional emphasis on selecting nodes deeper in the tree and can also cater for some assessment error, while the former relies only on the assessors' judgements. This is illustrated in Table 1, which shows the obtained ideal recall-bases for each of the sample XML trees of Figure 2, using the SOG quantisation function. For tree a) both methods select the same ideal node. For tree b) Kazai's method selects nodes 2 and 3 initially then keeps only node 2, while Piwowarski's method selects nodes 3 and 5. For tree c) Kazai's method selects node 3, while Piwowarski's method selects all relevant leaf nodes: 4,5,6,7,8 and 9.

It could be debated as to which method is better than the other. For example, in tree b) one might argue that nodes 3 and 5 provide a better representation of our user's ideal results based on the SOG user model and if the assessor had judged node 2 (`bdy[1]`) as $(e, s) = (3, 1)$ (instead of $(3, 2)$) then Kazai's method would also select these nodes as ideal. Without looking at element size, we cannot be sure if the assessor's decision was a correct one or a possible mistake[6]. An advantage of the continuous specificity scale and the highlighting assessment procedure would be that such problems would be eliminated.

Tree c) represents an interesting situation, whereby the two sets of ideal nodes cover the same content[7], but - depending on how they are presented to the user (e.g. highlighted text in an XML document or XML elements in a ranked list) and what measure is employed - could obtain different effectiveness results. This motivates the need for more elaborate methods for constructing ideal recall-bases, taking into account result presentation.

Table 1: Ideal nodes for Kazai's and Piwowarski's methods for the XML trees in Figure 2

Method	Tree a)	Tree b)	Tree c)
Kazai [10]	3.	2.	3.
Piwowarski [17]	3.	3. and 5.	4. - 9.

A metric should then take the chosen ideal recall-base as its parameter. The total score for retrieving any number of elements in a given sub-tree, having an ideal node as its root, should be limited by the quantised score of the ideal node as its maximum. For example, if in tree c) node 3 (`sec[1]`) is the ideal node (score of 1), then a run consisting of `p[4]` (0.75), `p[5]` (1) and `p[2]` (0.75) would score $min(1 - 0, 0.75) = 0.75$, $min(1 - 0.75, 1) = 0.25$ and $min(1 - (0.75 + 0.25, 0.75) = 0$, respectively. The score of retrieving an ascendant of a set of ideal nodes should, in our opinion, be based on the result's quantised score. For example, if in tree c) the ideal nodes are all the relevant leaf nodes, then `sec[1]`'s score is simply 1. It may be argued that if the assumed result presentation is highlighted text, then `sec[1]` should get the same score as the total score of the ideal nodes as it would highlight the same content. However, for this to be true we need to assume that the user has access to the context of a highlighted paragraph, in which case the ideal node should anyhow be the largest most specific element (focused task).

4.2 Don't call me ideal, I am only E3S3

There seems to be a widely popular misunderstanding of E3S3 (i.e. $(e, s) = (3, 3)$) elements being referred to as ideal results, and arguments are being raised as to how it should not be possible to have multiple nodes on a given path assessed as E3S3.

We would like to reiterate here that $(e, s) = (3, 3)$ or even $s = 3$ are NOT sufficient conditions of ideal elements! Specificity is simply a measure of the amount of relevant content vs. irrelevant content within a node. An element is highly specific ($s = 3$) *iif* it contains only relevant information (or only minimal irrelevant information), and so there is absolutely no reason why there could not be more than one highly specific nodes on a path.

This may be easier to see when considering the new highlighting assessment process. Take for example a highlighted section. The fact that it has been highlighted means that it must be fully specific, i.e. contains only relevant information. Therefore, each of its paragraph child nodes must also be fully specific (since the section contains no irrelevant information). Now, take a highlighted article that is also highly exhaustive. There is no reason why it could not have descendant elements that are also highly exhaustive

[5]For example, if only a sentence of a paragraph has been highlighted then the paragraph is selected as the ideal element.
[6]If `bdy[1]` consists only of the two sections judged relevant, then $s = 2$ is reasonable. However, if it has other irrelevant sections, then $s = 1$ would seem more appropriate.
[7]Assuming `sec[1]` does not have a text child node.

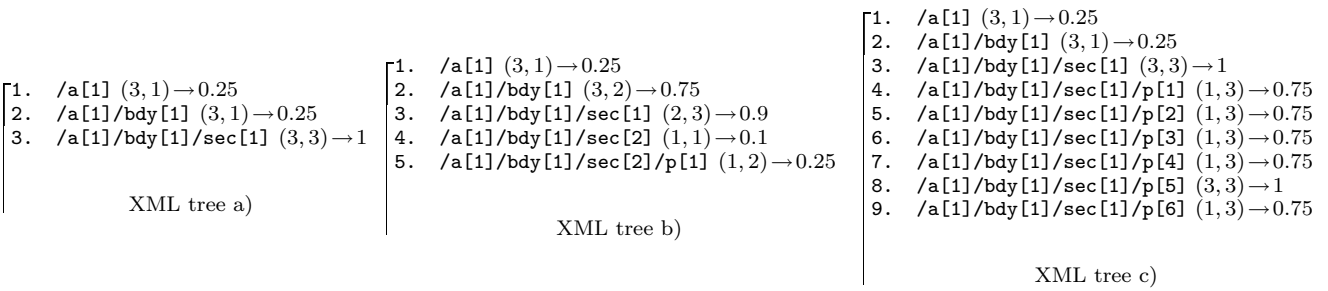


Figure 2: Relevance assessments for sample XML trees. For each node, its path (with `article` shortened to `a`), exhaustivity and specificity values (e, s), and derived SOG quantised values are shown. Note that only relevant nodes are included.

(e.g. `bdy`, `app` or `sec` nodes), therefore producing a number of highly specific and highly exhaustive nodes on a path.

5. REQUIREMENTS FOR METRICS

In the previous sections we have detailed a number of requirements that a suitable measure for XML IR should take into account. We summarise these factors here.

In section 2 we stated that the main criterion of any evaluation measure is that it should be able to rank systems according to how well they satisfy a user's information need given a retrieval task and a model of user behaviour. Then during our examination of the retrieval task and user model, we noted that - due to the varying granularity of retrieval units - element size or required reading time should be taken into account when measuring users' effort to view result elements. Because of the structural relationships that exist among result elements, users' browsing behaviour should be considered. An aspect of this is that near-miss components may be considered as partial successes. Another aspect is that overlap should also need to be handled by a suitable metric.

For the focused and thorough tasks, metrics need only to consider the output as a ranked list of XML elements, with most relevant elements at the top of the ranking. Users are assumed to view the ranked list in a linear fashion, moving from the top of the list down, stopping either when the end of the list is reached, or at a point where their information need has been satisfied or where they give up. For the fetch and browse strategies, the evaluation may need to consider additional factors due to the clustering of related results.

Given that INEX employs two relevance dimensions, a measure of effectiveness should be able to either handle these dimensions separately or be able to combine them in a way that reflects a set task and user model. The metric must also be able to handle multiple degree scales of relevance (where counting mechanisms are no longer suitable). Following the proposal for a continuous scale for specificity, the ideal metric should be flexible enough to cater for both discrete and continuous scales.

As a result of the overlap of reference elements within the INEX recall-base, a suitable metric should also incorporate appropriate mechanisms to derive ideal recall-bases from the full set of assessments based on a given user model. The metric should also employ appropriate score normalisation mechanisms to ensure that the total achievable score for retrieving any combinations of relevant nodes (including the ideal node) from the sub-tree of an ideal node does not exceed the score obtainable by retrieving the ideal node itself.

In Section 6.1, we will look at the various current and proposed INEX metrics and attempt to answer whether they meet these requirements:

- Element size: Consider user effort as a function of varying granularity result elements
- Near-misses: Consider near-miss components as partial successes
- Overlap: Do not penalise systems that do not return overlapping nodes
- Output: Take into account ranking and other non-linear presentation
- Exhaustivity and specificity: Handle dimensions separately or able to combine them
- Multiple degrees: Handle multiple degree scales (and continuous scales)
- Ideal recall-base: Incorporate mechanisms to select ideal nodes from the full recall-base
- Normalisation: Incorporate mechanisms to normalise the scoring of elements in the sub-trees of ideal nodes.

6. AN ABUNDANCE OF METRICS

Up to date the following metrics have been used and/or proposed (a more detailed summary of each can be found in the Appendix)[8]:

i2: The inex-2002 (aka. inex_eval) metric [5] applies an intuitive extension of the measure of *precall* [18] to document components and computes the probability $P(rel|retr)$ that a component viewed by the user is relevant.

[8]A further metric, Expected Ratio of Relevant (ERR) [16] is not discussed here.

Table 2: Metrics and requirements matrix (y: yes, n: no, i: indirectly)

Requirements:	i2	i3	XCG	PRUM
Element size	n	y	i	n
Ideal recall-base	n	i	y	y
Near-misses	n	i	y	y
Overlap	n	y	y	y
Output: linear	y	y	y	y
Output: non-linear	n	n	n	n
Exh/Spec	y	y	y	y
Multiple degrees	n	n	y	n
Normalisation	n	n	y	n

i3: The inex-2003 (aka. inex_eval_ng) metric [6, 4] is based on an interpretation of the relevance dimensions within an ideal concept space [23]. Instead of measuring recall or precision after a certain number of document components retrieved, the total size of the retrieved document components is used as the basic parameter. For our experiments we use the version of inex-2003 detailed in [4].

XCG: The XCG (cumulated gain for XML) metrics [10, 11] are an extension of the set of cumulated gain based metrics proposed in [8] for measuring effectiveness in a traditional IR setting but considering multiple degrees of relevance.

PRUM: The PRUM (Precision Recall with User Modelling) [17] metric is an extension of the traditional recall precision metrics that considers users' browsing behaviour.

T_2I: The T_2I (Tolerance to Irrelevance) metric [3] measures success or failure based on whether the user finds relevant text starting from a returned entry point before his/her tolerance to irrelevance is reached.

The inex-2002 metric has been criticised for not considering overlap and leading to misleading effectiveness scores [10]. The inex-2003 metric's disadvantage is that it is hard to interpret and assumes that relevant information is distributed uniformly throughout a component. A shortcoming of the XCG metrics is that effectiveness is only measured at rank positions and not at recall values. PRUM is based on counting mechanisms, where the interpretation of results based on non-strict quantisations is not clear. In addition, its numerous parameters and their exact estimations may appear more of an obstacle than an advantage. T_2I has so far remained a theoretical model without concrete integration into a specific measure, and as such is not further discussed.

6.1 Metrics and requirements

In this section, we take a look at all current and proposed metrics and how they satisfy the requirements identified in the previous sections.

Table 2 lists the collected metric requirements and whether these are catered for by the various metrics.

Element size has only been considered explicitly within the definitions of recall and precision of the inex-2003 metric. XCG uses element size indirectly when calculating the relevance score of a partially seen element (see Equation 8). Element size could, however, be incorporated into PRUM, inex-2002 and directly into XCG (i.e. to measure cumulated gain against the size of the consulted text instead of its rank) by adding a quantisation function that uses element size. It is arguable, however, whether larger relevant texts should warrant higher effectiveness scores (as is the case for inex-2003. It may be more intuitive to consider element size only for irrelevant information (T_2I) or when irrelevant and relevant information is combined (as in T_2I and XCG) in a component as the amount of irrelevant information a user needs to wade through directly influences his/her satisfaction with the system.

Both XCG and PRUM make use of ideal recall-bases. The mechanisms for deriving an ideal recall-base, based on a given user model (quantisation function and assumptions about overlap), are currently implemented as an integral part of the metrics. However, there are plans to allow for a more flexible setup, where arbitrary ideal recall-bases can be applied as a parameter of the metrics. The version of the inex-2003 metric detailed in [4] defines an entity Rel^U, which represents the maximum number of relevant concepts in the full recall-base (counting a relevant concept only once)[9]. This could hence be interpreted as the total relevance score of an ideal recall-base, whose elements are chosen to maximise the total relevance score for the collection's XML tree. In general, this leads to the ideal recall-base consisting of relevant leaf nodes (i.e. the deepest relevant nodes). Since the definition of Rel^U is fixed (due to concept space), it can only be associated with a single given user model and result presentation (a bit like recall and precision).

Both PRUM and XCG are able to give partial reward for near-misses (due to the fact that they both make use of an ideal recall-base). Unlike XCG, however, PRUM is also able to consider irrelevant sibling nodes as near-misses. PRUM does this by increasing the score of a result element (even if irrelevant) if it has structural links to relevant content (based on assumptions about the user's browsing behaviour: no, hierarchical or T_2I browsing). XCG relies only on the ideal and full recall-bases for determining a near-miss. The latest version of inex-2003 [4] also (indirectly) supports the evaluation of near-misses due to scoring elements based on the full recall-base while the collection's total relevance score is based on Rel^U.

Overlap is handled by all metrics except the inex-2002 measure. In XCG overlap is handled within the relevance value functions, which return a node's unmodified quantised value if it has not yet been seen, and otherwise calculate a modified relevance score if it has been seen in full or in part. The relevance value of partially seen elements is derived recursively based on the size and relevance score of the node's not-yet-seen descendants. In inex-2003, overlap is handled

[9]The earlier version of the inex-2003 metric [6] calculated total relevance as $\sum_{i=1}^{N} q_e(e)$ over all N elements of the full recall-base, which resulted in the same problems as with the inex-2002 metric that 100% recall could only be reached by systems returning the full recall-base.

in a similar way, by only considering the not-yet-seen parts, but the relevance score is estimated by assuming that relevant information is distributed uniformly within the component. This means that a section will still obtain a score even if its only relevant paragraph has already been seen. PRUM employs probability estimations for a user's browsing behaviour, and updates the probability of a node being seen by the user depending on its structural relationship to the currently visited node and assumptions about the user's interaction (i.e. no, hierarchical or T_2I browsing). The more structurally related elements have been returned to the user and hence the more chances the user had to access the current result element, the more its score is reduced.

There is no difference between the four metrics as far as the output presentation is concerned: they are all able to evaluate linear ranked result lists. Further investigation of how the metrics can be adapted to deal with clustered representations is required.

All the metrics are able to cope with the two relevance dimensions via the use of quantisation functions.

Since all metrics, except XCG, are extensions of recall and precision, they are all based on counting mechanisms that result in non-perfect effectiveness for ideal runs (see Section 6.2). For example, although PRUM does work with multiple degree relevance scales, it only produces perfect score for an ideal run, if a strict quantisation (or the recently added "binary" option) is applied.

All metrics can adopt a continuous specificity scale via the definition of a suitable quantisation function. For example, a simple quantisation function may be given as: $f_{quant} = q_e(e) \cdot q_s(s)$, where $q_e(e) = e/3$ and $q_s(s) = s$ if $s \in [0, 1]$ (where $s = 1$ would mean fully specific). The function f_{quant} would be used by the metrics inex-2002, XCG and PRUM, while the functions $q_e(e)$ and $q_s(s)$ could be used directly in inex-2003.

Normalisation mechanisms to ensure that the total achievable score for retrieving any combinations of relevant nodes (including the ideal node) from the sub-tree of an ideal node does not exceed the score obtainable by retrieving the ideal node itself are implemented in XCG [11].

From the above, it seems that no single metric ticks all the requirements, although the inex-2002 metric seems to be the one lagging behind all others. A reason for this is that most of the problems associated with the evaluation of XML retrieval have not actually came to light until after the first effectiveness results were in. For example, implicit assumptions about overlap (i.e. that systems would avoid returning overlapping nodes) meant that overlap was not explicitly considered by the metric. An obvious question is whether the inex-2002 metric could be extended upon to cater for the additional requirements. We will examine this question in future work.

6.2 What do they measure

In this section we detail the results of some very simple experiments, where we investigated the behaviour of four of the INEX metrics with the use of a single relevant XML tree (taken from the INEX 2004 recall-base for the topic 163). We used four simulated runs for the experiments: see Table 3. The result elements of all runs have been sorted according to our chosen quantization function: SOG (Equation 2).

Table 3: Simulated runs

`frb_SOG_163_r7022: #All relevant nodes in topic 163's assessments for co/2001/r7022.xml, sorted by SOG quantised value (see Figure 1).`
1. `/article[1]/bdy[1]/sec[6]` $(3, 3) \rightarrow 1$
2. `/article[1]/bdy[1]/sec[4]/ip1[2]` $(2, 3) \rightarrow 0.9$
3. `/article[1]/bdy[1]/sec[4]/p[1]` $(2, 3) \rightarrow 0.9$
4. `/article[1]/bdy[1]/sec[6]/ip1[2]` $(2, 3) \rightarrow 0.9$
5. `/article[1]/bdy[1]/sec[6]/p[1]` $(2, 3) \rightarrow 0.9$
6. `/article[1]/bdy[1]/sec[6]/p[2]` $(2, 3) \rightarrow 0.9$
7. `/article[1]/bdy[1]/sec[4]` $(2, 2) \rightarrow 0.5$
8. `/article[1]` $(3, 1) \rightarrow 0.25$
9. `/article[1]/bdy[1]` $(3, 1) \rightarrow 0.25$
10. `/article[1]/bdy[1]/sec[4]/p[2]` $(1, 2) \rightarrow 0.25$

`irb_SOG_163_r7022: #Ideal nodes from the full recall-base run above, based on Kazai's method and sorted by SOG quantised value.`
1. `/article[1]/bdy[1]/sec[6]` $(3, 3) \rightarrow 1$
2. `/article[1]/bdy[1]/sec[4]` $(2, 2) \rightarrow 0.5$

`reverse_irb_SOG_163_r7022: #Nodes from the ideal run above, but in reverse order.`
1. `/article[1]/bdy[1]/sec[4]` $(2, 2) \rightarrow 0.5$
2. `/article[1]/bdy[1]/sec[6]` $(3, 3) \rightarrow 1$

`lo_SOG_163_r7022: #All relevant leaf nodes from the full recall-base run, sorted by SOG quantised value.`
1. `/article[1]/bdy[1]/sec[6]/ip1[2]` $(2, 3) \rightarrow 0.9$
2. `/article[1]/bdy[1]/sec[6]/p[1]` $(2, 3) \rightarrow 0.9$
3. `/article[1]/bdy[1]/sec[6]/p[2]` $(2, 3) \rightarrow 0.9$
4. `/article[1]/bdy[1]/sec[4]/ip1[2]` $(2, 3) \rightarrow 0.9$
5. `/article[1]/bdy[1]/sec[4]/p[1]` $(2, 3) \rightarrow 0.9$
6. `/article[1]/bdy[1]/sec[4]/p[2]` $(1, 2) \rightarrow 0.25$

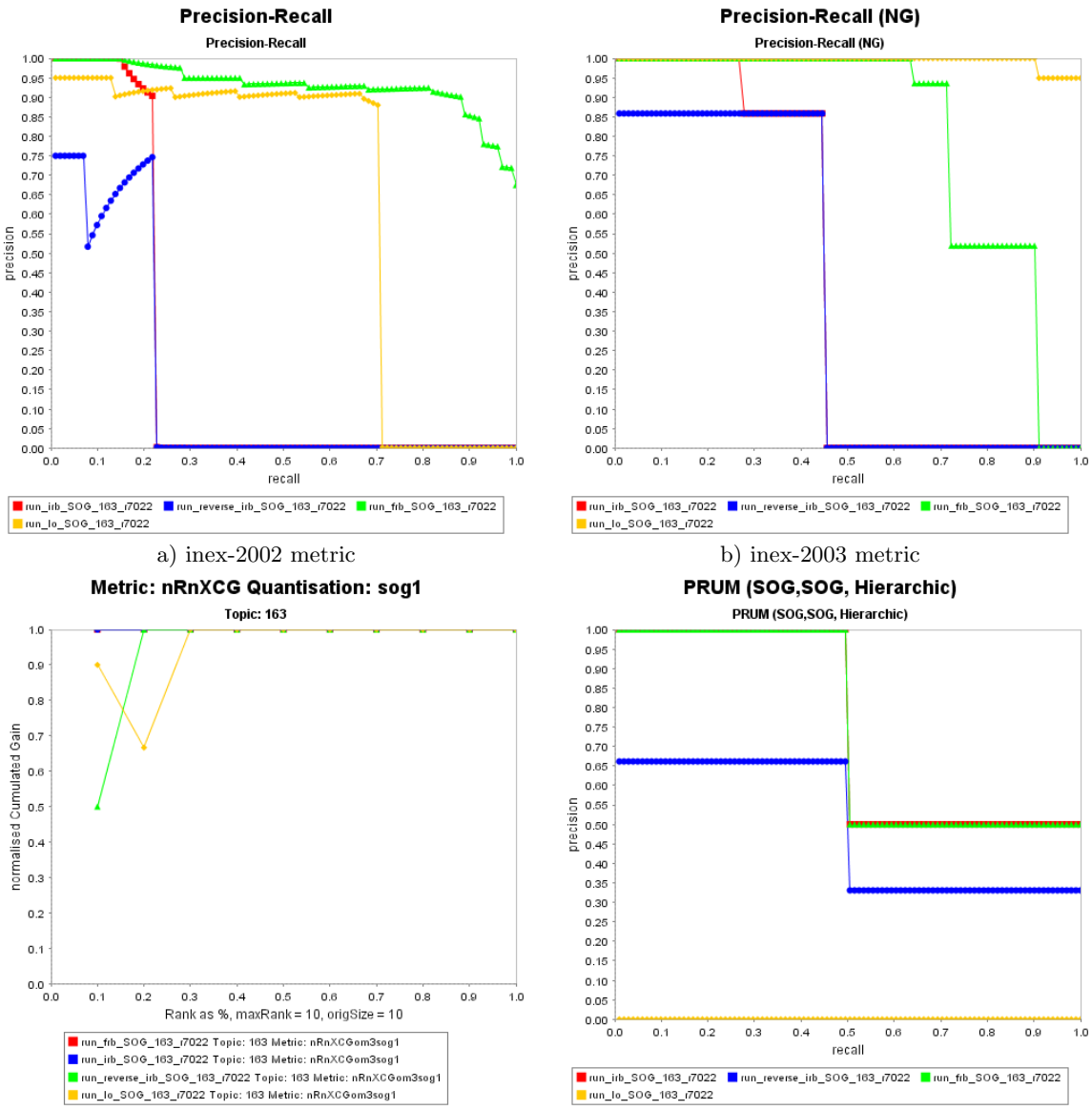
We used the EvalJ source code for the evaluation[10], which implements all four metrics within a single java project.

The runs were evaluated against a full recall-base consisting only of the relevant nodes from the article file `co/2001/r7022` from the assessments of topic 163 (Figure 1). For PRUM and XCG, the ideal recall-bases were automatically generated using the SOG quantisation function during the evaluation (using Kazai's algorithm, detailed in section 4.1)[11].

As it can be seen in Figure 3, the inex-2002 metric ranks the reverse ideal run worst followed by the ideal run, which performs slightly better than its reversed version at low recalls. This is intuitive and reflects that highly relevant elements are expected to be ranked before less relevant elements. Ta-

[10] https://sourceforge.net/projects/evalj/

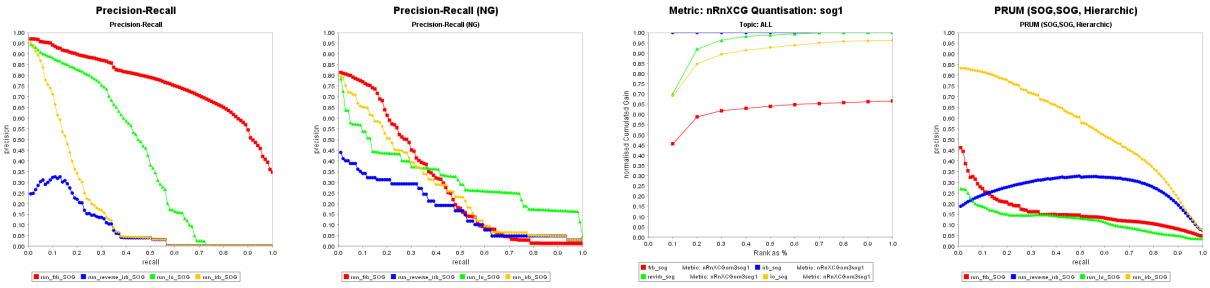
[11] Note that for this, we modified PRUM's code in EvalJ so that the same ideal recall-base is created as with XCG: evalj.corpus.AssessDoxel.addIdealDoxels method.



a) inex-2002 metric b) inex-2003 metric

c) nXCG metric d) PRUM metric

Figure 3: Effectiveness scores for a single XML tree in the article file co/2001/r7022 in topic 163, using the SOG quantisation



a) inex-2002 metric b) inex-2003 metric c) nXCG metric d) PRUM metric

Figure 4: Effectiveness scores for all INEX 2004 CO topics, using the SOG quantisation

Table 4: Effectiveness scores for the irb_SOG_163_r7022 simulated run (see Appendix for formulas and Figure 1 for element size information)

run	i2	i3	XCG
	$n = 6.75$	$Rel^U = 3.67$	$max_{XCG_{ideal}} = 1.5$
1.	$x = \frac{1}{6.75} = 0.14$ $p = \frac{1}{1+0+\frac{1\cdot 0}{1+1}} = 1$	$r = \frac{1\cdot\frac{360}{360}}{3.67} = 0.27$ $p = \frac{1\cdot 360}{360} = 1$	$rank = 1$ $nXCG = \frac{1}{1} = 1$
2.	$x = \frac{1+0.5}{6.75} = 0.22$ $p = \frac{1.5}{1.5+0+\frac{0.5\cdot 0.5}{1+0.5}} = 0.9$	$r = \frac{1\cdot\frac{360}{360}+0.67\cdot\frac{266}{266}}{3.67} = 0.45$ $p = \frac{1\cdot 360+0.67\cdot 266}{360+266} = 0.86$	$rank = 2$ $nXCG = \frac{1+0.5}{1.5} = 1$

Table 5: Effectiveness scores for the reverse_irb_SOG_163_r7022 simulated run (see Appendix for formulas and Figure 1 for element size information)

run	i2	i3	XCG
	$n = 6.75$	$Rel^U = 3.67$	$max_{XCG_{ideal}} = 1.5$
1.	$x = \frac{0.5}{6.75} = 0.07$ $p = \frac{0.5}{0.5+0+\frac{0.5\cdot 0.5}{1+0.5}} = 0.75$	$r = \frac{0.67\cdot\frac{266}{266}}{3.67} = 0.18$ $p = \frac{0.67\cdot 266}{266} = 0.67$	$rank = 1$ $nXCG = \frac{0.5}{1} = 0.5$
2.	$x = \frac{1+0.5}{6.75} = 0.22$ $p = \frac{1.5}{1.5+0.5+\frac{1\cdot 0}{1+1}} = 0.75$	$r = \frac{1\cdot\frac{360}{360}+0.67\cdot\frac{266}{266}}{3.67} = 0.45$ $p = \frac{0.67\cdot 266+1\cdot 360}{360+266} = 0.86$	$rank = 2$ $nXCG = \frac{0.5+1}{1.5} = 1$

Table 6: Effectiveness scores for the frb_163_r7022_sog simulated run (see Appendix for formulas and Figure 1 for element size information)

run	i2	i3	XCG
	$n = 6.75$	$Rel^U = 3.67$	$max_{XCG_{ideal}} = 1.5$
1.	$x = \frac{1}{6.75} = 0.14$ $p = \frac{1}{1+0+\frac{1\cdot 0}{1+1}} = 1$	$r = \frac{1\cdot\frac{360}{360}}{3.67} = 0.27$ $p = \frac{1\cdot 360}{360} = 1$	$rank = 1$ $nXCG = \frac{min(1-0,1)}{1} = 1$
2.	$x = \frac{1+0.9}{6.75} = 0.28$ $p = \frac{1.9}{1.9+0+\frac{0.9\cdot 0.1}{1+0.9}} = 0.975$	$r = \frac{1\cdot\frac{360}{360}+0.67\cdot\frac{108}{108}}{3.67} = 0.45$ $p = \frac{1\cdot 360+1\cdot 108}{360+108} = 1$	$rank = 2$ $nXCG = \frac{1+min(0.5-0,0.9)}{1.5} = 1$
3.	$x = \frac{1.9+0.9}{6.75} = 0.41$ $p = \frac{2.8}{2.8+0.1+\frac{0.9\cdot 0.1}{1+0.9}} = 0.95$	$r = \frac{(1.67+0.67\cdot\frac{38}{38}}{3.67} = 0.637$ $p = \frac{468+1\cdot 38}{468+38} = 1$	$rank = 3$ $nXCG = \frac{1.5+min(0.5-0.5,0.9)}{1.5} = 1$
4.	$x = \frac{2.8+0.9}{6.75} = 0.54$ $p = \frac{3.7}{3.7+0.2+\frac{0.9\cdot 0.1}{1+0.9}} = 0.937$	$r = \frac{(2.34+0.67\cdot\frac{0}{125}}{3.67} = 0.637$ $p = \frac{506+1\cdot 0}{506+0} = 1$	$rank = 4$ $nXCG = \frac{1.5+0}{1.5} = 1$
5.	$x = \frac{3.7+0.9}{6.75} = 0.68$ $p = \frac{4.6}{4.6+0.3+\frac{0.9\cdot 0.1}{1+0.9}} = 0.929$	$r = \frac{(2.34+0.67\cdot\frac{0}{148}}{3.67} = 0.637$ $p = \frac{506+1\cdot 0}{506+0} = 1$	$rank = 5$ $nXCG = \frac{1.5+0}{1.5} = 1$
6.	$x = \frac{4.6+0.9}{6.75} = 0.81$ $p = \frac{5.5}{5.5+0.4+\frac{0.9\cdot 0.1}{1+0.9}} = 0.924$	$r = \frac{(2.34+0.67\cdot\frac{0}{65}}{3.67} = 0.637$ $p = \frac{506+1\cdot 0}{506+0} = 1$	$rank = 6$ $nXCG = \frac{1.5+0}{1.5} = 1$
7.	$x = \frac{5.5+0.5}{6.75} = 0.88$ $p = \frac{6}{6+0.5+\frac{0.5\cdot 0.5}{1+0.5}} = 0.90$	$r = \frac{(2.34+0.67\cdot\frac{266-108-38}{266}}{3.67} = 0.72$ $p = \frac{506+0.67\cdot(266-108-38)}{506+120} = 0.936$	$rank = 7$ $nXCG = \frac{1.5+min(0.5-0.5,0.5)}{1.5} = 1$
8.	$x = \frac{6+0.25}{6.75} = 0.925$ $p = \frac{6.25}{6.25+1+\frac{0.25\cdot 0.75}{1+0.25}} = 0.84$	$r = \frac{(2.64+1\cdot\frac{2028-266-360}{2028}}{3.67} = 0.9$ $p = \frac{586.4+0.34\cdot(2028-266-360)}{626+1402} = 0.524$	$rank = 8$ $nXCG = \frac{1.5+0}{1.5} = 1$
9.	$x = \frac{6.25+0.25}{6.75} = 0.96$ $p = \frac{6.5}{6.5+1.75+\frac{0.25\cdot 0.75}{1+0.25}} = 0.77$	$r = \frac{(3.33+1\cdot\frac{0}{2011}}{3.67} = 0.9$ $p = \frac{1063.08+0.34\cdot(0)}{2028} = 0.524$	$rank = 9$ $nXCG = \frac{1.5+0}{1.5} = 1$
10.	$x = \frac{6.5+0.25}{6.75} = 1$ $p = \frac{6.75}{6.75+2.5+\frac{0.25\cdot 0.75}{1+0.25}} = 0.71$	$r = \frac{(3.33+0.34\cdot\frac{0}{87}}{3.67} = 0.9$ $p = \frac{1063.08+0.67\cdot(0)}{2028} = 0.524$	$rank = 10$ $nXCG = \frac{1.5+0}{1.5} = 1$

bles 4 and 5 show that the reduced effectiveness of the reversed ideal run is due to the irrelevant score obtained for `sec[4]` $(1 - 0.5)$ contributing to Cooper's variable i (irrelevant score at current rank) at rank 1 and then to variable j (irrelevant score up to current rank) at rank 2.

According to the inex-2002 metric, the full recall-base run performs best, followed by the leaf-only run. This is expected as inex-2002 calculates the 100% recall value as the sum of the quantised values of all elements in the full recall-base. Therefore, 100% recall is only reached by the full recall-base run. However, even returning the whole recall-base still does not result in perfect precision. This slope of the precision curve is due to the use of the non-binary relevance scale. Since the quantised exhaustivity and specificity values directly influence the effectiveness score, any quantised values < 1 will result in non-perfect precision scores. For example, at rank 2 of the full recall-base run Cooper's r and s is 0.9, which then results in $i = 1 - 0.9 = 0.1$ and at rank 3 this 0.1 irrelevant score is addedd to Cooper's j variable. While the estimation of these variables in Cooper's formula were based on counting mechanisms (i.e. the number of irrelevant documents), their interpretation in INEX is that of relevance or irrelevence value, where the underlying assumption is that $r = 1 - i$. A problem here is that $r < 1$ does not necessarily mean that a retrieved element contains $1 - r$ irrelevant information, e.g. $f_{SOG}(2, 3) = 0.9$. Employing a quantisation function where $(e, 3) \rightarrow 1$ provides only a partial solution, due to possible XML trees where no $s = 3$ nodes exist, while also resulting in a metric that is insensitive to the level of exhaustivity. One solution to the problem would be to calculate Copper's parameters at a given rank in relation to a maximum ideal relevance score achievable at that rank (instead of using 1), e.g. hence resulting in $i = 0$ in the above example as 0.9 is the highest achievable relevance score at rank 2.

Similarly to the inex-2002 metric, the inex-2003 metric ranks the reverse ideal run worst followed by the ideal run, where the ideal run performs slightly better than its reversed version at low recalls. The reason that these runs don't achieve perfect recall is because Rel^U is calculated from the relevant leaf nodes' quantised scores: $Rel^U = 0.\dot{6} \cdot 5 + 0.\dot{3}$. I.e. our ideal test run does not actually match an ideal run for inex-2003. However, the precision values are not affected by this problem, but are nevertheless inperfect. This is again due to the non-binary relevance grades, where normalising the actual relevance score by a maximum score could provide a solution. Unlike inex-2002, the inex-2003 metric ranks the leaf-only run best followed by the full recall-base run. The reason for this is twofold. On the one hand, the leaf-only run is actually the ideal run for inex-2003, and so it achieves 100% recall. On the other hand, the overlap present in the full-recall-base run leads to reduced performance at various recall levels, depending on the ordering of the elements within the run. The reason that even the perfect run for inex-2003, i.e. the leaf-only run, does not achieve perfect precision is simply because `sec[4]/p[2]` has $q_s(s) < 1$, which directly reduces precision.

XCG is the only metric that shows the ideal run having a perfect score of 1. It also shows that in this special case the run derived from the full recall-base achieves the same result as the ideal run. This is because the first two nodes in frb's ranking match exactly the ideal run (due to results being sorted by SOG value): `sec[6]` and `sec[4]`. Therefore, for the first two ranks, the full recall-base run matches the ideal run and hence achieves maximum score. Due to the fact that all remaining nodes in the full recall-base run overlap with an already retrieved node, no further scores are accumulated. The reverse ideal and leaf only runs perform very similar to the ideal, only dipping slightly at the beginning of the curve. The reverse ideal run's non-perfect score is due to the non-ideal ordering of its elements. The leaf-only run starts off at 0.9 normalised cumulated gain, but then drops due to the fact that the cumulated relevance score of further elements in the sub-tree of `sec[6]` ideal node is not allowed to exceed the ideal node's score (i.e. `sec[6]/ip1[2]` scores 0.9, then `sec[6]/p[1]` scores only 0.1, etc.).

PRUM ranks the ideal and full recall-base runs as best (identical performance). The reverse ideal run comes in at third place and the leaf-only run scores the worst. For PRUM there are two possible relevant units ($P(TR = 1) = 0.5$ and $P(TR = 2) = 0.5$ where TR is the total number of relevant elements). Let's consider both cases: Case 1) $TR = 1$ (i.e. only `sec[4]` is relevant for the user). Then both the full recall-base and the ideal runs achieve precision 1 for all recall levels. The leaf-only run's precision is close to 0 due to the fact that for each result the user will potentially have to inspect all the elements of the database to find the relevant nodes. The reverse ideal run's precision is 1/2. Case 2) $TR = 2$ (i.e. both `sec[4]` and `sec[6]` are relevant to the user). Then for $R = 1$ (i.e. recall level = 0.5) we can arrive at the same observations as in Case 1). This is a problem with the non-binary assessments which is only visible when the number of relevant elements is low. In this case, a human should infer that `sec[6]` is relevant. But for PRUM, $Pr(sec[6]/TR = 2)$ is still 0.5, which more or less implies that the second relevant element will have to be searched again in the whole database $\rightarrow$ precision is near to 0 for this case. After that, curves are obtained taking the average of Case 1) and Case 2).

For reference, we also include the results for a further four simulated runs, which are based on all INEX 2004 CO topics, see Figure 4.

7. CONCLUSIONS

In this paper we focused on issues regarding the evaluation of XML retrieval. We identified a number of requirements that a suitable measure of XML retrieval effectiveness should meet. We commented on the current task definitions and provided suggestions for their future development. We also expressed support for the proposed continuous specificity dimension and reported on an assessment framework to support it. Finally, we examined four of the current and proposed metrics: how they fit the requirements and how they behave when only a single XML tree formed the recall-base.

Our findings showed that although no single metric met all requirements, the XCG and PRUM metrics showed potential. In addition, the XCG metric seemed to behave the most intuitively (best matching expectation), although PRUM also produced intuitive results when a binary quantisation

function was used (Figure not included).

Our future work concentrates on recall-oriented XCG based on [13, 8, 1] adopted to XML, and in particular to INEX.

8. ACKNOWLEDGMENTS

We would like to thank Benjamin Piwowarski for implementing the inex-2002 and inex-2003 metrics in EvalJ, and for his many useful comments and email discussions. Special thanks for providing the ideal recall-base algorithm for PRUM, the basis for the examples in Figure 2 and the explanatory text on PRUM in Section 6.2.

9. REFERENCES

[1] G. Amati. *Probability Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, University of Glasgow, 2003.

[2] W. Cooper. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation*, 19(1):30–41, 1968.

[3] A. de Vries, G. Kazai, and M. Lalmas. Tolerance to irrelevance: A user-effort oriented evaluation of retrieval systems without predefined retrieval unit. In *Recherche d'Informations Assistée par Ordinateur (RIAO 2004)*, Avignon, France, Apr. 2004. To appear.

[4] N. Goevert, N. Fuhr, M. Lalmas, and G. Kazai. Evaluating the effectiveness of content-oriented xml retrieval. *Submitted to Information Retrieval*, 2005.

[5] N. Gövert and G. Kazai. Overview of the INitiative for the Evaluation of XML Retrieval (INEX) 2002. In N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas, editors, *Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX). Dagstuhl, Germany, December 8–11, 2002*, ERCIM Workshop Proceedings, pages 1–17, Sophia Antipolis, France, March 2003. ERCIM. http://www.ercim.org/publication/ws-proceedings/INEX2002.pdf.

[6] N. Gövert, G. Kazai, N. Fuhr, and M. Lalmas. Evaluating the effectiveness of content-oriented XML retrieval. Technischer bericht, University of Dortmund, Computer Science 6, 2003.

[7] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In N. Belkin, P. Ingwersen, and M.-K. Leong, editors, *Proceedings of the 23rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 41–48, Athens, Greece, 2000.

[8] K. Järvelin and J. Kekäläinen. Cumulated Gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (ACM TOIS)*, 20(4):422–446, 2002.

[9] G. Kazai. Report of the inex 2003 metrics working group. In N. Fuhr, M. Lalmas, and S. Malik, editors, *Proceedings of the 2nd Workshop of the INitiative for the Evaluation of XML retrieval (INEX), Dagstuhl, Germany, December 2003*, pages 184–190, April 2004.

[10] G. Kazai, M. Lalmas, and A. de Vries. The overlap problem in content-oriented XML retrieval evaluation. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, 2004.*, pages 72–79. ACM, July 2004.

[11] G. Kazai, M. Lalmas, and A. de Vries. Reliability tests for the xcg and inex-2002 metrics. In 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*, 3493, pages 60–72. Lecture Notes in Computer Science, 2005.

[12] 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.

[13] J. Kekäläinen and K. Järvelin. Using graded relevance assessments in IR evaluation. *Journal of the American Society for Information Science and Technology*, 53(13):1120–1129, 2002.

[14] M. Lalmas. Inex 2005 retrieval task and result submission specification. Technical report, Queen Mary, University of London, 2005.

[15] M. Lalmas and S. Malik. Inex 2004 retrieval task and result submission specification. In 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*, 3493, pages 237–240. Lecture Notes in Computer Science, 2005.

[16] B. Piwowarski and P. Gallinari. Expected ratio of relevant units: A measure for structured document information retrieval. In N. Fuhr, M. Lalmas, and S. Malik, editors, *Proceedings of the 2nd Workshop of the INitiative for the Evaluation of XML retrieval (INEX), Dagstuhl, Germany, December 2003*, pages 158–166, April 2004.

[17] B. Piwowarski, P. Gallinari, and G. Dupret. Precision Recall with User Modelling: Application to XML retrieval. *Submitted for publication*, 2005.

[18] V. Raghavan, P. Bollmann, and G. Jung. A critical investigation of recall and precision. *ACM Transactions on Information Systems*, 7(3):205–229, 1989.

[19] J. Reid, M. Lalmas, K. Finesilver, and M. Hertzum. Best entry points for structured document retrieval - part i: Characteristics. *Information Processing & Management*, 2005. In press.

[20] J. Reid, M. Lalmas, K. Finesilver, and M. Hertzum. Best entry points for structured document retrieval - part ii: Types, usage and effectiveness. *Information Processing & Management*, 2005. In press.

[21] E. Sormunen. Liberal relevance criteria of trec - counting on negligible documents? In K. Järvelin, M. Beaulieu, R. Baeza-Yates, and S. Myaeng, editors, *Proceedings of the Twenty-Fifth Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, 2002.

[22] T. Tombros, B. Larsen, and S. Malik. The interactive track at INEX 2004. In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlavik, editors, *Proceedings of the 3rd Workshop of the INitiative for the Evaluation of XML retrieval (INEX), Dagstuhl, Germany, December 2004*, 2005.

[23] S. Wong and Y. Yao. On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1):38–68, 1995.

APPENDIX

A. METRICS

A.1 The inex-2002 metric

The inex-2002 metric [5] applies the measure of *precall* [18] to document components and computes the probability $P(rel|retr)$ that a component viewed by the user is relevant:

$$P(rel|retr)(x) : \frac{x \cdot n}{x \cdot n + esl_{x \cdot n}} \quad = \quad \frac{x \cdot n}{x \cdot n + j + \frac{s \cdot i}{r+1}} \tag{1}$$

where $esl_{x \cdot n}$ denotes the *expected search length* [2], i.e. the expected number of non-relevant elements retrieved until an arbitrary recall point x is reached, and n is the total number of relevant components with respect to a given topic. In $esl_{x \cdot n}$, let l denote the rank from which the $x \cdot n$th relevant component is drawn. Then j is the score of non-relevant information within the ranks before rank l, s is the relevant score to be taken from rank l, and r and i are the relevant and non-relevant scores in rank l, respectively.

To apply the above metric, the two relevance dimensions are first mapped to a single relevance scale by employing a quantisation function, $\mathbf{f}_{quant}(e, s) \colon ES \rightarrow [0, 1]$. There are a number of quantisation functions currently in use in INEX, e.g. strict or generalised (see Equations 2 and 3 in [9]), each representing a different set of user preferences. The "specificity-oriented generalised" (SOG) quantisation function proposed in [10] is given as:

$$\mathbf{f}_{SOG}(e, s) := \begin{cases} 1 & \text{if} \quad (e, s) = (3, 3) \\ 0.9 & \text{if} \quad (e, s) = (2, 3) \\ 0.75 & \text{if} \quad (e, s) \in \{(1, 3), (3, 2)\} \\ 0.5 & \text{if} \quad (e, s) = (2, 2) \\ 0.25 & \text{if} \quad (e, s) \in \{(1, 2), (3, 1)\} \\ 0.1 & \text{if} \quad (e, s) \in \{(2, 1), (1, 1)\} \\ 0 & \text{if} \quad (e, s) = (0, 0) \end{cases} \tag{2}$$

A.2 The inex-2003 metric

The inex-2003 metric incorporates component size and overlap within the definition of recall and precision (Equations 3 and 4). (For the derivation of the formulae based on an interpretation of the relevance dimensions within an ideal concept space [23] refer to [6].) Instead of measuring, e.g., precision or recall after a certain number of document components retrieved, the total size of the retrieved document components is used as the basic parameter, while overlap is accounted by considering only the increment to the parts of the components already seen. The calculations here assume that relevant information is distributed uniformly throughout a component.

$$\text{recall}_o \frac{\sum_{i=1}^{k} e\,(c_i) \cdot \frac{|c_i'|}{|c_i|}}{Rel^U} \tag{3}$$

$$\text{precision}_o \frac{\sum_{i=1}^{k} s\,(c_i) \cdot |c_i'|}{\sum_{i=1}^{k} |c_i'|} \tag{4}$$

Components $c_1, \ldots, c_k$ in Equations 3 and 4 form a ranked result list, N is the total number of components in the collection, $e(c_i)$ and $s(c_i)$ denote the quantised assessment value

of component c_i according to the exhaustivity and specificity dimensions, respectively, $|c_i|$ denotes the size of the component, while $|c_i'|$ is the size of the component that has not been seen by the user previously. Given a component representation such as a set of (term, position) pairs, $|c_i'|$ can be calculated as:

$$|c_i'| = |c_i - \bigcup_{c \in C[1,n-1]} (c)| \quad (5)$$

where n is the rank position of c_i in the output list, and $C[1, n-1]$ is the set of components retrieved between the ranks $[1, n-1]$.

A.3 The XCG metrics

The XCG metrics are extensions of the cumulated gain (CG) based metrics of [8]. The motivation for the CG metrics was to develop a measure for multi-grade relevance values, i.e. to credit IR systems according to the retrieved documents' degree of relevance. The motivation for XCG was to extend CG in such a way that the problem of overlapping result and reference elements can be addressed within the evaluation framework.

The Cumulated Gain (CG) measure, accumulates the relevance scores of retrieved documents along the ranked list G, where the document IDs are replaced with their relevance scores. The cumulated gain at rank i, $CG[i]$, is computed as the sum of the relevance scores up to that rank:

$$\mathbf{CG[i]} := \sum_{j=1}^{i} G[j] \quad (6)$$

For each query, an ideal gain vector, I, can be derived by filling the rank positions with the relevance scores of all documents in the recall-base in decreasing order of their degree of relevance. A retrieval run's CG vector can then be compared to this ideal ranking by plotting the gain value of both the actual and ideal CG functions against the rank position. We obtain two monotonically increasing curves (levelling after no more relevant documents can be found).

By dividing the CG vectors of the retrieval runs by their corresponding ideal CG vectors, we obtain the normalised CG (nCG) measure. Here, for any rank the normalised value of 1 represents ideal performance. The area between the normalised actual and ideal curves represents the quality of a retrieval approach.

XCG makes use of both the CG and nCG metrics. The extension of these metrics to XML documents, and in particular to INEX, lies partly in the way the relevance score for a given document - or in this case document component - is calculated via the definition of so-called relevance value (RV) functions, and partly in the definition of the ideal recall-bases.

While I is derived from the ideal recall-base, the gain vectors, G, for the runs under evaluation are based on the full recall-base in order to enable the scoring of near-miss components. All relevant components of the full recall-base that are not included in the ideal recall-base are considered as near-misses.

In order to obtain a given component's relevance score (both for I or G) at a given rank position, XCG defines the following result-list dependent relevance value (RV) function:

$$rv(c_i) = f(quant(assess(c_i))) \quad (7)$$

where $assess(c_i)$ is a function that returns the assessment value pair for the component c_i, if given within the recall-base and $(0, 0)$ otherwise. The $rv(c_i)$ function then returns, for a not-yet-seen component c_i, the quantised assessment value pair $quant(assess(c_i))$, where quant is a chosen quantisation functions, e.g. *sog*. In this case $f(x) = x$. For a component, which has been previously fully seen by the user, we have $rv(c_i) = (1 - \alpha) \cdot quant(assess(c_i))$, i.e. $f(x) = (1 - \alpha) \cdot x$. With α set to 1, the RV function returns 0 for a fully seen, hence redundant, component, reflecting that it represents no value to the user any more. Finally, if c_i has been seen only in part before (i.e. some descendant nodes have already been retrieved earlier in the ranking), then $rv(c_i)$ is calculated as:

$$rv(c_i) = \alpha \cdot \frac{\sum_{j=1}^{m} (rv(c_j) \cdot |c_j|)}{|c_i|} + (1 - \alpha) \cdot quant(assess(c_i)) \quad (8)$$

where m is the number of c_i's relevant child nodes.

In addition to the above, the final RV score is obtained by applying a normalisation function, which ensures that the total score for any group of descendant nodes of an ideal result element cannot exceed the score achievable if retrieving the ideal node itself. For example, in Figure 1 the two ideal result nodes for the quantisation function *sog* are `sec[4]` and `sec[6]`. Since these results represent the best nodes for the user, a system returning these should be ranked above others. However, if another system retrieved all the leaf nodes, it may achieve a better overall score if the total RV score for these nodes exceeds that of the ideal nodes. The following normalisation function safeguards against this by ensuring that for any $c_j \in S$:

$$\sum_{c \in S} rv(c) \leq rv(c_{ideal}) \quad (9)$$

where S is the set of retrieved descendant nodes of the ideal node and where c_{ideal} is the ideal node that is on the same relevant path as c_j.

A.4 T_2I

T_2I is based on an alternative definition of correct results. The main idea is that a user merely needs an entry-point into the document that is 'close' to relevant information. Taking this view, a retrieval system produces a ranked list of entry points. The user starts reading the retrieved article from the suggested entry point, giving up when no relevant information is found for some number of words or sentences. So, the user processes the retrieved information until his or her *tolerance to irrelevance* (T_2I) has been reached, at which point the user proceeds to the next system result.

This discourages systems from returning fragments that are too large, since if the entry-point is too far away from the relevant reference component, the user's tolerance to irrelevance will have been exhausted before the relevant informa-

tion has been reached. The problem with multiple system results intersecting the same reference component is eliminated by extending the definition of irrelevance, according to which a previously seen reference fragment is no longer considered relevant.

T_2I variants of three existing evaluation metrics for system performance are given in [3]. Their common underlying principle is that retrieval systems are ranked on their ability to maximise the number of relevant fragments shown to the user while minimising the amount of user effort wasted on irrelevant information. The tolerance to irrelevance is expressed by a single parameter, τ_{NR}, that represents the maximum amount of non-relevant text the user is expected to read before giving up. The length of retrieved relevant components is ignored, assuming that each result has equal value to the user.

A.5 PRUM

The PRUM (Precision-Recall with User Modelling) metric [17] is an extension of the probabilistic precision recall proposed by Raghavan. While the latter supposes a simple user model, where the user consults retrieved elements (elements returned by the retrieval system) independently, PRUM "allow" the user to consult the context of retrieved elements: For each element in the list returned by the retrieval system, the user consults the context of the element. In the context of XML Retrieval, this context is possibly made of the siblings, ancestors and descendants of a retrieved element. Note that this behaviour is defined stochastically, that is we only know that the user has seen a context element with a given probability. For instance, if the user consults a section in the retrieved list, we know that the user has seen this section with a probability 1, and that (s)he has seen also its first paragraph with a probability .95, etc.

Like some other metrics (e.g. XCG), PRUM supposes a set of ideal results, which are the most appropriate non-overlapping elements of the XML database to return to the user. The PRUM metric is then defined as the probability that the user sees a newly relevant element when (s)he consults the context of a retrieved element, knowing that the user wants to see a given amount of relevant units:

$$PRUM(l) = P(Lur|Retr, L = l, Q = q) \qquad (10)$$

where l is the recall level between 0 and 1, q the topic for which PRUM is computed; Retr is the event "the element is in the list consulted by the user" while searching for $I\%$ of the relevant units, and Lur is the event "the element Leads to an Unseen Relevant unit".

Obtrusiveness and relevance assessment in interactive XML IR experiments

Birger Larsen
Department of Information Studies
Royal School of Library and Information Science
Copenhagen, Denmark
blar@db.dk

Anastasios Tombros
Department of Computer Science
Queen Mary University of London
London, United Kingdom
tassos@dcs.qmul.ac.uk

Saadia Malik
Fak 5/IIS, Information Systems
University of Duisburg-Essen
Duisburg, Germany
malik@is.informatik.uni-duisburg.de

ABSTRACT
Ensuring realism in Information Retrieval (IR) experiments (whether laboratory or user based) is always a difficult problem. Obtaining relevance assessments of high quality is of pivotal importance to most studies and a significant challenge. In element retrieval from structured documents, where both whole documents but also parts of documents (elements) may be retrieved as answers, the type of research questions being posed accentuates this problem. In this opinion paper we reflect on the range of aspects we would ideally like to have assessed – in particular with regard to involvement of end-users. The problems involved in requiring assessment of several aspects for each interaction are discussed and a number of alternatives considered.

1. BACKGROUND
Documents formatted in XML and similar mark-up languages are attractive for IR because the mark-up defines the logical structure of the documents and has the potential to assist IR systems in providing more appropriate results to users, i.e., to return relevant document components (i.e. XML elements) rather than whole documents. In addition, the XML tags can have specific semantics that may be exploited purposefully in IR.

This has formed the impetus behind the establishment of INEX – the INitiative for the Evaluation of XML Retrieval. Since 2002 INEX has built test collections to make it possible to test different XML IR approaches [3]. The central research issue is how to exploit the logical structure of documents (explicitly represented by the XML mark-up) to provide more precise answers to end-users. Therefore the relevance assessments not only consider whether retrieved elements are relevant, but also if they have an appropriate level of granularity. Two important, and logical, extensions to traditional IR have been made to facilitate this: In response to earlier criticism against the limited realism of binary relevance assessments [see, e.g., 9] *graded assessments* are used in INEX to express the degree to which a given element is relevant to the information need, and two *different dimensions of relevance* are considered: exhaustiveness and specificity[1]. While the measurement of performance with these assessments has been facilitated by novel non-binary measures [5-7], the use of graded relevance and the two dimensions continue to be debated in INEX: *First*, as the assessments are provided by humans there are concerns about the consistency of them, in particular with such an elaborate two-dimensional relevance scale. *Second*, as not only the retrieved elements but also their descendants and ascendants need to be assessed, the assessment process becomes very laborious when two dimensions of relevance have to be assessed on graded scales.

[1] *Exhaustiveness* describes the extent to which the component discusses the topic of request, and *specificity* the extent to which the component focuses on the topic of request.

2. REALISM AND ASSESSMENT
From 2004 INEX includes an interactive track. Where the main ad hoc track in INEX facilitates laboratory tests of the performance of different XML IR techniques, the interactive track aims at investigating the behaviour of users when interacting with elements of XML documents, and ultimately to facilitate the development of approaches for XML retrieval which are effective in user-based environments. The interactive track thus attempts to put the techniques developed in the ad hoc track into practise so that they may be used by end-users in realistic search environments. An additional purpose of the track is to give useful information to the main track in INEX. Details about the track and results of an initial analysis of the collected data can be found in [10].

For the first year, the interaction of end-users with an XML IR prototype system was studied. The main goal was to investigate if end users would at all like to have elements as answers (rather than the usual whole documents), how they would browse within documents, and which kinds of elements they would assess as relevant. In order to study this in detail, some sort of relevance assessments were needed. Ideally, we would like to have a number of aspects assessed each time a test person has looked at an element:

1. The amount of relevant information the element contains versus irrelevant information (~ *specificity*),
2. How much of the information need can be solved by the element (~ *exhaustiveness*),
3. Whether the retrieved information is *redundant* or not (i.e., has been seen already in other elements)
4. How *useful* the element is overall in solving the information need.

Together with the sequence of interactions, such detailed information on each viewed element could help answer a number of pressing research questions in XML IR, e.g.,

1. What granularity of retrieved elements do users prefer?
2. What do users gain by browsing up/down the XML tree?
3. Would users rather skim larger parts of documents than risk having smaller irrelevant elements?
4. Are users very sensitive to redundant information?
5. …and ultimately, is the retrieval of elements of value to end users, or would they rather just have the full documents?

However, the cognitive load on the test persons would be great if they had to judge and balance all four aspects for each interaction. Experimentally this is undesirable as it is a goal to minimise the cognitive load deriving from factors that do not occur in normal searching behaviour. Having to interrupt the search to give complex relevance assessments may not only result in an unrealistic searching behaviour, but may even be experienced as *obtrusive* by the test persons. This problem is particularly pressing in XML IR where users are likely to browse the document structure to identify other relevant elements than those initially proposed by the system. We would preferably have the test persons to assess the four aspects for each viewed element, and to ensure capturing this information perhaps even forcing the user to do so before moving on to the next element. This has been tried successfully in IR previously in the Okapi experiments, but with much simpler document surrogates and binary relevance assessments [1]. Asking or forcing user to perform complex assessments on all four aspects would inhibit the natural interaction with the system given the much more complex documents and desired aspect to be assessed in XML IR. Here the risk is that the better part of the test persons' attention would be spent on doing the assessments, and not on the interaction.

A compromise between the ideal situation outlined above and a slightly less obtrusive setting was attempted in the interactive track in 2004. The graded scales and two relevance dimensions from the ad hoc track were maintained[2], but merged into to a single dropdown list with 10 points. Figure 1 displays a screenshot of the system interface including the relevance scale. The prototype system retrieved a ranked list of XML elements. Any element chosen for display was placed in the context of the containing document by showing its position in a table of contents. To allow the test persons to interact as naturally as possible, they were free to choose any element from the ranked list and to browse within the documents as they saw fit. The test persons were, however, instructed to assess viewed elements, but not forced to.

However, this method of collecting assessments also presented some drawbacks. It did not guarantee, for example, that test persons would provide assessments for every single element they viewed; it was possible for them to leave a viewed page without providing any assessments. Unassessed elements were viewed as providing an indication of non-relevance. However, there is no tangible evidence to suggest that this is always the case. Further, although the test persons provided a quantitative indication of relevance, they did not provide a qualitative one, i.e., why was a certain element too specific or too exhaustive, or why was a certain element not relevant at all? This kind of qualitative data was not captured explicitly in the experimental set up, but was mostly inferred by the logs of the search sessions, the time stamp data, etc.

Very few of the test persons communicated difficulties in understanding or using the 10 point relevance scale. Nevertheless, the results of an initial analysis of the collected assessments indicate that the test persons may have had such difficulties as parts of the scale were underused [8]. In addition, only 60% of the viewed components were assessed [10] and there are qualitative comments in the questionnaire data indicating that some test persons were tired of having to assess every viewed element.

The next section lists a number of alternatives and discusses the advantages and limitations of each.

3. ALTERNATIVES

A first alternative would be *not* to ask the test persons to assess the documents at all, and use the relevance assessments from the ad hoc track instead. This approach would provide easy access to already available relevance assessments, and would impose minimum strain on the test persons. On the other hand, such an approach is fundamentally opposed to the very idea of interaction and of simulated work-task situations; the subjective notion of relevance is disregarded in this approach.

A second alternative would be to use implicit indicators of relevance (as opposed to explicitly indicating relevance by means of a quantitative scale). Implicit indicators can include the time spent viewing an element, the amount of scrolling involved, etc. [12]. This approach would also impose a minimum strain on the test persons as indicators of relevance would stem from the way the persons interact with elements. However, an inherent difficulty with relating relevance to implicit indicators, is that there is no unambiguous evidence that the behaviour indeed suggests relevance. For example, a test person may choose to spend longer time reading the contents of an element because he finds this element difficult to comprehend, and not necessarily because he finds it relevant. Further, it is also difficult to correlate implicit indicators with certain levels of exhaustiveness or specificity.

Considering some more concrete indicators of relevance based on user behaviour is also possible. For example, test persons may be asked to bookmark elements that are of interest. This approach is also cognitively easy on test persons, as the act of bookmarking is rather natural during information seeking tasks nowadays. In addition, it allows us to consider in detail fewer elements for further analysis, i.e. to focus on the elements that test persons found interesting. However, this approach would present us with a large fraction of viewed elements for which no data is available.

Alternatively, if every viewed element is to be assessed the act of doing so should be made as straightforward and easily comprehensible to the test persons as possible because of the associated cognitive load and risk of obtrusiveness. A complex relevance scale, such as those used in the ad hoc and 2004 interactive track, or the assessment of several aspects for each interaction work against this. Rather a simple scale gauging a

[2] *Exhaustiveness* was renamed *Usefulness*, but the same definition was used in the instructions for test persons.

single aspect or one dimension should be employed. A simple scale will allow the test persons to complete their assessments without much delay, and have been successfully implemented in interactive IR experiments in the past (See e.g. [2]). A limitation both with bookmarking and the use of a simple scale is that data about why and how each element is relevant would still not be made available by the test persons.

It is possible to obtain explicit accounts of why elements were assessed at a certain relevance level through the use of more sophisticated equipment and experimental techniques. For example, it is possible to use eye tracking equipment to monitor the test persons' eye movements while reading the contents of elements. By analysing fixation periods and saccades, it is possible to make inferences about the test persons' perception of importance of the various elements. This can be combined with a structured interview after the search session, in which the test persons will elaborate during a replay of the session why certain decisions were made [4]. Such 'talk-after interviews' can also be carried out with less expensive on-screen video capturing software [11]. Alternatively, think-alouds could also be employed during the search sessions in order to capture the reasons for the test persons' assessments. These approaches have the advantage that they enable us to document why test persons assess certain elements at a specific relevance level. However, the need for specialised equipment and for more laborious experimental techniques (e.g. analysing structured interviews) may present some practical challenges in implementing this approach.

4. CONCLUSIONS

In order to answer some of the important fundamental questions in XML IR, a wide range of aspects should ideally be assessed at each interaction with the test system. This would, however, prevent the tests persons from interacting naturally with the system, and thus undermine the purpose of an interactive study.

Therefore, different alternatives were discussed. A common thread in these is the challenge of finding a method that can inform us why something is relevant or not-relevant, while at the same time not being obtrusive enough to obscure the browsing and searching behaviour of the test persons.

We do not regard complex relevance scales such as those employed in the ad hoc track or a requirement to judge several aspects for each viewed element as fruitful in an interactive setting. Instead, bookmarking or the use of a simple scale should be used to minimise the cognitive load on the test persons and allow a searching behaviour that is as natural as possible. This in combination with eye-tracking or desktop video approaches may help answer some of the important research questions in XML IR by allowing the collection of data that can inform us not only about *what* but also *why* test persons may find elements relevant.

5. ACKNOWLEDGMENTS

We wish to thank the Danish HCI Forum for a stimulating discussion that led to the idea for this paper, and Gabriella Kazai for fruitful debates about ideal relevance assessments.

6. REFERENCES

1. Beaulieu, M. (1997): Experiments on interfaces to support query expansion. *Journal of Documentation*, 53(1), p. 8-19. (Special issue on Okapi)
2. Borlund, P. (2000): *Evaluation of interactive information retrieval systems*. Åbo: Åbo Akademi University Press, 276 p. (PhD dissertation)
3. Fuhr, N., Lalmas, M. and Malik, S. (2005): *Advances in XML Information Retrieval: Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004, Dagstuhl Castle, Germany, December 6-8, 2004, Revised Selected Papers*. Berlin: Springer. (LNCS ; 3493)
4. Hansen, J. Paulin. (1991): The use of eye mark recordings to support verbal retrospection in software testing. *Acta Pcychologica*. 76, pp. 31 - 49
5. Järvelin, K. and Kekäläinen, J. (2002): Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4), 422-446.
6. Kazai, G., Lalmas, M. and de Vries, A. P. (2004): The overlap problem in content-oriented XML retrieval evaluation. In: *Proceedings of SIGIR 2004*, p. 72-79.
7. Kekäläinen, J. and Järvelin, K. (2002): Using graded relevance assessments in IR evaluation. *Journal of the American Society for Information Science and Technology*, 53(13), 1120-1129.
8. Pharo, N. and Nordlie, R. (2005): Context matters – an analysis of assessments of XML documents. In: *Proceedings of CoLIS5*, p. 238 – 248. (LNCS 3507)
9. Sormunen, E. (2002): Liberal relevance criteria of TREC : counting on negligible documents? In: *Proceedings of SIGIR 2002*, p. 324-330.
10. Tombros, A., Larsen, B. and Malik, S. (2005): The Interactive Track at INEX 2004. In: Fuhr, N., Lalmas, M. and Malik, S. eds. *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.* Berlin: Springer, p. 410-423. (Lecture Notes in Computer Science ; 3493)
11. Toms, E. G., O'Brian, H. L., Kopak, R. & and Freund, L. (2005): Searching for relevance in the relevance of search. In: *Proceedings of CoLIS5*, p. 59 – 78. (LNCS 3507)
12. White, R.W., Ruthven, I., Jose, J.M. (2002). Finding relevant documents using top ranking sentences: an evaluation of two alternative schemes. In: *Proceedings of SIGIR 2002*, pp. 57-64.

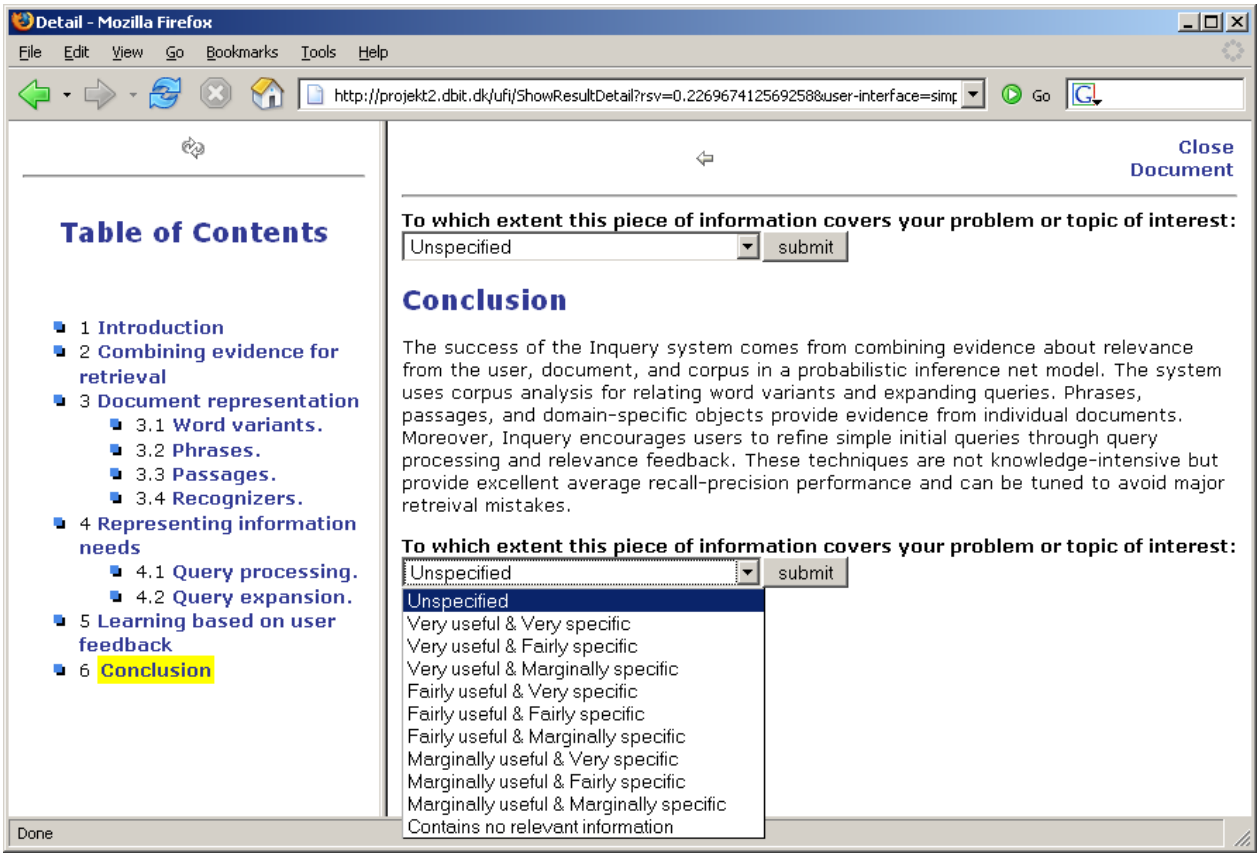


Figure 1. The HyREX XML IR system with prototype interface as used in the INEX 2004 interactive track [see 10]. Detailed component view containing the full text of the component and a table of contents for the whole document, and showing the relevance assessment scale.

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

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

ABSTRACT

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

Keywords

Information Retrieval, Heterogeneous Search, XML Element Retrieval

1. INTRODUCTION

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

1.1 Heterogenous Collections Track Motivation

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

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

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

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

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

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

1. Creation of a heterogeneous test collection.

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

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

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

1.2 Testbed creation

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

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

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

1.3 Retrieval experiments

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

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

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

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

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

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

2. LESSONS AND ISSUES FROM THE 2004 HETEROGENEOUS TRACK

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

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

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

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

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

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

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

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

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

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

3. DISCUSSION AND CONCLUSION

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

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

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

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

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

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

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

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

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

4. REFERENCES

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

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

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

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

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

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

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

Users and Assessors in the Context of INEX: Are Relevance Dimensions Relevant?

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

James A. Thom
School of CS and IT
RMIT University
Melbourne, Australia
jat@cs.rmit.edu.au

Anne-Marie Vercoustre
AxIS research group
INRIA
Rocquencourt, France
anne-marie.vercoustre@inria.fr

ABSTRACT

The main aspects of XML retrieval are identified by analysing and comparing the following two behaviours: the behaviour of the assessor when judging the relevance of returned document components; and the behaviour of users when interacting with components of XML documents. We argue that the two INEX relevance dimensions, Exhaustivity and Specificity, are not orthogonal dimensions; indeed, an empirical analysis of each dimension reveals that the grades of the two dimensions are correlated to each other. By analysing the level of agreement between the assessor and the users, we aim at identifying the best units of retrieval. The results of our analysis show that the highest level of agreement is on highly relevant and on non-relevant document components, suggesting that only the end points of the INEX 10-point relevance scale are perceived in the same way by both the assessor and the users. We propose a new definition of relevance for XML retrieval and argue that its corresponding relevance scale would be a better choice for INEX.

1. INTRODUCTION

The INitiative for the Evaluation of XML retrieval[1] (INEX) is a coordinated effort that promotes evaluation procedures for content-oriented XML retrieval. In order to evaluate XML retrieval effectiveness, the concept of *relevance* needs to be clearly defined. There are two *relevance dimensions* used by INEX, *Exhaustivity* and *Specificity*, which measure the extent to which a given information unit *covers* and is *focused on* an information need, respectively [16]. In this paper we provide a detailed empirical analysis of the two INEX relevance dimensions. More specifically, we investigate what the experience of both the assessor and the users suggests on how relevance should be defined and measured in the context of XML retrieval.

The INEX test collection consists of three parts: an XML document collection, a set of topics required to search for information stored in this collection, and a set of relevance assessments that correspond to these topics [12]. The XML document collection comprises 12,107 IEEE Computer Society articles published in the period between 1997-2002, with approximately 500MB of data. To search for information stored in this collection, two types of topics are explored in INEX: Content-Only (CO) topics and Content-And-Structure (CAS) topics. CO topics do not refer to the existing document structure, whereas CAS topics enforce restrictions on the document structure and explicitly specify the target element. In this paper, we focus on the CO topics to analyse the behaviour of the assessor and the users in the context of INEX.

Tombros et al. [20] demonstrate that, while assessing relevance of retrieved pages on the Web, the context determined by a task type has an effect on the user behaviour. A similar effect is likely to be expected when users assess the relevance of XML document components (rather than of whole documents, such as Web pages) [19]. The CO topics used in this study are thus selected such that they correspond to different types of tasks, or different *topic categories*: a *Background* category and a *Comparison* category.

Since 2002, a new set of topics has been introduced and assessed by INEX participants each year. Analysing the behaviour of assessors when judging the relevance of returned document components may provide insight into the possible trends within the relevance judgements. Such studies have been done for both the INEX 2002 [9] and the INEX 2003 [16] test collections. We have recently also analysed the relevance judgements of the INEX 2004 topics, where we aimed at understanding what assessors consider to be the most useful answers [14].

There is growing interest among the research community in studying the user behaviour in the context of XML retrieval; however, little work has been done in the field so far. The most notable is the work done by Finesilver and Reid [4], where a small-scale experimental study is designed to investigate the information-seeking behaviour of users in the context of structured documents. Recently, an Interactive track was established at INEX 2004 to investigate the retrieval behaviour of users when components of XML documents – estimated as likely to be relevant by an XML retrieval system – are presented as answers [19]. Ten of the 43 active research groups in INEX 2004 were also involved in the Interactive track, and each group was required to provide a minimum of eight users to interact with the retrieval system. The analysis of the user behaviour in this paper is based on the user judgements provided by these groups.

When judging the relevance of a document component, two relevance dimensions – *Exhaustivity* and *Specificity* – are used by INEX. Each dimension uses four grades of relevance.

[1] `http://inex.is.informatik.uni-duisburg.de/2005/`

To assign a relevance score to a document component, the grades from each dimension are combined into a single 10-point relevance scale. However, the latter choice of combining the grades poses the following question: *is the 10-point relevance scale well perceived by users*?

Due to hierarchical relationships between the XML document components, an XML retrieval system may often return components with varying granularity. The problem that often arises in this retrieval scenario is the one of distinguishing the *appropriate level of retrieval granularity*. This problem, which is often referred to as the *overlap problem*, remains an open research problem in the field of XML retrieval. Indeed, it has been shown that it is not only a retrieval problem [14, 15], but also a serious evaluation problem [8]. This then raises the question: *is retrieving overlapping document components what users really want*?

In this work, we aim at finding answers to the above research questions. We show that the overlap problem is handled differently by the assessor and the users, and that the two INEX relevance dimensions are perceived as one. We propose a new definition of relevance for INEX and argue that its corresponding relevance scale would bring a better value for the XML retrieval evaluation.

The remainder of the paper is organised as follows. In Section 2 we provide an overview of the methodology used in this study. The concept of *relevance* in information retrieval is thoroughly discussed in Section 3, where we particularly focus on how the INEX definition of relevance fits in the unified relevance framework. We study the behaviour of the assessor and the users in Sections 4 and 5, when two categories of retrieval topics are considered, respectively. Our new definition of relevance is described in Section 6. We conclude in Section 7 with a brief discussion of our findings.

2. METHODOLOGY

In this section, we provide a detailed overview of the methodology used in this study. More precisely, we describe the type and the number of participants involved; the choice of the two categories of topics used; and the way the data – reflecting the observed behaviour of participants – was collected. The data reflecting the observed behaviour, as analysed in this study, was collected from well-established INEX activities, which are also explained in separate studies. For instance, for a particular CO topic we use the relevance judgements obtained from the interactive online assessment system [16] to analyse the behaviour of the assessor. Similarly, for the same topic we use the data collected for the purposes of the INEX 2004 Interactive track [19] to analyse the retrieval behaviour of users. We actively participated in both INEX activities.

2.1 Participants

Two types of participants are used in this study: assessors and users. In general, both can be regarded as users; however, it is often necessary to distinguish between them, since their purpose in the XML retrieval task is quite different.

Assessors

Every year since 2002 when INEX started, each participant is asked to submit at least one retrieval topic (query). If a topic is accepted, the same participant is (usually) required to assess the relevance of the retrieved document components. The assessor can, therefore, be seen as an entity that provides the ground-truth for a particular retrieval topic. There is usually one assessor per topic, although for the purpose of checking whether the relevance judgements were done in a consistent manner, two or more assessors may be assigned to a given topic [16]. In this study we analyse the relevance assessments provided by one assessor per topic.

```
Topic B1 (INEX 2004 CO topic 192):

You are writing a large article discussing virtual reality
(VR) applications and you need to discuss their negative
side effects. What you want to know is the symptoms
associated with cybersickness, the amount of users who get
them, and the VR situations where they occur. You are not
interested in the use of VR in therapeutic treatments
unless they discuss VR side effects.
```

Figure 1: A *Background* topic example.

Users

A total of 88 users were employed for the purposes of the Interactive track at INEX 2004, with an average age of 29 years [19]. Although most of the users had a substantial level of experience in Web or other related searches, it was expected that very few (if any) were experienced in interacting with XML document components. For this purpose, users were given the same (or rather, slightly modified) retrieval topics as the ones proposed and judged by the assessors. Analysing the data collected from the user interaction may thus indicate how well an XML retrieval system succeeds in satisfying users' information needs. Our analysis in this study is based on the user judgements provided by roughly 50 users per topic.

2.2 Retrieval Topics

To make users better understand the objectives of the retrieval task, the CO topics were reformulated as simulated work task situations [19]. A simulated work task situation requires users to interact with the retrieval system, which in turn – by allowing users to formulate as many queries as needed – results in different individual interpretations of the information need [2]. Thus, the reformulated CO topics not only describe *what* the information need represents, but also *why* users need to satisfy this need, and what is the *context* where the information need arises.

The CO topics used in the INEX Interactive track are divided in two task categories: a *Background* category and a *Comparison* category. Topics that belong to the *Background* category seek to find as much general information about the area of interest as possible. Two retrieval topics were used in this category, B1 and B2, which are based on the INEX 2004 CO topics 192 and 180, respectively [19]. Figure 1 shows Topic B1, which is the *Background* topic used in this study. Topics that belong to the *Comparison* category seek to find similarities or differences between at least two items discussed in the topic. Two retrieval topics were used in this category, C1 and C2, which are respectively based on

```
Topic C2 (INEX 2004 CO topic 198):

You are working on a project to develop a next generation
version of a software system. You are trying to decide on
the benefits and problems of implementation in a number
of programming languages, but particularly Java and Python.
You would like a good comparison of these for application
development. You would like to see comparisons of Python
and Java for developing large applications. You want to
see articles, or parts of articles, that discuss the
positive and negative aspects of the languages. Things
that discuss either language with respect to application
development may be also partially useful to you. Ideally,
you would be looking for items that are discussing both
efficiency of development and efficiency of execution
time for applications.
```

Figure 2: A *Comparison* topic example.

the INEX 2004 CO topics 188 and 198 [19]. Figure 2 shows Topic C2, which is the *Comparison* topic used in this study.

The motivation of using topics B1 and C2 in our study comes from the fact that both of these topics have corresponding relevance judgements available, and that data from roughly 50 users was collected for each of these topics. In contrast, no relevance judgements are available for topic B2, while data from around 18 users was collected for each of the topics B2 and C1. Previous work has also shown that XML retrieval systems exhibit varying behaviour when their performance is evaluated against different CO topic categories [7, 15]. It is then reasonable to expect that the level of agreement between the assessor and the users, which concerns the choice of the best units of retrieval, may depend on the topic category. Thus, in our forthcoming analysis of the retrieval behaviour, we clearly distinguish between topics B1 and C2.

2.3 Collecting the Data

Different means were used to collect the data from the assessor and the users, and different time restrictions were put in place in both cases.

In the case of the assessor, an interactive online assessment system is used to collect the judgements for a particular topic [16]. This is a well-established method used in INEX, where the assessment system implements some rules to ensure that the collected relevance judgements are as exhaustive and as consistent as possible. On average it takes one week for the assessor to judge all the retrieved elements for a particular topic. The relevance judgements are then stored in an *XML assessment file* where, for each XML document retrieved by participant systems, the judged elements are kept in document order. We use two assessment files, one for each topic B1 and C2, to analyse the relevance judgements made by assessors.

For users, a system based on HyREX [6] is used to collect the user judgements and to log their activities. Tombros et al. [19] explain the process of user interaction with the HyREX system in detail. Users are able to choose between two retrieval topics for each topic category, for which they are required to find as much information as possible for completing the search task. A time limit of 30 minutes is given to each user. The data obtained from the user interaction is stored in corresponding log files. For each user, we create an assessment file that follows the same structure as the assessor's assessment file. We use these files to analyse the judgements made by users for each of the topics B1 and C2.

An important point to note is that HyREX uses the concept of "index objects" [6] to limit the level of retrieval granularity that will be returned to users. This means that users were able to make judgements for only four (out of 192) element names. These names are `article`, `sec`, `ss1`, and `ss2`, which correspond to full article and to section and subsection elements of varying nesting levels, respectively. Although this may be seen as a limitation of the HyREX system, the obtained element granularity is nevertheless sufficient for the purpose of our analysis. To be consistent in our comparison of the observed behaviour between the assessor and the users, all element names different from these four were also removed from the two files containing assessors' judgements. If an element has been judged more than once, either by a user or an assessor, only the last relevance judgement is stored in the assessment files.

2.4 Measuring Overlap

When collecting assessor or user judgements for a particular topic, we also measure the level of overlap between the judged elements. There are *at least* two ways by which the overlap can be measured:

- *set-based overlap*, which for a *set* of returned elements measures the percentage of elements for which there exists another element that *fully contains* them; and
- *list-based overlap*, which takes into account the order of processing of returned elements, and measures the percentage of elements for which there exists another element *higher in the list* that fully contains them.

Consider the following set of returned elements:

1. `/article[1]/sec[1]`
2. `/article[1]/sec[1]/ss1[1]`
3. `/article[1]/sec[1]/ss1[1]/ss2[1]`
4. `/article[1]/sec[2]/ss1[1]`
5. `/article[1]/sec[2]`

Let us assume that the elements are returned in the above order, and that all the elements belong to one XML document. The set-based overlap in this case would be 60%, because three (out of five) elements in this set are fully contained by other element in the set (the three elements are the ones belonging to ranks 2, 3 and 4). The list-based overlap, however, would be 40%, because there are only two elements for which there exists another element higher in the list that fully contains them (the two elements that belong to ranks 2 and 3).

In this study we use the set-based overlap, as defined above, to measure the overlap between the judged elements. How-

ever, unlike in the assessor's case where the relevance judgements were obtained from only one assessor, the user judgements for a given topic were obtained from more than one user. To deal with this issue in a consistent manner, in users' case we measure the overlap *separately for each user*, and take the average to represent the resulting set-based overlap.

3. RELEVANCE: DEFINITIONS AND DIMENSIONS

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 [13, 17]. Indeed, the main purpose of a retrieval system is to retrieve units of information estimated as *likely to be relevant* to an information need, as represented by a query. To build and evaluate effective information retrieval systems, the concept of relevance needs to be clearly defined and formalised.

Mizzaro [13] provides an overview of different definitions of relevance. These are also conveniently summarised by Lavrenko [10]. In general, there is a system-oriented, a user-oriented, and a logical definition of relevance. However, there are also other definitions of relevance, which relate to its nature and the notion of dependence. With respect to its nature, there is a binary or non-binary (graded) relevance. With respect to whether the relevance of a retrieved unit is dependent or not on any other unit already inspected by the user, there is a dependent or independent relevance. In the case of the former, the relevance is often distinguished either as a relevance conditional to a set of relevant retrieved units, or as a novel relevance, or as an aspect relevance.

In the following we provide an overview of several definitions of relevance, including the INEX relevance definition. We then describe a notable attempt to construct a unified definition of relevance [13].

3.1 System-oriented Relevance Definition

The system-oriented definition provides a binary relation between a unit of information (a document or a document component) and a user request (a query). To model this relation, both the unit of information and the user request are represented by a set of terms, reflecting the contents of the unit and the interest of the user, respectively. In this case, relevance is simply defined by the level of semantic overlap between the two representations; the more similar these representations are, the more likely the information unit is relevant to the user request. According to this definition, relevance is not dependent on any factors other than the two representations above. More precisely, it depends neither on the user who issued the request (or on the user information need, for that matter), nor on any other information units (regardless of whether they have been previously considered to be relevant or not), nor on any other requests to which the unit of information may or may not be relevant.

3.2 Novel Relevance

Novel relevance deals with the impact of retrieving redundant information units on user's perception of relevance. For example, if a system retrieves two near-duplicate information units, which may both be relevant to a request, the user will very likely not be interested in reading both of them, since once the first one is read, the second becomes entirely redundant. Carbonell and Goldstein proposed the concept of *Maximal Marginal Relevance* [3], which attempts to provide a balance between the relevance of a document to a query, and the redundancy of that document with respect to all the other documents previously inspected by the user. An interesting approach that may be seen as an extension of the above work was proposed by Allan et al. [1]. Their work attempts to address redundancy on a sub-document level and is based on the following idea: even if a document is considered to be mostly redundant by a user, it may still contain a small amount of novel information (which is, for example, often the case in news reporting). Therefore, they independently evaluate the performance of an information retrieval system with respect to two separate definitions of relevance: a topical relevance and a novel relevance. We believe that this (or a similar) approach is particularly attractive for the field of XML retrieval, where systems tend to retrieve mutually overlapping (and thus redundant) information units. Some aspects of novel relevance are investigated in detail by the TREC Novelty track [18].

3.3 Aspect Relevance

A user request often represents a complex information need that may comprise smaller (and possibly independent) parts, often called *aspects*. The goal of an information retrieval system is then to retrieve information units that cover as many aspects of the information need as possible. In this context, *aspect relevance* is defined as topical relevance of the retrieved unit to a particular aspect of the information need, whereas *aspect coverage* is defined as the number of aspects for which relevant retrieved units exist. Zhai [22] describes a formal approach to modelling aspect relevance. INEX uses a somewhat modified definition of aspect relevance, which will be discussed in more detail below.

3.4 The INEX Relevance Definition

From 2003 in INEX, the relevance of an information unit (a document or a document component) to a request (a query) is described by two dimensions: *Exhaustivity*, which represents topical relevance that models the extent to which the information unit discusses aspects of the information need represented by the request, and *Specificity*, which also represents topical relevance, but models the extent to which the information unit focuses on aspects of the information need. For example, an information unit may be highly exhaustive to a user request (since it discusses most or all the aspects of the information need), but only marginally specific (since it also focuses on aspects other than those concerning the information need). Conversely, an information unit may be highly specific to a user request (since there is no non-relevant information and it only focuses on aspects concerning the information need), but it may be marginally exhaustive (since it discusses only a few aspects of the information need).

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[2]. The rele-

[2]Recent Robust and Web tracks in TREC, however, use a non-binary relevance scale for evaluation.

Specific	Exhaustive			
	Highly	*Fairly*	*Marginally*	*None*
Highly	**E3S3**	**E2S3**	**E1S3**	**E0S0**
Fairly	**E3S2**	**E2S2**	**E1S2**	**E0S0**
Marginally	**E3S1**	**E2S1**	**E1S1**	**E0S0**
None	**E0S0**	**E0S0**	**E0S0**	**E0S0**

Table 1: The 10-point relevance scale, as adopted by INEX. Each point of the relevance scale combines a particular grade from the Exhaustivity dimension with a corresponding grade from the Specificity dimension.

vance 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). INEX, however, adopts a four-graded relevance scale for each of the relevance dimensions, such that the relevance of an information unit to a request ranges from none, to marginally, to fairly, or to highly exhaustive or specific, respectively. To identify *relevant* units of information, that is, units of information that are both exhaustive and specific to a user request, a combination of the grades from each of the two relevance dimensions is used. These relevant units are then, according to INEX, "the most appropriate units of information to return as an answer to the query" [16]. Table 1 shows the combination of the grades from each of the two relevance dimensions, which represents the 10-point relevance scale used by INEX.

The two relevance dimensions, *Exhaustivity* and *Specificity*, are not completely independent. An information unit that is not exhaustive is at the same time not specific to the request (and vice versa), which restricts the space of combining the grades of the two dimensions to ten possible values. In the remainder of the paper, a relevance value of an information unit to a request will be denoted as `EeSs`, where `E` represents *Exhaustivity*, `S` represents *Specificity*, and `e` and `s` represent integer numbers between zero and three. For example, `E1S3` represents an information unit that is marginally exhaustive and highly specific to a request. An information unit is considered relevant only if both `e` and `s` are greater than zero. The relevance value `E0S0` therefore denotes a non-relevant information unit, whereas the value `E3S3` denotes a highly relevant information unit.

Comparison with Aspect Relevance

A strong parallel may be drawn between *Exhaustivity* and *Specificity*, the two INEX relevance dimensions, with *aspect coverage* and *aspect relevance*. *Exhaustivity* maps the aspect coverage to a four-point relevance scale, from `E0` being "the XML element does not discuss the query at all" [16], to `E3` being "the XML element discusses most or all aspects of the query" [16]. *Specificity*, on the other hand, is almost identical to aspect relevance.

3.5 Unified Relevance Definition

A notable attempt to construct a unified definition of relevance is given by Mizzaro [13]. He formalises a framework capable of modelling various definitions of relevance by embedding it in a four-dimensional space.

The first dimension deals with the type of entities for which the relevance is defined. It can take one of the following three values: *Document*, *Surrogate*, or *Information*. *Document* refers to the information unit a user will obtain as a result of their search; this may represent a full-text document, an image, video, or, in the case of XML retrieval, a document component. *Surrogate* refers to a form of representation of *Document*; this may be of a set of terms, bibliographic data, or a condensed abstract of the information unit. The third value, *Information*, refers to a rather abstract concept, which depends on the type and amount of information the user receives while reading or consuming the contents of the returned unit of information.

The second dimension relates to the level at which the user request is dealt with. There are four possible levels: *Problem*, *Information need*, *Request*, or *Query*. The *Problem* (also referred to as Real Information Need – RIN [10]) relates to the actual problem that a user is faced with, and for which information is needed to help solve it. The user may not be fully aware of the actual problem; instead, in their minds they perceive it by forming a mental image. This mental image in fact represents the *Information need* (also referred to as Perceived Information Need – PIN [10]). *Request* is a way of communicating the *Information need* to others by specifying it in a natural language. For the *Request* to be recognised by a retrieval system, it needs to be represented by a *Query*. The *Query* usually consists of a set of terms, optionally including phrases or logical query operators.

Relevance can then simply be seen as a combination of any of the entities from the two dimensions above; that is, it can be seen as a combination of any of the *values* from the first dimension with any of the *levels* from the second dimension. Indeed, phrases such as "relevance of a *Surrogate* to a *Query*" or "relevance of a *Document* to a *Request*" are often used. Mizzaro, however, argues that this relevance space does not actually represent the space of all possible relevances. Rather, there is also a third dimension that specifies the nature of the relationship between the two dimensions. The components of this third dimension are *Topic*, *Task*, *Context*, or any combination of the three. The *Topic* (or topical relevance [10]) specifies how similar the two entities are to user's area of interest. For example, if the user is interested in finding information about the overlap problem in XML retrieval, the topical relevance will represent the level of similarity of the retrieved unit to the query with respect to that particular area of interest. The *Task* (or task relevance [10]) specifies the level of usefulness of the information found in an entity for the actual task performed by the user (for example, writing a paper or preparing a lecture). The final component, *Context*, includes everything that is not previously covered by *Topic* and *Task*, but which nevertheless affects the whole process of retrieval (such as search costs, or the amount of novel information found, or anything else).

Since the information seeking process may evolve in time, a fourth dimension, *Time*, is needed to model the fact that users often change their perception of the information they seek to find. For example, at a certain point in time an information unit (*Surrogate* or *Document*) may not be rel-

evant to a user request (*Query* or *Request*), however due to the evolving nature of the seeking process the user may learn something that would permit them to understand the content of the unit, which, in turn, may make the same unit relevant to the request.

A definition of relevance can, therefore, be seen as a point in the above four-dimensional space. Mizzaro [13] argues that the above framework can be used to model and compare different definitions of relevance. For example, the following expression may be used to model the system-oriented definition of relevance described in Section 3.1: *Topical relevance* of a *Surrogate* to a *Query* at a certain point in *Time* (the time when the request was formulated as a query and submitted to the retrieval system). However, finding an expression that may be used to model the INEX definition of relevance turns out to be quite a challenging task. The main problem is that both the INEX relevance dimensions, *Exhaustivity* and *Specificity*, are based on topical relevance, which corresponds to the *Topic* component of the third relevance dimension in the unified framework. We contend that one relevance dimension based on topical relevance should be used, or possibly two orthogonal dimensions that correspond to different components of the above framework. In Section 6 we propose a much simpler definition of relevance, and argue that its corresponding relevance scale would be a better choice for INEX.

4. BEHAVIOUR ANALYSIS FOR BACKGROUND TOPICS

In this section, we separately analyse the assessor's and users' behaviour when judging the relevance of returned elements for the *Background* topic B1. In order to identify the best retrieval elements for this topic, we also analyse and compare the level of agreement between the assessor and the users.

4.1 Analysis of Assessor's Behaviour

Figure 3 shows an analysis of the relevance judgements for topic B1 (the INEX 2004 CO topic 192) that were obtained from one assessor. As shown in the figure, we use only four element names in our analysis: `article`, `sec`, `ss1`, and `ss2`. The x-axis contains the 9-point relevance scale which is a result of combining the grades of the two INEX relevance dimensions (the case `E0S0` is not shown). The y-axis contains the number of occurrences of relevant elements for each point of the relevance scale. For a relevance point, the number of occurrences of each of the four element names is also shown.

The total number of relevant elements for topic B1 is 32. Of these, 11 elements have been judged as `E2S1`, nine as `E1S1`, six as `E2S3`, two as `E3S3` or `E2S2`, and one as `E3S1` or `E1S2`. Interestingly, none of the relevant elements have been judged as either `E3S2` or `E1S3`. The number of occurrences of the four element names is as follows. The `sec` elements occur most frequently with 18 occurrences, followed by `article` with ten, `ss1` with three, and `ss2` with one occurrence, respectively. The total number of elements that have been judged as non-relevant (`E0S0`) for topic B1 is 1158, of which 513 are `sec` elements, 411 are `ss1`, 186 are `article`, and 48 are `ss2` elements.

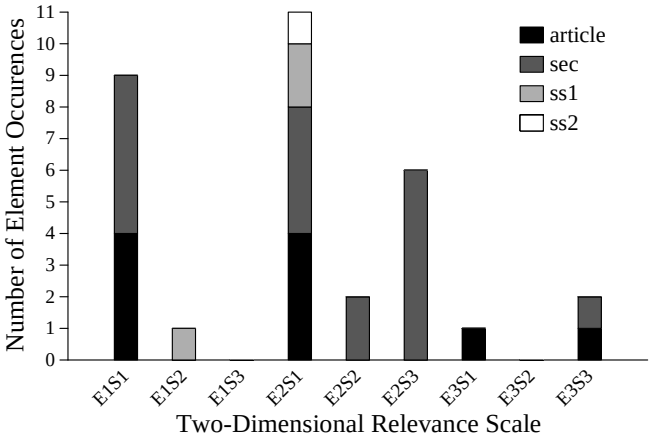


Figure 3: Analysis of assessor's behaviour for topic B1. For each point of the relevance scale, the figure shows the total number of relevant elements, and the number of relevant elements for each of the element names.

Level of Overlap

The above statistics show that the `E2S1` and `E1S1` points of the relevance scale contain around 63% of the relevant elements for topic B1. Moreover, further analysis reveals that there is a substantial amount of overlap among these elements. More precisely, there is 64% set-based overlap among the 11 `E2S1` elements, where the four `article` elements contain all of the section and sub-section elements. Similarly, there is 56% overlap among `E1S1` elements, where of nine elements, four `article` elements contain the other five `sec` elements. Interestingly, the other points of the relevance scale do not suffer from overlap. The two highly relevant elements (`E3S3`), for example, belong to different XML files.

Correlation between Relevance Grades

In the following we investigate the correlation between the grades of the two relevance dimensions for topic B1. We want to check whether, while judging relevant elements, the assessor's choice of combining the grades of the two relevance dimensions is influenced by a common aspect [9].

The top half of Table 2 shows the correlation between the grades of the two relevance dimensions for topic B1, as judged by the assessor. For each grade of the *Exhaustivity* relevance dimension (columns), the value of `Sp|Ex` shows the percentage of the cases where an element is judged as `Sp` (specific), given that it has already been judged as `Ex` (exhaustive). Similarly, for each grade of the *Specificity* relevance dimension (rows), the value of `Ex|Sp` shows the percentage of the cases where an element is judged as `Ex` (exhaustive), given that it has already been judged as `Sp` (specific). For example, the `Sp|Ex` value of column `E3` and row `S3` is 66.67, indicating that in 66.67% of the cases a highly exhaustive element is also judged as highly specific. We now analyse the correlation between the grades of each separate relevance dimension.

For *Exhaustivity*, we observe that in 90% of the cases a marginally exhaustive (`E1`) element is also judged as margi-

Assessor:	Specificity	Exhaustivity E3 Sp|Ex (%)	E3 Ex|Sp (%)	E2 Sp|Ex (%)	E2 Ex|Sp (%)	E1 Sp|Ex (%)	E1 Ex|Sp (%)
	S3	**66.67**	25.00	31.57	*75.00*	0.00	0.00
	S2	0.00	0.00	10.53	*66.67*	10.00	33.33
	S1	33.33	4.62	**57.90**	*52.38*	**90.00**	43.00

Users:	Specificity	Exhaustivity E3 Sp|Ex (%)	E3 Ex|Sp (%)	E2 Sp|Ex (%)	E2 Ex|Sp (%)	E1 Sp|Ex (%)	E1 Ex|Sp (%)
	S3	**69.62**	*69.62*	36.84	22.15	12.26	8.23
	S2	27.22	*41.34*	**40.00**	36.54	21.70	22.12
	S1	3.16	5.16	23.16	22.68	**66.04**	*72.16*

Table 2: Correlation between the grades of the two relevance dimensions for topic B1, as judged by both the assessor and the users. Depending on the relevance dimension, the highest correlation of each grade is shown either in bold (for Exhaustivity) or italics (for Specificity).

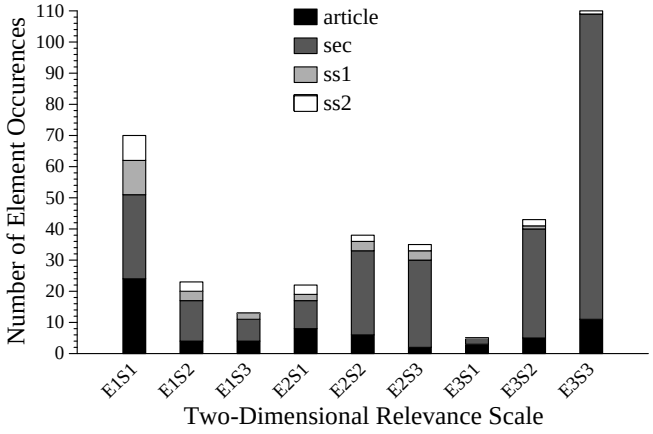


Figure 4: Analysis of users' behaviour for topic B1. For each point of the relevance scale, the figure shows the total number of relevant elements, and the number of relevant elements for each of the element names.

nally specific (*S1*). This is somehow intuitive, since by definition a marginally exhaustive element discusses only a few aspects of the information need, so its focus may be on aspects other than those concerning the information need. However, for topic B1, the number of `E1` elements is around 30% of the total number of relevant elements, so the above correlation should be treated carefully. In contrast, the number of fairly exhaustive elements (`E2`) is around 60% of the total number of relevant elements, and in 58% of the cases a fairly exhaustive element is (again) judged as *S1*. For highly exhaustive (`E3`) elements, we find that in 67% of the cases an `E3` element is also judged as highly specific (*S3*), although the number of `E3` elements is very low (only 10% of the total number of relevant elements).

For *Specificity*, the number of marginally specific (*S1*) elements is around 66% of the total number of relevant elements, where in 52% of the cases an *S1* element is judged as fairly exhaustive (`E2`), while in 43% of the cases it is judged as marginally exhaustive (`E1`). Fairly specific (*S2*) elements are 9% of the total number of relevant elements, and in 67% of the cases an *S2* element is judged as `E2`. Finally, in 75% of the cases a highly specific (*S3*) element is (again) judged as `E2`, although the number of highly specific elements is around 25% of the total number of relevant elements.

4.2 Analysis of Users' Behaviour

Figure 4 shows the relevance judgements for topic B1 that were obtained from 50 users. Unlike in the assessor's case, an element may have been judged by more than one user, so each relevance point in Figure 4 may contain multiple occurrences of a given element.

The total number of occurrences of relevant elements for topic B1 is 359. Around 61% of this number are elements that have been judged either as `E3S3` (110), `E1S1` (70), or `E2S2` (38). All the 10 points of the relevance scale were used by users. However, different number of users have judged elements for each relevance point. For example, 41 (out of 50) users have judged at least one element as `E3S3`, whereas this number is 35 for `E1S1`, 23 for `E2S2`, and 20 and below for the other points of the relevance scale. The `sec` elements occur most frequently with 246 occurrences, followed by `article` with 67, `ss1` with 25, and `ss2` with 21 occurrences, respectively. The total number of element occurrences judged as non-relevant (`E0S0`) for topic B1 is 181, of which 80 are `sec` elements, 72 are `article`, 26 are `ss1`, and only 3 are `ss2` elements. Also, 39 (out of 50) users have judged at least one element as `E0S0`.

Level of Overlap

A more detailed analysis of the user judgements for topic B1 reveals that there is almost no overlap among the elements that belong to any of the nine points of the relevance scale. More precisely, there is 14% set-based overlap among the 110 `E3S3` elements, 0% overlap among the 70 `E1S1` elements, and 0% overlap for the other seven points of the relevance scale. The above finding therefore confirms the hypothesis that users do not want to retrieve (and thus do not tolerate) redundant information.

Correlation between Relevance Grades

The lower half of Table 2 shows the correlation between the grades of the two relevance dimensions for topic B1, as judged by users. For both *Exhaustivity* and *Specificity*, two strong correlations are visible. First, in 66% of the cases a marginally exhaustive (`E1`) element is also judged as marginally specific (*S1*) (and vice versa). Second, in 70%

Assessor		User judgements											Agreement
Judgement	Total	E3S3	E3S2	E3S1	E2S3	E2S2	E2S1	E1S3	E1S2	E1S1	E0S0	Total	(%)
E3S3	2	25	10	0	5	4	1	0	2	1	0	48 (2)	52.08
E3S2	0	0	0	0	0	0	0	0	0	0	0	0 (0)	0.00
E3S1	1	1	0	0	0	0	2	0	0	0	0	3 (1)	0.00
E2S3	6	60	14	1	18	13	4	3	7	8	0	128 (6)	14.06
E2S2	2	14	4	1	2	1	0	1	0	0	1	24 (1)	4.17
E2S1	11	1	0	0	2	1	0	0	0	2	0	6 (3)	0.00
E1S3	0	0	0	0	0	0	0	0	0	0	0	0 (0)	0.00
E1S2	1	0	0	0	2	1	0	0	0	1	0	4 (1)	0.00
E1S1	9	3	2	1	1	7	3	0	2	11	17	47 (5)	23.40
E0S0	1158	1	6	2	2	7	9	7	6	36	99	175 (59)	56.57
Total	1190	105	36	5	32	34	19	11	17	59	117	435 (78)	**15.10**

Table 3: The level of agreement between the assessor and the users for topic B1. For each point of the relevance scale, the percentage of users that agree with the assessor's judgements of corresponding elements is shown. Numbers in brackets represent numbers of unique elements judged by users. The overall level of agreement for topic B1 is shown in bold.

of the cases a highly exhaustive (E3) element is also judged as highly specific (*S3*) (and vice versa). The number of E1 elements is around 30% of the total number of relevant elements, whereas 44% of the total number of relevant elements are E3 elements. The number of *S1* and *S3* elements is almost the same as the number of E1 and E3 elements, respectively. No strong correlations are, however, visible in the case of E2 and *S2* elements.

4.3 Analysis of the Level of Agreement

The analysis of the level of agreement concerns the amount of information identified as relevant by *both* the assessor and the users. The aim of this analysis is to identify the best units of retrieval for topic B1.

Table 3 shows the level of agreement between the assessor and the users for each point of the relevance scale. The two columns on the left refer to the assessor's judgements, where for each relevance point (the Judgement column), the total number of judged elements that belong to this point is shown (the Total column). The values in the User Judgements columns show how users actually judged any (or all) of the corresponding elements judged by the assessor. The Total column on the right shows the total number of user judgements for each point of the relevance scale. Numbers in brackets represent numbers of unique elements judged by users. The Agreement column shows the level of agreement between the assessor and the users, where the percentage is calculated for each relevance point.

For example, the first row in the table indicates that there are two elements judged as E3S3 by the assessor, and that of 48 total user judgements, there are 25 cases when users judged any (or both) of these two elements as E3S3, ten cases as E3S2, five cases as E2S3, and so on. The level of agreement between the assessor and the users for the E3S3 point of the relevance scale is 52.08% (since in 25 out of 48 cases users judged these elements as E3S3). Note that for this relevance point we only consider the user judgements made on two unique elements, which correspond to the same elements judged as E3S3 by the assessor. As shown in the table, the overall level of agreement between the assessor and the users for topic B1 is 15%.

Several observations can be made from the statistics shown in Table 3.

First, users judged 19 (unique) of the 32 *relevant* elements as identified by the assessor for topic B1. In 7% of the cases, however, users judged some of these elements to be *not relevant*. Conversely, 59 (unique) of the 1,158 *non-relevant* elements, as identified by the assessor, were also judged by users, and in 43% of the cases users judged some of those elements to be *relevant*.

Second, the highest level of agreement between the assessor and the users is on highly relevant (E3S3) and non-relevant (E0S0) elements, with agreement values of 52% and 57%, respectively. This shows that both the assessor and the users clearly perceive the end points of the relevance scale. However, the other points of the relevance scale are not perceived as well. For example, although the highest number of user judgements is on the E2S3 relevance point (around 50%), in only 14% of the cases users actually judged these elements as E2S3. In fact, in the majority of the cases (47%), the users judged these elements to be highly relevant (E3S3). Similar observations can be made for the E1S1 relevance point, where in 36% of the cases the users judged these elements to be non-relevant (E0S0). Note that, even though the number of judged E3S3 and E1S1 elements is roughly the same, the level of agreement for the E3S3 relevance point is more than two times greater than the level of agreement for the E1S1 relevance point.

Last, a more detailed analysis of the above statistics reveals that the agreement between the assessor and the users is almost the same for each separate relevance dimension. More precisely, the overall agreement for *Exhaustivity* is 45%, whereas the overall agreement for *Specificity* is 44%. The agreement for highly exhaustive (E3) elements is 71%, where 20% of the total number of confirmed *relevant* elements is on E3 elements. On the other hand, the agreement for highly specific (S3) elements is 63%, where 68% of the confirmed relevant elements are S3 elements. This shows that although the number of user judgements for the S3 grade is more than three times greater than the number of judgements for the E3 grade, highly exhaustive elements are perceived better than highly specific elements.

File: cg/1998/g1016

Assessor		User judgements										Total
Element	Judgement	E3S3	E3S2	E3S1	E2S3	E2S2	E2S1	E1S3	E1S2	E1S1	E0S0	(users)
/article[1]	E3S3	9	3	0	0	0	0	0	0	0	0	12
//bdy[1]/sec[2]	E2S3	9	5	1	7	6	2	1	2	2	0	35
//bdy[1]/sec[3]	E2S2	14	4	1	2	1	0	1	0	0	1	24
//bdy[1]/sec[4]	E2S3	19	0	0	4	1	1	0	0	2	0	27
//bdy[1]/sec[5]	E2S3	18	3	0	3	2	1	0	2	1	0	30
//bdy[1]/sec[6]	E2S3	8	2	0	2	1	0	1	0	1	0	15
//bdy[1]/sec[7]	E2S3	6	4	0	2	2	0	1	3	2	0	20

File: cg/1995/g5095

Assessor		User judgements										Total
Element	Judgement	E3S3	E3S2	E3S1	E2S3	E2S2	E2S1	E1S3	E1S2	E1S1	E0S0	(users)
/article[1]	E3S1	1	0	0	0	0	2	0	0	0	0	3
//bdy[1]/sec[1]	E3S3	16	7	0	5	4	1	0	2	1	0	36
//bdy[1]/sec[2]	E0S0	0	0	0	0	0	0	0	0	1	2	3
//bdy[1]/sec[3]	E0S0	0	0	0	0	0	0	0	0	0	2	2
//bdy[1]/sec[4]	E0S0	0	0	0	0	0	0	0	0	0	3	3

Table 4: Distribution of relevance judgements for the XML files *cg/1998/g1016* (top) and *cg/1995/g5095* (bottom) for topic B1. For each element, the assessor judgement and the distribution of users' judgements are shown. The total number of users who judged a particular element is listed in the last column.

Best Units of Retrieval

Previous analysis shows that of all the *relevant* elements as judged by users, the `E3S3` point of the relevance scale has the highest level of agreement. There are two elements judged as highly relevant by the assessor for topic B1 – one `article` and one `sec` – that belong to different XML files. The `article` element belongs to file `cg/1998/g1016`, while the `sec` element belongs to `cg/1995/g5095`. We are interested in finding in these files the *best units of retrieval* for topic B1. In the following analysis, we examine the retrieval behaviour of both the assessor and the users for each of these files.

Table 4 shows the distribution of relevance judgements for relevant elements in the two XML files, as done by both the assessor and the users. The two columns on the left refer to the assessor, where for each relevant element in the file (the `Element` column), the assessor's judgement is also shown (the `Judgement` column). The values in the `User Judgements` columns show the distribution of users' judgements for each particular element; that is, the number below each relevance point represents the number of users that judged that element. The total number of users who judged a particular element is shown in the `Total` column.

For the file `cg/1998/g1016`, the top half of the table shows that the highly relevant (`E3S3`) `article` element was judged by 12 (out of 50) users, and that 75% of them confirmed it to also be highly relevant. Interestingly, around 70% of the relevant elements in this file have been judged as `E2S3` by the assessor, and there were 25 users (on average) who have also judged these elements. However, there is only a 14% agreement (on average) between the assessor and the users for the `E2S3` relevance point. In fact, if we take a closer look at the user judgements, we see that most users judged the `E2S3` elements to be highly relevant (`E3S3`) elements. For example, there were 27 users in total who judged the `sec[4]` element (judged as `E2S3` by the assessor), and 70% of them judged this element to be highly relevant (`E3S3`).

The above analysis shows that the agreement between the users and the assessor on the *best units of retrieval* for the file `cg/1998/g1016` is not exact. Further analysis confirms that the level of agreement between the assessor and the users is greater for highly exhaustive elements than for highly specific ones. More precisely, although the number of user judgements for the `S3` grade is more than ten times greater than the number of judgements for the `E3` grade, there is a 65% agreement for highly specific elements, while there is a 100% agreement for highly exhaustive elements.

For the file `cg/1995/g5095`, the lower half of Table 4 shows that there are only two elements identified as *relevant* by the assessor, which makes it impossible to draw any sound conclusions. The highly relevant `sec` element was judged by 36 (out of 50) users, and around 45% of the users also confirmed it to be highly relevant. Interestingly, three `sec` elements were judged as not relevant by the assessor, and almost all of the users who judged these elements also confirm them to be non-relevant.

5. BEHAVIOUR ANALYSIS FOR COMPARISON TOPICS

In this section, we separately analyse the assessor's and users' behaviour when judging the relevance of returned elements for the *Comparison* topic C2. In order to identify the best retrieval elements for this topic, we also analyse and compare the level of agreement between the assessor and the users.

5.1 Analysis of Assessor's Behaviour

Figure 5 shows the relevance judgements for the INEX 2004 CO topic 198 (topic C2) that were obtained from one assessor. As shown in the figure, the total number of relevant elements for topic C2 is 153, of which the majority (81%)

Assessor:		Exhaustivity					
		E3		E2		E1	
	Specificity	Sp|Ex (%)	Ex|Sp (%)	Sp|Ex (%)	Ex|Sp (%)	Sp|Ex (%)	Ex|Sp (%)
	S3	**100.00**	33.33	22.22	33.33	1.41	33.33
	S2	0.00	0.00	**66.67**	27.27	11.27	*72.73*
	S1	0.00	0.00	11.11	0.80	**87.32**	*99.20*
Users:		**Exhaustivity**					
		E3		**E2**		**E1**	
	Specificity	Sp|Ex (%)	Ex|Sp (%)	Sp|Ex (%)	Ex|Sp (%)	Sp|Ex (%)	Ex|Sp (%)
	S3	**52.99**	*48.84*	30.13	32.56	13.77	18.60
	S2	35.04	27.33	**43.59**	*43.33*	27.54	29.33
	S1	11.97	8.50	26.28	27.45	**58.68**	*64.05*

Table 5: Correlation between the grades of the two relevance dimensions for topic C2, as judged by both the assessor and the users. Depending on the relevance dimension, the highest correlation of each grade is shown either in bold (for Exhaustivity) or italics (for Specificity).

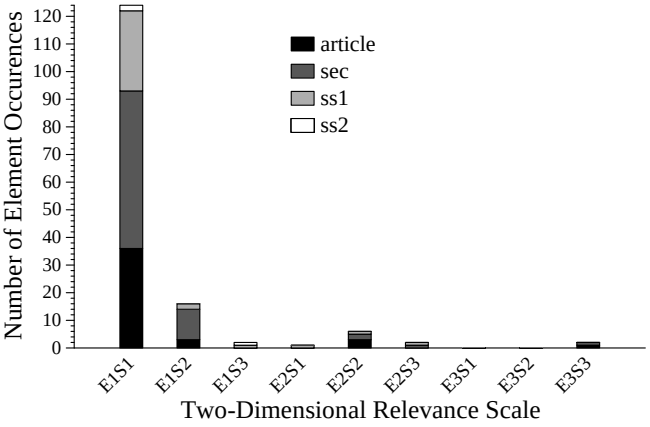


Figure 5: Analysis of assessor's behaviour for topic C2. For each point of the relevance scale, the figure shows the total number of relevant elements, and the number of relevant elements for each of the element names.

have been judged as `E1S1`. Interestingly, none of the relevant elements have been judged as either `E3S2` or `E3S1`. The distribution of the four element names is as follows. The `sec` elements occur most frequently with 72 occurrences, followed by `article` with 43, `ss1` with 35, and `ss2` with only three occurrences, respectively. The total number of elements that have been judged as non-relevant (`E0S0`) for topic C2 is 1094, of which 547 are `sec` elements, 304 are `ss1`, 191 are `article`, and 52 are `ss2` elements.

Level of Overlap

The above statistics show that the `E1S1` point of the relevance scale contains almost all of the relevant elements for topic C2. However, as for the topic B1, there is a substantial overlap among these elements. More precisely, there is a 63% set-based overlap among the 124 `E1S1` elements. On the other hand, the other points of the relevance scale – except the `E3S3` point – do not suffer from overlap. For the `E3S3` point, there is a 50% set-based overlap, where the two highly relevant elements (one `article` and one `sec`) belong to the same XML file.

Correlation between Relevance Grades

The top half of Table 5 shows the correlation between the grades of the two relevance dimensions for topic C2, as judged by the assessor. We observe that each of the three grades of the *Exhaustivity* dimension is strongly correlated with its corresponding grade of the *Specificity* dimension. This is most evident for the `E1` grade, where in 87% of the cases a marginally exhaustive (`E1`) element is also judged as marginally specific (*S1*). The number of `E1` elements is 93% of the total number of relevant elements. The same is not true for the grades of the Specificity dimension, however, where both the `S2` and `S1` grades are strongly correlated with the `E1` grade. Most notably, in 99% of the cases a marginally specific (`S1`) element is also judged as marginally exhaustive (*E1*), where the number of `S1` elements is 82% of the total number of relevant elements.

5.2 Analysis of Users' Behaviour

Figure 6 shows the relevance judgements for topic C2 that were obtained from 52 users. As shown in the figure, the total number of occurrences of relevant elements is 445, of which around half of that number are elements that belong to the following three points of the relevance scale: `E1S1` (101), `E2S2` (66), and `E3S3` (63). Interestingly, approximately the same number of users (34 out of 52) judged at least one element that belongs to each of these three points. In contrast, 22 users (on average) judged at least one element that belongs to the other six points of the relevance scale.

The distribution of the four element names is as follows. The `sec` and `article` elements occur most frequently with 159 and 153 occurrences, followed by `ss1` elements with 130, and `ss2` elements with only three occurrences, respectively. The total number of element occurrences judged as non-relevant (`E0S0`) for topic C2 is 170, of which 116 are `sec` elements, 27 are `ss1`, 26 are `article`, and only one element is an `ss2` element. Also, 38 out of 52 users have judged at least one element as `E0S0`.

Level of Overlap

Further analysis of the user judgements for topic C2 reveals that there is almost no overlap among the elements that belong to any of the nine points of the relevance scale. More specifically, there is 3% set-based overlap for the `E1S1` point,

Assessor		Users											Agreement
Relevance	Total	E3S3	E3S2	E3S1	E2S3	E2S2	E2S1	E1S3	E1S2	E1S1	E0S0	Total	(%)
E3S3	2	6	4	1	1	0	0	1	0	1	0	14 (2)	42.86
E3S2	0	0	0	0	0	0	0	0	0	0	0	0 (0)	0.00
E3S1	0	0	0	0	0	0	0	0	0	0	0	0 (0)	0.00
E2S3	2	16	4	1	6	7	1	0	0	2	1	38 (2)	15.79
E2S2	6	20	8	0	12	12	6	3	5	7	3	76 (5)	15.79
E2S1	1	2	0	1	1	4	0	0	1	1	0	10 (1)	0.00
E1S3	2	0	0	0	0	1	0	1	0	1	1	4 (2)	25.00
E1S2	16	1	1	1	2	1	0	1	2	1	2	12 (7)	16.67
E1S1	124	17	19	6	16	24	16	8	18	45	38	207 (34)	21.74
E0S0	1094	2	2	2	3	3	10	5	9	25	85	146 (52)	58.22
Total	1247	64	38	12	41	52	33	19	35	83	130	507 (105)	**19.61**

Table 6: The level of agreement between the assessor and the users for topic C2. For each point of the relevance scale, the percentage of users that agree with the assessor's judgements of corresponding elements is shown. Numbers in brackets represent numbers of unique elements judged by users. The overall level of agreement for topic C2 is shown in bold.

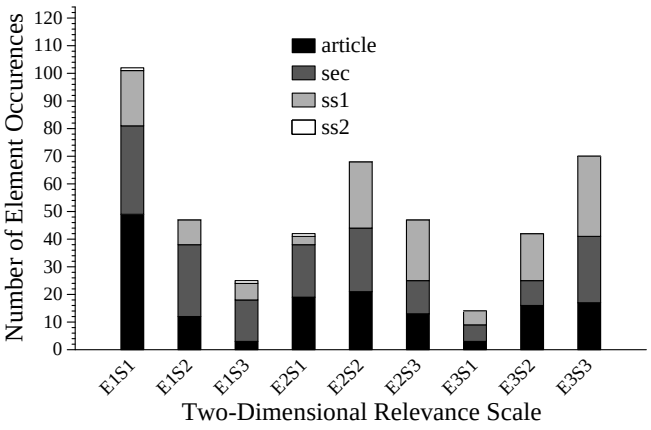


Figure 6: Analysis of users' behaviour for topic C2. For each point of the relevance scale, the figure shows the total number of relevant elements, and the number of relevant elements for each of the element names.

0% for the E2S2 point, 9% overlap for the E3S3 point, and 0% overlap for the other six points of the relevance scale.

Correlation between Relevance Grades

The lower half of Table 5 shows the correlation between the grades of the two relevance dimensions for topic C2, as judged by users. Although no strong correlations are visible, the values in the table show that, as in assessor's case, the highest correlations are between the same grades of each of the two relevance dimensions.

5.3 Analysis of the Level of Agreement

In this section we analyse the amount of information identified as relevant by *both* the assessor and the users. Table 6 shows the level of agreement between the assessor and the users for each point of the relevance scale. Three observations can be made from the statistics shown in the table.

First, users judged 53 (unique) of the 153 *relevant* elements as identified by the assessor for topic C2. In 12% of the cases, however, users judged these elements to be *not relevant*. Conversely, 52 (unique) of the 1094 *non-relevant* elements, as identified by the assessor, were also judged by users, and in 42% of the cases users judged these elements to be *relevant*.

Second, as for topic B1 the highest level of agreement between the assessor and the users is on the end points of the relevance scale: E3S3 (43%) and E0S0 (58%), although the number of user judgements for the E3S3 relevance point is much less than the number of judgements for the E0S0 point. The E1S1 relevance point has the highest number of user judgements (207 out of 507), and in 22% of the cases users also judged these elements to be E1S1. Also, there are 76 user judgements for the E2S2 relevance point, however in 26% of the cases users actually judged the E2S2 elements to be highly relevant (E3S3) elements.

Third, a more detailed analysis shows that the level of agreement between the assessor and the users differs for each separate relevance dimension. More precisely, the overall agreement for *Exhaustivity* is 53%, while the overall agreement for *Specificity* is 45%. The agreement for highly exhaustive (E3) elements is 79%, and 4% of the total number of confirmed *relevant* elements is on E3 elements. In contrast, the agreement for highly specific (S3) elements is 55%, where 18% of confirmed relevant elements are S3 elements. This shows that, as for topic B1, highly exhaustive elements are perceived better than highly specific elements.

Best Units of Retrieval

There are two elements judged as highly relevant by the assessor for topic C2, one article and one sec, which belong to the same XML file: co/2000/rx023. To identify the best units of retrieval, in the following we examine the behaviour of both the assessor and the users for this file.

Table 7 shows the distribution of relevance judgements for relevant elements in the XML file co/2000/rx023, as done by both the assessor and the users. As shown in the table, the two highly relevant (E3S3) elements were judged by the same number of users (seven out of 52). Of the users that judged each of these elements, 57% confirmed the article[1] to be highly relevant, while only 29% confirmed the sec[3] element to be highly relevant. Many users, however, found

File: co/2000/rx023

Assessor		User judgements										Total
Element	Judgement	E3S3	E3S2	E3S1	E2S3	E2S2	E2S1	E1S3	E1S2	E1S1	E0S0	(users)
/article[1]	E3S3	4	3	0	0	0	0	0	0	0	0	7
//bdy[1]/sec[1]	E2S2	5	0	0	1	2	2	1	3	1	2	17
//bdy[1]/sec[2]	E2S2	0	0	0	1	1	1	0	1	0	1	5
//bdy[1]/sec[3]	E3S3	2	1	1	1	0	0	1	0	1	0	7
//bdy[1]/sec[3]/ss1[1]	E2S2	10	3	0	8	6	2	2	1	1	0	33
//bdy[1]/sec[3]/ss1[2]	E2S1	2	0	1	1	4	0	0	1	1	0	10
//bdy[1]/sec[3]/ss1[4]	E1S1	3	0	1	1	1	0	0	0	1	0	7
//bdy[1]/sec[3]/ss1[5]	E2S3	7	3	1	4	2	0	0	0	0	0	17
//bdy[1]/sec[3]/ss1[6]	E1S3	0	0	0	0	1	0	0	0	0	1	2
//bdy[1]/sec[4]	E2S3	9	1	0	2	5	1	0	0	2	1	21
//bm[1]/app[1]/sec[1]	E1S1	0	0	0	1	0	0	0	1	0	3	5

Table 7: Distribution of relevance judgements for the XML file *co/2000/rx023* for topic C2. For each element, the assessor judgement and the distribution of users' judgements are shown. The total number of users who judged a particular element is listed in the last column.

the child elements of the `sec[3]` element (such as `ss1[1]`, `ss1[4]` and `ss1[5]`) to be highly relevant.

From the above distribution of relevance judgements it is hard to draw any sound conclusions as to which elements constitute *best units of retrieval* for this file. Further analysis of the two behaviours for this file again confirms that, for topic C2, the level of agreement between the assessor and the users is greater for highly exhaustive than for highly specific elements. Specifically, although the number of user judgements for the `S3` grade is four times greater than the number of judgements for the `E3` grade, the agreement for highly specific elements is 56%, while there is a 79% agreement for highly exhaustive elements.

6. DISCUSSION

In previous sections we separately studied the behaviour of the assessor and the users when judging the relevance of returned elements. We also analysed the level of agreement between the assessor and the users in order to identify the best units of retrieval for each of the two topics.

According to the assessor, most of the relevant elements for topic B1 reside in the `E2S1` and `E1S1` points of the relevance scale. The `E1S1` relevance point also contains most of the relevant elements for topic C2. In both topic cases, however, there is a substantial overlap among these relevant elements: 60% for topic B1, and 63% for topic C2. There are no visible correlations between the grades of each relevance dimension for the assessor of topic B1, whereas for for the assessor of topic C2 each of the three grades of the *Exhaustivity* dimension is strongly correlated with its corresponding grade of the *Specificity* dimension.

According to users, most of the relevant elements in both topic cases reside in the `E1S1`, `E2S2`, and `E3S3` relevance points. Moreover, there is almost no overlap among the relevant elements. Unlike in the assessor's case, the highest correlations between the grades of the relevance dimensions are between the same grades of each of the two dimensions, irrespective of the choice of the topic used. This shows that the two INEX relevance dimensions are not perceived as orthogonal dimensions; in fact, users behave as if each of the grades from either dimension belongs to only one relevance dimension.

The latter finding suggests that the *common aspect* influencing the choice of combining grades from the two INEX relevance dimensions is the fact that the users can not make a clear distinction between the two dimensions (since they are both based on topical relevance). However, it does not mean that the two INEX relevance dimensions are the same. On the contrary, from the *Exhaustivity* definition, higher aspect coverage does not imply that there is less non-relevant information in an element, which means there is no one-to-one correspondence between the two INEX dimensions. Rather, the users' perception – which was empirically identified in this study – suggests that the cognitive load of simultaneously choosing the grades for *Exhaustivity* and *Specificity* is too difficult a task. Part of the problem may be that the users (and the assessor) may not have understood an important property of the *Specificity* dimension: an element should be judged as *highly specific (S3)* if it *does not* contain *non-relevant* information.

The low level of overlap between the judged elements in the users' case shows that *retrieving overlapping units of information is not what users really want.* However, the higher level of overlap in the assessor's case does not necessarily mean that the assessor's behaviour is very different from that of users; indeed, there are *at least* two external factors that may have influenced the observed level of overlap for the assessor:

- The assessor was required to judge many more elements than the users, in order for the obtained relevance judgements to be as exhaustive (and as consistent) as possible; and
- The assessor and the users used different system interfaces, which may have introduced a bias in the way the elements were judged.

The highest level of agreement between the assessor and the users in both topic cases is respectively on highly relevant (`E3S3`) and non-relevant (`E0S0`) elements, which shows

that both the assessor and the users clearly perceive the end points of the relevance scale. However, *the other points of 10-point relevance scale were not perceived as well*. When the two relevance dimensions were analysed separately, we observed that – in both topic cases – *Exhaustivity* is perceived better than *Specificity*.

The above findings suggest that a much simpler relevance scale, and therefore, a much simpler relevance definition, would be a preferable choice for INEX. In the following we propose one such definition of relevance.

Aspects and Dimensions of Relevance

There are three aspects on which our new definition of relevance is based on:

- There should be only *one* dimension of relevance based on *topical relevance* (rather than two);
- The relevance dimension should use a *binary* relevance scale (rather than graded relevance scale), which determines whether a unit of information is *relevant* or *not* to an information need; and
- There should be second *orthogonal* dimension of relevance, based on the hierarchical relationships among the units of information in XML documents.

The first aspect makes the new relevance definition much simpler than the current one, and more importantly, enables a straightforward integration in the unified relevance framework [13]. The second aspect is directly inspired by the analysis of the level of agreement between the assessor and the users; indeed, the highest level of agreement was shown to be either on *highly relevant* or on *non-relevant* units of retrieval. This means that both the assessor and users clearly agree upon the *binary nature* of topical relevance of the retrieved units, indicating that a unit is either *relevant* or *not* to an information need. The second dimension of relevance, as introduced in the third aspect above, is completely orthogonal to the first dimension. It is defined as follows.

The extent to which a unit of information is relevant to an information need is measured by considering the *difference* between:

- The extent to which aspects of the information need are covered within the unit; and
- The extent to which these aspects are covered within the other *related* units (ancestors or descendants) in the document hierarchy.

For example, a relevant information unit is *just right* to an information need if it mainly just covers aspects of the information need. Alternatively, the information unit can be either *too broad* or *too narrow* to the information need. A relevant information unit is *too broad* if there is a *descendant* that mainly just covers aspects of the information need. Conversely, a relevant information unit is *too narrow* if there is an *ascendant* that is *just right*.

The second dimension of relevance, as defined above, is very similar to *document coverage* used in INEX 2002 [9]. 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. List and de Vries [11] describe a formal approach to modelling the document coverage. Similar to our second dimension, some aspects of document coverage depend on the context where the information unit resides, stating that "the component is too small to act as a meaningful unit of information when retrieved by itself" [9]. This, however, makes the document coverage to also be dependent on the size of the retrieved unit. The size of the unit of retrieval, on the other hand, is not explicitly considered in our relevance dimension.

New Relevance Definition for XML Retrieval

Considering the above observations, we propose the following definition of relevance:

- An information unit is *not relevant* to an information need if it does not cover any of the aspects of the information need;
- An information unit is *relevant* to an information need if it covers any of the aspects of the information need. The extent to which the unit is relevant to the information need can be one of the following:
  - *Broad*, if the unit is too broad and includes other, non-relevant information;
  - *Narrow*, if the unit is too narrow and is part of a larger unit that better covers aspects of the information need; and
  - *Just right*, if the unit mainly just covers aspects of the information need.

The above relevance definition has the following properties:

- In any one document path from the root element to a leaf, *at most* one element can be *Just right*. However, multiple *Just right* elements can exist in an XML document if they belong to different paths;
- Every element in a path that resides *above* the *Just right* element is too broad, and only such elements are considered to be too broad; and
- Every element considered to be too narrow is either a *child* of an element that is *Just right*, or a child of an element that is too narrow. Also, not every child of a relevant element has to be relevant.

There are two relevance dimensions described by the above definition: one based on the topical relevance, which uses a binary relevance scale (*relevant* or *non-relevant*); and another based on hierarchical relationships among the information units in XML documents, which uses a three-graded relevance scale (*Broad*, *Narrow*, or *Just right*).

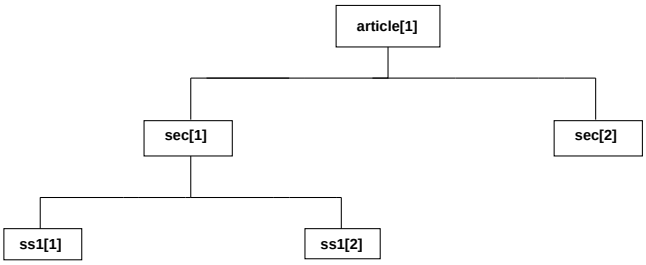


Figure 7: A representation of an XML document

Example Scenarios

We further explain the new relevance definition with several example scenarios, with reference to the XML document representation in Figure 7.

Scenario 1: Assume that only `ss1[1]` is relevant to an information need, and that it mainly just covers aspects of the information need. Because of the hierarchical relationships between the elements in the above document, both `sec[1]` and `article[1]` will also be relevant to the information need. However, since `ss1[2]` contains no relevant information, `sec[1]` becomes *too broad*. The same is also true for `article[1]`. The set of relevant elements (or the full recall base) in this scenario consists of three elements: one *Just right* and two *Broad*.

Scenario 2: Assume that both `ss1[1]` and `ss1[2]` are relevant to an information need, and they also mainly just cover its aspects. The `sec[1]` element in this case contains two *Just right* children, which also makes it *Just right*. Indeed, the two `ss1` elements may cover two different aspects of the information need, or they may cover a single aspect from two different perspectives. Since the additional context provided by `sec[1]` is (arguably) more desirable than each of the two separate contexts of its children, both the `ss1` elements become *too narrow*. Also, since `sec[2]` contains no relevant information, `article[1]` becomes *too broad*. The full recall base in this scenario consists of four relevant elements: one *Just right*, one *Broad*, and two *Narrow*.

Scenario 3: Assume that the three elements, `ss1[1]`, `ss1[2]`, and `sec[2]`, are relevant to an information need, and all of them mainly just cover its aspects. The full recall base in this scenario consists of five relevant elements: one *Just right* and four *Narrow*, where `article[1]` is the only element that is *Just right*.

Exploring Aspects of XML Retrieval

Different aspects of XML retrieval may be explored by using the new relevance definition.

One aspect would be to measure the XML retrieval effectiveness when only *Just right* elements are considered in the retrieval task. Note that in this case the full recall base consists of non-overlapping relevant elements, so there is no overlap problem during evaluation.

Another aspect would be to separately consider the *Broad* and the *Narrow* relevant elements in the recall base, and to measure the retrieval effectiveness against each of these elements. Indeed, different topics (or queries) require different granularity or relevant elements [15]. However, in both of these cases different techniques may be needed to deal with the overlap problem.

Previous work done by Voorhees in the field of Web retrieval confirms the hypothesis that different retrieval techniques need be used to retrieve highly relevant, rather than just any relevant, Web pages [21]. It may thus be worthwhile exploring whether, in the field of XML retrieval, different retrieval techniques would be needed to retrieve *Just right*, rather than any *Broad* or *Narrow*, relevant units of information.

Comparison with the INEX Relevance Definition

Compared to the current INEX relevance definition, the new definition of relevance is much simpler. Indeed, instead of having a 10-point relevance scale that uses various combination of grades of the two INEX dimensions as values, the new relevance definition uses a four-point relevance scale with the following values: *Non-relevant*, *Narrow*, *Just right*, and *Broad*.

Also, more than one mappings may be possible between the INEX relevance definition and the new one. For example, a partial mapping of the new four-point relevance scale to the INEX 10-point relevance scale is as follows.

1. *Non-relevant* `<=> E=0, S=0 (E0S0)`
2. *Just right* `<=> E=3, S=3 (E3S3)`
3. *Broad* `<=> E=3, S<3 (E3S2, E3S1)`
4. *Narrow* `<=> E<3, S=3 (E2S3, E1S3)`

The above mapping is partial as it does not include the following four INEX relevance points: `E2S2`, `E2S1`, `E1S2`, and `E1S1`. One reason for this is that we choose only a highly relevant (`E3S3`) element on a path to represent a *Just right* element. From the properties of the new relevance definition (as outlined above), it follows that a *Broad* or a *Narrow* element could then be either *above* or *bellow* the *Just right* element, which limits the mapping choices. Another reason, however, stems from the fact that these four points of the relevance scale were not well perceived by both the assessor and the users. The latter may be the most probable cause for the observed inconsistencies regarding the *Specificity* dimension. Nevertheless, for the purposes of the evaluation of XML retrieval there is almost no need to modify some of the current INEX metrics in order to use the new relevance definition.

The new relevance definition could also easily be applied to the recent proposal of performing the assessor's relevance judgements at INEX 2005. This proposal is as follows: first, for a returned article the assessor will be asked to highlight all of the relevant content. Second, after the assessment tool automatically identifies the elements that enclose the highlighted content, the assessor will need to judge the level of *Exhaustivity* of these elements and of all their ancestors. Last, based on the highlighted text, the level of *Specificity*

will be computed automatically as a ratio of relevant to non-relevant information, however a mapping may be needed to get the four relevance grades for the *Specificity* dimension.

Although we agree that the above approach is very promising, it is still unclear whether keeping the current INEX relevance dimensions, along with their corresponding grades, would help reducing the cognitive load of the assessor (or the users) while performing the relevance judgements. The new relevance definition, on the other hand, is much simpler, and it also fits very nicely with the above proposal.

7. CONCLUSIONS

In this work, we have undertaken a detailed analysis of assessor's and users' behaviour in the context of XML retrieval. We have shown that the two relevance dimensions used by INEX, *Exhaustivity* and *Specificity*, are not orthogonal and are perceived as one dimension by users. By analysing the level of agreement between the assessor and the users, we also wanted to identify how both of them perceive the points of the INEX 10-point relevance scale; the results of our analysis show that the highest level of agreement is on the end points of the relevance scale, which means that a much simpler relevance scale would be a preferable choice for the field of XML retrieval. We have proposed a new definition of relevance to be used by INEX, and argued that its corresponding relevance scale is simpler and more comprehensive than the one currently used.

Our analysis also shows that, although the assessor handles the overlap problem differently than users, in the users' case there is almost no overlap between the elements judged as relevant. The latter confirms the hypothesis that users do not want to retrieve, and thus do not tolerate, redundant information.

We have not discussed how the overlap problem may be modelled by the new relevance definition. As argued previously, it may be possible to model the overlap problem by using a separate relevance dimension based on novel relevance, which can be integrated into the *Context* component of the unified relevance framework [13]. However, in this paper we do not pursue this discussion any further.

The observed retrieval behaviour of the assessors and users was based on two topics, each from a different topic category. We did not observe any notable differences among the above behaviours for the two topics. However, analysis of a greater number of topics is needed to confirm the significance of our findings. This will enable a comparison between the observed and the overall behaviour of the assessors and users, which will certainly establish the XML retrieval environment in a more consistent manner. We leave the activities related to this analysis for future work.

It is our hope that, by analysing the different aspects of the observed retrieval behaviour, the work presented in this paper will aid better understanding of the important issues surrounding INEX and the field of XML retrieval.

Acknowledgements

We thank Saied Tahaghoghi and the anonymous reviewers for providing useful comments on earlier drafts of this paper.

8. REFERENCES

[1] J. Allan, R. Gupta, and V. Khandelwal. Temporal summaries of news topics. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10–18, 2001.

[2] P. Borlund. The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3):1–38, 2003. `http://informationr.net/ir/8-3/paper152.html`.

[3] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336, 1998.

[4] K. Finesilver and J. Reid. User behaviour in the context of structured documents. In *Proceedings of the 25th European Conference on IR Research (ECIR), Pisa, Italy, April 2003*, pages 104–119, 2003.

[5] 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. Springer-Verlag GmbH, May 2005.

[6] N. Gövert, N. Fuhr, M. Abolhassani, and K. Großjohann. Content-oriented XML retrieval with HyREX. In *Proceedings of the First International Workshop of the INitiative of the Evaluation of XML Retrieval, INEX 2002, Dagstuhl Castle, Germany, December 8–11, 2002*, pages 26–32, 2003.

[7] K. Hatano, H. Kinutan, M. Watanabe, Y. Mori, M. Yoshikawa, and S. Uemura. Keyword-based XML fragment retrieval: Experimental evaluation based on INEX 2003 relevance assessments. In *Proceedings of the Second International Workshop of the INitiative of the Evaluation of XML Retrieval, INEX 2003, Dagstuhl Castle, Germany, December 15–17, 2003*, pages 81–88, 2004.

[8] 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 on Research and Development in Information Retrieval*, pages 72–79, 2004.

[9] 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), Sunderland, UK, April 2004*, pages 296–310, 2004.

[10] V. Lavrenko. A generative theory of relevance. *PhD dissertation*, University of Massachusetts, Amherst, MA, 2004. `http://ciir.cs.umass.edu/~lavrenko/thesis.pdf`.

[11] J. A. List and A. P. de Vries. XML-IR: Coverage as a part of relevance. In *Proceedings of the Dutch-Belgian IR Workshop (DIR), Leuven, Belgium, December 2002*, pages 7–12, 2002.

[12] S. Malik, M. Lalmas, and N. Fuhr. Overview of INEX 2004. In Fuhr et al. [5], pages 1–15.

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

[14] J. Pehcevski, J. A. Thom, S. M. M. Tahaghoghi, and A.-M. Vercoustre. Hybrid XML retrieval revisited. In Fuhr et al. [5], pages 153–167.

[15] J. Pehcevski, J. A. Thom, and A.-M. Vercoustre. Hybrid XML retrieval: Combining information retrieval and a native XML database. *Information Retrieval*, 8(4):571–600, 2005.

[16] 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, 2004.

[17] T. Saracevic. Relevance reconsidered. In *Proceedings of the Second Conference on Conceptions of Library and Information Science (COLIS), Copenhagen, Denmark*, pages 201–218, 1996.

[18] I. Soboroff and D. Harman. Overview of the TREC 2003 Novelty track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC-12)*, pages 38–53, 2004.

[19] A. Tombros, B. Larsen, and S. Malik. The Interactive track at INEX 2004. In Fuhr et al. [5], pages 410–423.

[20] A. Tombros, I. Ruthven, and J. M. Jose. How users assess web pages for information seeking. *Journal of the American Society for Information Science and Technology*, 56(4):327–344, 2005.

[21] E. M. Voorhees. Evaluation by highly relevant documents. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–82, 2001.

[22] C. Zhai. Risk minimization and language modeling in text retrieval. *PhD dissertation*, Carnegie Mellon University, Pittsburgh, PA, 2002. `http://www.cs.cmu.edu/~czhai/thesis.pdf`.

Wanted: Element Retrieval Users

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

ABSTRACT

Document centric information retrieval is used every day by people all over the world. It is an application well studied, well understood, and of which there is a sound user model. Element retrieval, on the other hand, is a new field of research, with no identified applications, no users, and without a user model.

Some of the methodological issues in element retrieval are identified. The standard document collection (the INEX / IEEE collection) is shown to be unsuitable for element retrieval, and the question is raised – does such a suitable collection exist? Some characteristics of querying behavior are identified, and the question raised – will users ever use structural hints in their queries? Examining the judgments and metrics, it is shown that the judgments are inconsistent and the metrics do not measure the same things.

It is suggested that identifying an application of element retrieval could resolve some of these issues. Aspects of the application could (and should) be modeled, resulting is a more sound field of element retrieval. Alternatively, whatever it is, users don't want it, judges can't judge it, and the metrics can't measure it.

1. INTRODUCTION

INEX [3] was introduced in 2002 as a forum for the evaluation of element retrieval from XML documents. Since then there has been considerable discussion on relevance ranking algorithms, exactly as expected. Unexpectedly, there has also been considerable discussion on element retrieval methodology.

Element retrieval differs enormously from traditional document retrieval. The chunk of retrieval is a document element, not a document. Since elements might overlap within a document, this raises the issue of identifying the "best" element to return (from a given path). In effect the search engine can increase the exhaustivity (E) of a result by returning elements close to the root of a document tree, or can increase the specificity (S) of a result by returning elements close to the leaves. The search engine must balance these two to identify the most appropriate elements to return to the user. At INEX E and S scores are marked on a 4 point scale (0 = not, 1 = marginally, 2 = fairly, 3 = highly), and written EnSm.

As structure exists within the documents, users are able to use that structure in their queries. INEX identifies two types of queries, those that contain structure (Content and Structure (CAS) queries) and those that do not (Content Only (CO) queries).

Already, the difference between document retrieval and element retrieval is apparent. With element retrieval the user might use structural hints in their query, the search engine must interpret these, identify the most appropriate elements to return, and return a ranked list of elements (out of context) to the user.

As element retrieval is radically different from document retrieval, it has proven difficult to transfer prior experience to this new area. So it is time to start over – to address element retrieval as a new field, and to address the issues in the context in which they lie.

The first vital move is to identify an application of element retrieval.

Arguments are given here that show the current "model" is unsound: the document collection is inappropriate [18], the method of query is inappropriate, and the metrics are inappropriate.

Once users of an element retrieval system are identified, a sound model can be built and methodological issues can be resolved with reference to the model.

2. DOCUMENT COLLECTION

Intuitively element retrieval is important. Given a large collection of large documents marked up in XML [1] (or any other element based markup language such as SGML [7]) it's obvious that document components are a better result than whole documents. After all, the documents are large and the information relevant to the need is likely to be only part of the document.

In the case of a collection of books, each book might be a separate document. A document centric query to the collection would result in a ranked set of books. The user is then forced to wade through each book to find the relevant information (only a few pages might prove useful). Element retrieval might be used to identify only those relevant pages – surely presenting snippets of a book is more valuable to a user than presenting whole books.

Problematically, "a few pages" is not an element. An element retrieval system would return a book chapter, section or subsection, and not the said pages. Returning "a few pages" is passage retrieval [10]. Returning elements imposes a restriction on the passages: each passage must be a complete element (or at best a series of elements). Returning elements gives disregard to the most appropriate information to return and gives regard only to the most appropriate element to return.

Worse, the example of a collection of books assumes those books are divided into chapters, sections, and subsections. This is not the case for a many novels. Such work is a continuous flow of paragraphs from start to end. Markup would be used to identify the colophon, the title, and the author. The body might be a single element, or just a sequence of paragraphs. There are, after all, no other document components to mark up.

Surely element retrieval is useful for the IEEE document collection? This is a collection of 12,107 articles published by the IEEE computer society between 1995 and 2002. The documents were taken from 12 magazines and 6 transactions. They are primary and secondary academic scientific literature with some conference calls and news articles. This collection is used at INEX.

Science has organized itself to allow ideas, no matter of what size, to be published and cited. There are conference posters of only a couple of pages, conference articles of half a dozen pages, journal contributions (short and full), and books of varying size and structure.

Science has organized itself to allow these items to be cited. Articles are cited as a whole, books are either cited as a whole or with additional page references. Citers are expected to have read, in its entirety, the work being cited.

Page ranges in books has already been discussed – that isn't element retrieval!

A consequence of how articles are written and cited is that they are atomic. Each is the smallest indivisible unit of information that makes sense in its entirety. As such, element retrieval is a mismatch – there are no parts that make sense out of context. Even if there were, entire articles must be cited and as such read – reaffirming the atomicity of the articles.

There is but one part of an article that might make sense on its own – the front matter (author, title, and abstract). It is written to be read in isolation and to lie in isolation. There exist databases of millions of such "abstracts" (e.g. Medline). Using element retrieval to extract abstracts from documents is overkill.

For element retrieval to be useful it is necessary to identify the environment in which it might be used. Given there is a user who wants the said technology, it's possible to identify the characteristics of the document collection they are using.

The document collection must be in a markup language that contains elements. This might be XML, SGML or any other mark-up language.

The documents must contain several disparate parts (elements) that, while atomic in themselves, are also atomic in the context of the document.

O'Keefe [18] defines coupling as the association of an element to its context. He identifies elements that have low coupling as those suitable for element retrieval. Specifically, newspaper articles are given as an example of low coupling elements in a larger document (the newspaper). He also identifies extracting chapters from text books as a possible application of element retrieval.

If each newspaper story is held as a separate document then element retrieval is not necessary. Such is the case with the TREC [5] Wall Street Journal collection. Searching books is identified above as an inappropriate use of the technology.

The document elements must be large enough to contain relevant information, while at the same time small enough to be sub-documents. Kamps *et al*. [9] analyses the probability of an XML element being relevant given its length (in the IEEE collection) and identifies that although most elements are small (mean 29 terms), most relevant elements are not (mean over 1000 terms). In short, there is a mismatch between how elements are used and what information is relevant.

Element retrieval falls in the middle ground between question-answering and document retrieval. The units of retrieval are too large to be question answers, and too small to be documents. But the technology might be used in either.

Identifying elements that contain question answers might result in a reduced amount of natural language processing to obtain answers.

For document retrieval it might be used to identify where (within a document) the most relevant information lies for highlighting [22].

Perhaps element retrieval is a technology not at all appropriate for text. To retrieve elements is to pluck information from its context and present it as atomic. Such a technology should be applied where atomic information is strung together in an essentially random mix. Identifying such a place has proven hard.

Popular radio broadcast consists of segments of news, idle chatter and music interspersed with advertising. Stations give different news stories on the hour and half past the hour (to avoid repeating the stories every half hour). Perhaps element retrieval could be used to extract news stories on a given topic from the idle chatter and advertising. The same principle applies to "magazine" television broadcast such as MTV and E!.

Even within this context, it is not clear why element retrieval is necessary. Surely the atomic chunks can be separated from each other and stored as separate documents? If so, then element retrieval is not necessary. If not then the chunks are not atomic.

2.1 Discussion Point

A document collection for element retrieval must consist of documents that have a low coupling to their elements, while at the same time the coupling must be strong enough to bind the elements to the documents (or else why should the documents be maintained as such). The elements must be large enough to contain information that satisfies an information need yet elements tend to be very small chunks of text.

It is not obvious whether such a text collection exists. A suitable document collection might be found by changing the focus from text documents to audio or video.

3. QUERYING

Several pages from a book is identified above as "not an element". The IEEE collection is a collection of atomic elements that should not be broken into pieces. Element retrieval is a technology that is waiting for an application.

In the IEEE collection tags are used for two purposes: presentation and structure.

Presentation tags include those for italics, bold, and bold-italics (<it>, <b> and <bi>). Such tags are used, for example, to identify Latin words that are traditionally written in italics in English text. These tags are not the target of element retrieval.

Structure tags are used to mark paragraphs, subsections, sections and to separate the body from the front matter and back matter. It is these tags that are the target of element retrieval.

O'Keefe [18] identifies the majority of INEX topics in 2003 and 2004 targeting <article> or <sec> elements

Table 1: Target elements and number of times each is requested as a target element in an INEX 2003 / 2004 topic

Element	Occurrences	Proportion
sec	26	0.406
article	17	0.266
* & (p|fgc)	5	0.078
abs	4	0.063
p	4	0.063
bb	2	0.031
vt	2	0.031
bdy	1	0.016
bib	1	0.016
fig	1	0.016
fm	1	0.016
Total	64	1.000

Table 1 presents the list of target elements from 2003 and 2004. It can be seen that the majority of topics (67%) target <article> and <sec>. Only 11 unique tags were chosen from a possible 192 in the DTD. None of the tags were formatting tags.

Only 6% of the tags were identified as useful by topic authors. All those tags were structural. In these topics there is a 67% likelihood of the target being either <article> or <sec>.

One interpretation of this is that only <article> and <sec> tags are useful in this document collection. Searching for <article> elements is synonymous with whole document retrieval (a wanted but absent track at INEX). Searching for <sec> elements might be occurring because topic authors are required to submit structured topics, and there is no other useful element in the collection. The document collection only has two tags that might be used for retrieval, <article> and <sec>.

An alternative interpretation is that topic authors don't know (or don't want to know) the intricacies of the DTD. This is an argument often tabled for not using structural queries (the so-called CAS topics in INEX).

O'Keefe and Trotman [19] identify that not only use of structure is problematic; but also the syntax and semantics of structured queries. Of the 30 CAS topics at INEX 2003, topics written by IR researchers, 19 (63%) contained errors. This error rate appears to have dropped as a consequence of the introduction of the query language NEXI [26], but in practice it has not. Trotman and Sigurbjörnsson [27] identify that the online NEXI syntax parser was used 635 times for 84 CAS topics, or 7.5 times per query!

Experiments into the nature of interactive searching of elements were conducted at INEX 2004 [24]. On the comparative task the average number of search terms was 3.4, and on information foraging tasks (the background task), the average length of a query was 3.0 terms (including stop-words). In some participating groups the overall average was lower [15].

Tombros *et al*. [24] report that none of the queries contained use of the "+" or "-" operators. Quoted phrases appeared in less than 10% of queries. Only 50% of the log was analyzed due to software issues – so their result is partial.

Accepting that these queries could not contain structural hints; there still remains no evidence to suggest that structural hints would be used if they could be used. If mastery of "emphasis" and "negative emphasis" operators is beyond the scope of an interactive user, then the complexities of NEXI [26] certainly are.

Although there is no definitive evidence yet, it appears as though including structural hints in a query increases precision [20]. Experiments comparing structural with non-structural queries (of the same information need) are being conducted as part of INEX 2005 [23].

This leads to a conundrum. If structural hints increase precision, and users won't use them, then what to do? Use natural language queries [30]? Of 3.4 words on average? Of which one term is structural?

The enormous gulf between the searching behavior of database users and search engine users may account for the lack of use of the sophisticated search techniques. Again; identifying the target audience of element retrieval is essential. This will shed light on the patterns of query use by the audience. This, in turn, will help identify how queries should be formulated.

3.1 Discussion Point

The evidence predicts that users won't use structural hints in a query. It is the view of the author that early interactive experiments at INEX will show a slight contradiction to this. This contradiction is expected because structural hinting will appear novel in the mind of the user, who will experiment with it. Use during subsequent experiments is expected to match that of the "emphasis" operator.

If users won't use structural hints, then the INEX CAS task is futile. Concentration on identifying the target element (the CO task) is far more valuable than the CAS task.

If experiments conducted at INEX 2005 show no significant difference in the performance of CO and CAS runs then CAS should be dropped along with syntax for writing queries containing structural hints.

4. MEASUREMENT

With the introduction of the Cranfield methodology a line was drawn between the user and the search engine. On one side of the line lies the human / computer interaction (HCI) issues of social computer science, while on the other lies an experimental science of search engine design. In the early TREC experiments the line was mean average precision.

The documents, topics, and judgments are a model of interaction. The documents are a model of the type of information a user might find. A topic is a model of the type of query a user might have. The judgments are a model of how useful each document is to that user for that topic.

Mean average precision is a way of measuring the performance of a search engine with respect to the model. For as long as the performance metric is rigid, it is possible to change either the user model or the search engine. With mean average precision it is possible to determine quantitatively if one search engine is better than another given the user model. Search engine experiments are quantitatively reproducible and ranking functions are comparable.

The user model for element retrieval is not fixed – and the existing metrics do not measure the same thing.

Figure 1 shows the performance of each of the submitted runs at INEX 2004 (as well as those submitted to the LIP6 interface).

Shown is the performance of generalized inex_2002 (RP_g) [11] against generalized NG (RP_ng_o_g) [4]. From visual inspection the best runs scored using RP_g are not a good runs when scored using RP_ng_o_g. At INEX 2003, the University of Otago CO run ranked 1st using generalized ng-o while ranking 34th using generalized inex_2002. Whatever Otago did that year, it was good at NG but bad at inex_2002.

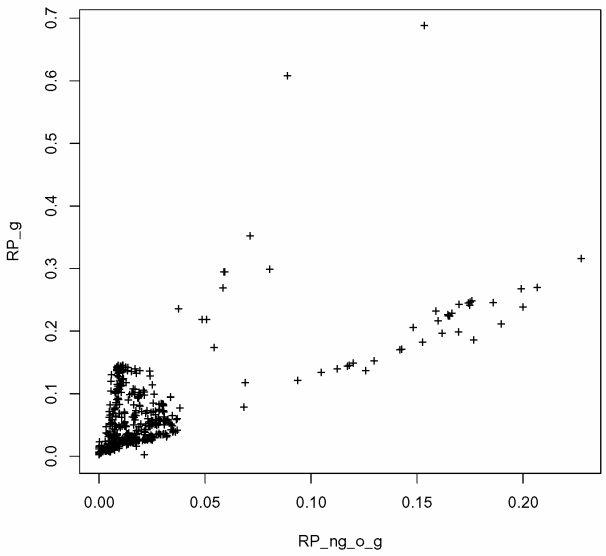


Figure 1: Performance of submitted runs at INEX 2004 (and the LIP6 web site) showing some systems perform well on NG (RP_ng_g) but badly on inex_2004 (RP_g). Image courtesy of Benjamin Piwowarski

Kazai *et al*.[13] recently introduced the metric XCG, an XML extension of the cumulative gain metric [8]. The top ten INEX 2004 runs scored with XCG share only 1 run with the top ten ranked using inex_2002 (and *vice versa*). These metrics measure different things.

Not only do the current metrics measure different things, but some are "hackable". For example, if a given element is known to be relevant then its parent must also be relevant, and its parent's parent, and so on up to the root of the document tree. Each of these elements might not be returning any additional relevant content, but returning such an ancestral list of elements is known to increase the inex_2002 score [13].

This process of "milking" the results to boost the score is commonplace. The two highest scoring CO runs at INEX 2004 show overlap of over 80%. Nine of the top ten show overlap of over 70%. That is, of the elements in the result list, over 70% of the elements have already been seen when scored.

Milking has been defended on two grounds.

If the search engine is trying to identify the most exhaustive and most specific elements (E3S3), then how can milking boost the performance?

Surely there can be no instance of a parent and child in the document tree both being E3S3. If the parent is the "ideal result" then how can a child also be the "ideal result"? In other words, it is simply not possible for both the root of the document tree and any descendant to be E3S3.

The "it can't happen" defense is unfounded. The description of topic 139 states "We wish to identify papers that cite work by authors Bertino or Jajodia that deal with "security models"". A whole document could be classified as E3S3 as could a single citation included in the same document. Of the 361 E3S3 judgments for this topic: 31 are whole documents, 304 are at or below a bibl (a reference).

Examining the judgments for 2004 (version 3[1]) 171 whole documents were judged E3S3, of which 163 have an element beneath the root that is also E3S3. A similar pattern can be seen in the 2003 judgments [20]. In other words, milking will also identify additional E3S3 elements as they, too, exist in paths through the document tree.

The second ground on which milking has been defended is simply that "everyone does it". This defense is untrue – 18 of the CO submissions at INEX 2004 contained no overlapping elements [12].

Milking violates the principle of user modeling. If there is no identifiable end user application of retrieval including milking, it is being done simply to score well on the metrics. In support, the interactive experiments have identified a user disapproval of the practice [15; 24].

The metrics are vitally important and must be treated with respect. Ranking function design is an optimization problem – the object is to optimize the metric score by changing the function. The recent use of Genetic Programming [25] as a method of finding the optimal function demonstrates this. The adoption of milking to boost the score is another. Whatever is necessary, however user-grounded or not, score boosting is rife.

Kekäläinen *et al*. [14] identify the influence of milking on runs at INEX 2004. By removing overlapping elements from their CO run the relative rank of their system dropped from 10th to 45th! It makes the difference from being in the top 10 to being in the top 50 (of 70).

A metric that is user grounded can't be hacked – if it could, it would no longer be modeling the behavior of a user. The current lack of user-grounded metrics at INEX makes the current evaluation round of questionable value. Participants are writing programs to score high on a metric for which there is no evidence of intrinsic value. In other words, they are writing programs that are getting very good at something that is known inherently to be very bad.

Identifying a user application of element retrieval will identify what users want – it is this that should be rewarded.

4.1 Discussion Point

The current collection of metrics is obscuring the problem of increasing performance. With high levels of overlap the current highest scoring systems are effective optimizations of the metric, however not likely to be viable information retrieval systems.

Before an effective metric can be devised it is necessary to stop looking at what might happen, what might be measured, and to

[1] Assessments for topic 127 are excluded as they were corrupt.

take a look at what a user might want. A metric should reflect user behavior and not *vice versa*.

5. JUDGMENTS

Before any metric can be used to compare the performance of two systems, it is necessary to have a stable set of judgments. During INEX 2004 twelve topics were chosen for judging by two judges each [17]. Lalmas *et al*. [16] report that the average level of exact agreement between judges was as low as 3.42% while the level of non-zero agreement was only 12.19%. Two judges non-zero agree if both consider the element to be other than E0S0.

Comparing these results to those of prior multiple judge experiments suggests INEX judge agreement is unacceptably low. Wilbur [29] reports that Saracevic reports that judges are known to agree somewhere between 40% and 70% of the time. At INEX the agreement level is (at best) 12% (according to Lalmas *et al*.).

Table 2: Document level relevance non-zero agreement between judges in the INEX 2004 judgments. P_j is the precision of judge, J_j, against the alternate judge

Topic	J_A	J_B	$\cup$	$\cap$	$\cap/\cup$	P_A	P_B
130	92	18	95	15	0.16	0.16	0.83
133	8	39	42	5	0.12	0.63	0.13
139	43	23	43	23	0.53	0.53	1.00
140	29	216	217	28	0.13	0.97	0.13
143	10	8	11	7	0.64	0.70	0.88
144	5	36	41	0	0.00	0.00	0.00
155	40	30	46	24	0.52	0.60	0.80
165	10	51	51	10	0.20	1.00	0.20
169	26	35	44	17	0.39	0.65	0.49
173	5	23	25	3	0.12	0.60	0.13
175	38	65	79	24	0.30	0.63	0.37
201	27	54	71	10	0.14	0.37	0.19
Total	333	598	765	166			
Mean					0.27	0.57	0.43

The Lalmas *et al*. comparison does not compare like with like.

Examining topic 130 (see Table 2) judge J_A identified 92 relevant elements whereas judge J_B identified 18 relevant elements. Of those 95 were unique, and of those only 15 were considered relevant by both judges.

Table 2 also presents the document level agreement (intersection divided by union) between the two judges. The level of agreement varies from no-agreement on topic 144, to agreement of 0.64 on topic 143. The mean document level agreement between the two judges is 0.27.

Table 3: Agreement levels at TREC and INEX

Evaluation	Agreement ($\cap/\cup$)
TREC-4 P/B	0.49
TREC-4 A/B	0.43
TREC-4 P/A	0.42
TREC-6	0.33
INEX-2004	0.27

Voorhees [28] examines the agreement levels in TREC-4 topics using three judges. Agreement levels of 0.42, 0.43, and 0.49 are seen between two judges and for all three 0.30 is seen. Voorhees reports these levels as high. In the INEX collection the overlap agreement for the 12 topics is 0.27. This is low by comparison to TREC.

Cormack *et al*. [2] report an experiment in which TREC-6 judgments from NIST were compared to those from judges at the University of Waterloo. In this experiment a mean overlap score of 0.33 is seen and reported (by Voorhees [28]). Again, huge variation is seen in the topics with agreement levels varying from none to total.

In Table 3 a comparison of the TREC-4, TREC-6 and INEX document based agreement levels is presented. From this it is clear the document-centered agreement levels at INEX are comparable to those at TREC. For details of agreement levels in pre-TREC collections see Harter [6].

The performance of a judge can be computed by taking the judgments for that judge and computing precision against the alternate set of judgments (presented in Table 2). Considering whole documents, the mean of precisions for the two judges is 0.57 and 0.43, comparable to that reported by Wilbur for Medline documents [29].

From this comparison it is reasonable to conclude that the performance of non-zero document centric judgments at INEX is consistent with those of TREC-4 and TREC-6. A larger study involving more than 12 topics is needed to confirm this observation. Experiments like those of Voorhees [28] are also needed to determine whether, or not, the disagreement between judgments affects the relative order of different systems.

Table 4: Element level non-zero agreement between judges for the 12 topics double judged at INEX 2004

Topic	J_A	J_B	$\cup$	$\cap$	$\cap/\cup$	P_A	P_B
130	1233	259	1328	164	0.12	0.13	0.63
133	37	451	474	14	0.03	0.38	0.03
139	562	889	1213	238	0.20	0.42	0.27
140	257	2418	2464	211	0.09	0.82	0.09
143	61	48	68	41	0.60	0.67	0.85
144	21	319	340	0	0.00	0.00	0.00
155	496	292	608	180	0.30	0.36	0.62
165	55	697	699	53	0.08	0.96	0.08
169	247	490	586	151	0.26	0.61	0.31
173	60	228	260	28	0.11	0.47	0.12
175	214	1468	1578	104	0.07	0.49	0.07
201	354	618	887	85	0.10	0.24	0.14
Total	3597	8177	10505	1269			
Mean					0.16	0.46	0.27

The INEX topic submission process demands that topic authors submit, along with the topic, a (small) list of elements considered relevant. This list can be compared to the topic assessments for consistency. Some of the with-topic judgments may not be in the judgment pool and *vice versa* so it is possible only to measure the extent to which a judge "changed their mind". That is, of the elements in both the with-topic list and the judgment list, how

many were judged non-relevant. This experiment was not conducted due to time constraints.

Examining the level of non-zero element agreement between judges, (in Table 4) judges do not agree on which elements are relevant. Comparing with the result in Table 2, it is reasonable to conclude that the judges do agree on which documents are relevant, but not on why!

The picture turns sour when E3S3 elements are examined (strict quantization). Table 5 lists the number of documents that are identified as containing E3S3 elements (that is, even if the document is not E3S3, there is an E3S3 element in the document). Judges do not agree on which documents contain the most specific and most exhaustive elements.

The level of E3S3 element agreement is shown in Table 6 where the agreement level is 0.05. There is almost total disagreement on which elements are most specific and most exhaustive.

Table 5: Documents judged to contain E3S3 elements by each judge of the multiple judged topics from the INEX 2004

Topic	J_A	J_B	$\cup$	$\cap$	$\cap/\cup$
130	1	10	10	1	0.10
133	0	0	0	0	0.00
139	38	20	39	19	0.49
140	0	9	9	0	0.00
143	0	0	0	0	0.00
144	0	7	7	0	0.00
155	4	7	9	2	0.22
165	10	3	11	2	0.18
169	3	7	8	2	0.25
173	0	4	4	0	0.00
175	2	3	4	1	0.25
201	0	4	4	0	0.00
Total	58	74	105	27	0.00
Mean					0.12

Table 6: E3S3 element level agreement for the 12 topics double judged at INEX 2004

Topic	J_A	J_B	$\cup$	$\cap$	$\cap/\cup$
130	2	42	42	2	0.05
133	0	0	0	0	0.00
139	361	169	451	79	0.18
140	0	32	32	0	0.00
143	0	0	0	0	0.00
144	0	10	10	0	0.00
155	5	22	24	3	0.13
165	29	15	38	6	0.16
169	10	21	30	1	0.03
173	0	26	26	0	0.00
175	18	5	22	1	0.05
201	0	4	4	0	0.00
Total	425	346	679	92	
Mean					0.05

This result suggests that although judges agree on which documents are relevant, they don't agree on why they are relevant or how relevant those documents are.

Experiments to determine if this disagreement affects the relative ranking of search engines is yet to be performed. At INEX 2004 two judgment sets were made available. Each contained judgments for 60 topics, however they only differed in the 12 topics discussed herein. That is, they were 75% identical because only 25% of topics were judged by more than one judge. Not surprisingly the relative performance of systems was relatively stable – no doubt because the judgment sets were essentially the same.

There could be many reasons why judges disagree on what constitutes a relevant element. Studies on whole document retrieval have identified a plethora of such factors [21]. With element retrieval there is at least one additional contributing factor – there is no agreed user model as there is no example application.

Identifying an application of element retrieval will help reduce disagreement levels. Judges will be aware of a common model and consequently will be able to refer to the model in case of uncertainty.

Of course, with an appropriate document collection and suitable queries such levels of disagreement may simply vanish. Disagreement levels may be a reflection of the collection and topics, not inherent in element retrieval.

5.1 Discussion Point

Judges agree on relevant documents at levels comparable to TREC. They agree less so on relevant elements, and less so again on relevance levels. This disagreement in relevance levels suggests quantization functions based on relevance levels will prove unsound.

The quantization functions rely on a judge's fine grained ability to identify the relevance level of a given element. It appears as though judges do not agree on this – if this is the case then developing ranking functions that utilize this is futile.

6. CONCLUSIONS

That element retrieval has methodological issues is evidenced by the INEX Element Retrieval Methodology Workshop. It is argued here that these problems stem from one cause, the lack of user grounding.

The identification of an (existing) application of element retrieval may resolve many of the issues.

A document collection containing elements that make sense as atomic retrieval results is needed. Should an application be identified then a document collection mirroring the collection in use could be built – the very collection in use might be used. With no application it is proving hard to identify even the distinguishing characteristics of a suitable collection. It is, however, proving possible to demonstrate that the characteristics of the existing collections make them unsuitable.

Given an application, the queries entered by users can be studied and suitable languages and querying methods can be identified. At present it appears as though even the simplest query operators are beyond the use of typical users. Given this, research into how to improve such searching strategies will have little or no measurable effect on performance.

The metrics used for measuring the performance of element ranking strategies have proven to be open to practices identified by users as of negative value. Again, if an application of element retrieval can be identified then the nature of a good result set can be identified. The metrics should reflect good user practice.

At present there is no identified application of element retrieval. There is no practical model and consequently no theoretical model. This has lead to multiple interpretations of the task and continued debate on what the search engine is trying to identify. In essence, each INEX participant has their own retrieval model.

Identifying an application of element retrieval is a vital first step. If it isn't possible to identify such an application, such an application may not exit. Unless the community can collectively identify such an application methodological issues will continue plague the research.

In summary, element retrieval methodological issues arise from one problem – the lack of a user model. To move beyond this, a real-world application must be identified and a model derived that is based on this use. In this way the identified element retrieval issues would be resolved against a user model.

7. REFERENCES.

[1] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., Yergeau, F., & Cowan, J. (2003). Extensible markup language (XML) 1.1 W3C proposed recommendation. The World Wide Web Consortium. Available: http://www.w3.org/TR/2003/PR-xml11-20031105/.

[2] Cormack, G. V., Palmer, C. R., To, S. S. L., & Clarke, C. L. A. (1997). Passage-based refinement (multitext experiements for TREC-6). In *Proceedings of the 6th Text REtrieval Conference (TREC-6)*, (pp. 171-186).

[3] Fuhr, N., Gövert, N., Kazai, G., & Lalmas, M. (2002). INEX: Initiative for the evaluation of XML retrieval. In *Proceedings of the ACM SIGIR 2000 Workshop on XML and Information Retrieval*.

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

[5] Harman, D. (1993). Overview of the first TREC conference. In *Proceedings of the 16th ACM SIGIR Conference on Information Retrieval*, (pp. 36-47).

[6] Harter, S. P. (1996). Variations in relevance assessments and the measurement of retrieval effectiveness. *Journal of the American Society for Information Science,* 47(1), 37-49.

[7] ISO8879:1986. (1986). *Information processing - text and office systems - standard generalised markup language (SGML)*.

[8] Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *Transactions on Information Systems,* 20(4), 422-446.

[9] Kamps, J., Rijke, M. d., & Sigurbjörnsson, B. (2004). Length normalization in XML retrieval. In *Proceedings of the 27th ACM SIGIR Conference on Information Retrieval*, (pp. 80-87).

[10] Kaszkiel, M., & Zobel, J. (1997). Passage retrieval revisited. In *Proceedings of the 20th ACM SIGIR Conference on Information Retrieval*, (pp. 178-185).

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

[12] Kazai, G., Lalmas, M., & Vries, A. d. (2004). Reliability tests for the xcg and inex-2002 metrics. In *Proceedings of the INEX 2004 Workshop*, (pp. 60-72).

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

[14] Kekäläinen, J., Junkkari, M., Arvola, P., & Aalto, T. (2004). Trix 2004 - struggling with the overlap. In *Proceedings of the INEX 2004 Workshop*, (pp. 127-139).

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

[16] Lalmas, M., Fuhr, N., Malik, S., Szlavik, Z., & Trang, V. H. (2004). *Some statistics for INEX 2004* (PDF of Presentation Slides). London: Queen Mary University of London.

[17] Malik, S., Lalmas, M., & Fuhr, N. (2004). Overview of INEX 2004. In *Proceedings of the INEX 2004 Workshop*, (pp. 1-15).

[18] O'Keefe, R. A. (2004). If INEX is the answer, what is the question? In *Proceedings of the INEX 2004 Workshop*, (pp. 54-59).

[19] O'Keefe, R. A., & Trotman, A. (2003). The simplest query language that could possibly work. In *Proceedings of the 2nd workshop of the initiative for the evaluation of XML retrieval (INEX)*.

[20] Pehcevski, J., Thom, J. A., Tahaghoghi, S. M. M., & Vercoustre, A.-M. (2004). Hybrid XML retrieval revisited. In *Proceedings of the INEX 2004 Workshop*, (pp. 153-167).

[21] Schamber, L. (1994). Relevance and information behavior. *Annual Review of Information Science and Technology,* 29, 3-48.

[22] Sigurbjörnsson, B. (2005 - to appear). Focused information retrieval from semi-structured documents (abstract). In *Proceedings of the 28th ACM SIGIR Conference on Information Retrieval*.

[23] Sigurbjörnsson, B., Trotman, A., Geva, S., Lalmas, M., Larsen, B., & Malik, S. (2005 - to appear). INEX 2005 guidelines for topic development. In *Proceedings of the INEX 2005 Workshop*.

[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] Trotman, A. (2005). Learning to rank. *Information Retrieval,* 8(3), 359-381.

[26] Trotman, A., & Sigurbjörnsson, B. (2004). Narrowed Extended XPath I (NEXI). In *Proceedings of the INEX 2004 Workshop*, (pp. 16-40).

[27] Trotman, A., & Sigurbjörnsson, B. (2004). NEXI, now and next. In *Proceedings of the INEX 2004 Workshop*, (pp. 41-53).

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

[29] Wilbur, W. J. (1998). A comparison of group and individual performance among subject experts and untrained workers at the document retrieval task. *Journal of the American Society for Information Science,* 49(6), 517-529.

[30] Woodley, A., & Geva, S. (2004). Nlpx at INEX 2004. In *Proceedings of the INEX 2004 Workshop*, (pp. 382-394).

Fine Tuning INEX

Alan Woodley
School of Software Engineering and Data Communications
Faculty of Information Technology
Queensland University of Technology
GPO Box 2434 Brisbane Q 4001 Australia
ap.woodley@student.qut.edu.au

Dr. Shlomo Geva
School of Software Engineering and Data Communications
Faculty of Information Technology
Queensland University of Technology
GPO Box 2434 Brisbane Q 4001 Australia
s.geva@qut.edu

ABSTRACT
Since 2002, INEX has been the benchmark for evaluating XML information retrieval (XML-IR) systems. INEX has based much of its evaluation methodology on that of existing workshops, albeit modified for the specific requirements of XML-IR. Due to some of the modifications, the time spent during evaluation phase of INEX takes a lot longer than comparable workshops. Here, we investigate ways to speed up the INEX evaluation process. We also investigate some structural changes and additional tasks that could be preformed at future INEX workshops.

Categories and Subject Descriptors
H.3.4 [**Information Storage and Retrieval**]: Systems and Software --- *performance evaluation*.

General Terms
Experimentation, Measurement

Keywords
Evaluation Methodology, System Pooling.

1. INTRODUCTION
In this position paper we propose and discuss several ideas that have been thrown around for quite some time at INEX workshops, on the mailing lists, and in private communications. In what follows we address some of the problems and proposed solutions and ideas in greater detail and some in less detail. Perhaps some of these can be discussed at the Glasgow workshop and adopted in future INEX collaborations.

Here we provided summaries of our proposals.

- Pooling submissions – Currently INEX uses a version of system pooling originally devised by Sparck Jones and Van Rijsbergen [3]. While system pooling has proven adequate, we propose a different pooling technique that may be superior. Our technique uses metasearch system to produce an assessment pool. Preliminary results indicate that metasearch pooling may be superior to the system pooling. We present a proposal to execute metasearch pooling at INEX 2005.
- Who contributes rare results? – Carrying on from the previous proposal, we know that under system pool results are taken from every submission, even those that will ultimately prove to be poor performing. The conjecture for including results from poor systems is that they may find rare (or even unique) relevant results. Here, we examine if this conjecture is true, and if INEX would be better served by not including results from poor performing systems in the pool (determined through committee ranking).
- Are inferred and additional results useful? - In INEX when a document contains at least one relevant result, the assessor must exhaustively score many other related elements in the document. To reduce assessment load some of these additional elements are automatically scored but the majority are assessed. The out-of-pool results are then added to the original pool. Unfortunately the process of exhaustive assessment is very time consuming. We argue that if the ranking of systems produced by an assessment set without exhaustive scoring is similar to the official INEX ranking, then assessment of out-of-pool results may not be necessary for all topics.
- Graded vs. Binary Assessment – In INEX, results are evaluated over 2 dimensions: exhaustiveness and specificity. In turn, these dimensions are scored over a range of 0-3. However, judging over these two dimensions is difficult and fraught with inconstancies between multiple assessors. However, binary relevance evaluation is much easier and quicker. In binary assessment each result is judged either relevant or irrelevant. We argue that if the ranking of systems produced by a binary assessment set is similar to the official INEX ranks, then two dimensional assessments may not be necessary.
- Miscellaneous issues:
  - Manual Runs – the pool of results may be enriched by through manual run submissions.
  - Re-using Topics – should topics from previous years be re-used? How?
  - XML Structure Changes to enrich the collection and the kind of tasks that can be performed by adding structure
  - Additional Tasks – What other tasks/sub-task could we undertake at INEX each year, to progress the state of XML-IR.

2. POOLING SUBMISSIONS

2.1 Background

In order to evaluate the systems we need a suitable baseline for comparison. In information retrieval we compare systems' results lists with a set of manual relevance assessments. In a sense, this allows us to compare system results lists with a results list from an 'ideal' system. This procedure allows us to produce standard recall and precision values for a system, and rank a set of systems according to these values. This approach has been followed since the early Cranfield experiments; however, several changes have been incorporated in order to scale to larger collections. Here we describe these changes, by comparing the methods used in the Cranfield experiments, to the methods employed by document retrieval experiments such as TREC, and finally the method used by INEX to handle structured information retrieval (Sections 4 and 5).

Early test collections (1960s, 1970s and early 1980s) such as Cranfield, were relatively small in size (less than 5MB). Since these collections contained a relatively small number of documents, human judgers were able to assess every document in the collection in relation its relevance to every query. With the emergence of TREC (1980s-current) much larger test collections (measuring in Gigabytes) became standard for laboratory information retrieval systems. Due to the large increase in collection size it was clear that the existing exhaustive method of evaluating every document in the collection was unfeasible. Therefore, a more scalable method of assessment was needed.

This challenge was handled by the use of system pooling, which was originally developed by Sparck-Jones and Van Rijsbergen [3]. The idea of system pooling is: for each topic, combine the top N results from each of the submission files. The results are merged, duplicates removed and are disassociated from their original submission. This becomes the system pool, and is sent to human judges for assessment. Results that are not in the system pool are automatically regarded as irrelevant. System pooling has proven to be an efficient means of evaluating systems, and has been used in several major international information retrieval workshops (for example, TREC, CLEF, NTCIR). Despite their proven worth, current evaluation methodologies have two shortcomings.

The first shortcoming is that the judges' decisions are inherently subjective. The notion of 'relevance' is at the very least a fuzzy concept, and people are bound to disagree on what constitutes a relevant result. Therefore, if two people are given the same set of results to judge, it is very unlikely that they will make exactly the same decision for every result in the set. The problem is even worse if relevance is judged on a graded, rather than binary scale. Incidentally, this is not a problem limited to pooling, and it could also occur with exhaustive assessment. However, research by Voorhees [4] concluded that while judges may disagree, the impact of their disagreement on systems ranking is not significant. The second shortcoming is that pooling inherently misses some relevant results. This is because all results ranked below the pool depth are automatically regarded as irrelevant. Research by Zobel [5] concluded that a system pool will only find about 70% of the relevant results in a collection; but, once again the impact of system ranking was not significant. However, it does raise the question of whether other, possibly more efficient or more effective pooling methods, could be used instead of system pooling.

2.2 Proposed INEX 2005 Experiment

Our proposal continues the work of Cormack et. al. [1] and Sanderson and Joho [2]. At present, the INEX Ad-hoc track uses a modified version of the Cranfield methodology that includes system pooling. The following six steps are undertaken annually:

1. Participants contribute topics (end user queries) and a subset of topics is selected for evaluation.
2. The topics are distributed to participants who run their search engines and produce a ranked list of results for each topic. The top 1500 ranked results for each topic are combined into a single submission file. Participates are allowed to send between 1 and 3 submissions, per task to INEX.
3. The top results from each submission are pooled together, disassociated from their originating submissions and duplicates are eliminated. We call this the s*ystem pool* (S*)* and say that it contains K_s results. We call the number of results taken from each result the *pool depth* (D_s) and it is currently set to 100.
4. The results in S are individually judged by the original topic contributors, who act as end users manually assessing the relevance of the results in terms of exhaustiveness and specificity. When judges find a document with a relevant result they must search the document for other relevant results, thus the size of S increases to $K_{s+i.}$ We shall refer to the results added to the pool as *inferred* results. We refer to the decisions made by the judges as *assessments*.
5. Using the assessment set and a standard evaluation module (inex_eval), the participating search engines are ranked in terms of performance (recall/precision) using several metrics.
6. Results are returned to participants who in turn write up and present their systems and discuss it at the workshop.

We propose replacing steps 3, 4 with the following.

3a. Produce a results pool (S) from the top N_s results from each submission in the usual manner. The pool depth N_s has to be determined in a certain manner and this is discussed later. We call this the s*ystem pool (S)* and say that it contains K_s results.

3b. In addition to the system pool, use a metasearch system to produce a merged ranked results list from all the submissions. From the list, select the top K_s results as a *metasearch pool* (M).

3c. Merge M and S (removing duplicates) to produce the *combined pool* (C) that contains K_c results.

Table 1: Metasearch Pool vs. System Pool

Assessments-Task	System			Metasearch		
	Average Precision/Topic	Average Recall/Topic	Average Unassessed/Topic	Average Precision/Topic	Average Recall/Topic	Average Unassessed/Topic
I-CO	0.131	0.471	125	0.146	0.507	383
II-CO	0.132	0.451	125	0.175	0.487	381
I-VCAS	0.208	0.440	10	0.224	0.460	211
II-VCAS	0.170	0.435	10	0.241	0.448	215

4. The results in C are judged by the original topic contributors, as if it was a traditional system pool. Again, inferred results are added to C, increasing its size to K_{s+i}

Submission evaluations are performed using the assessments in exactly the same manner as they were in previous years; so the assessors need not be aware of the source of the pool and scoring procedures need not change. The only problem that could arise is an increase in workload – only if the number of results in the combined pool is very large, since during judgments this would require much more work than the status-quo approach. However, by carefully controlling N_s, the pool depth, this can be avoided. We know that the size of the combined pool equal to the set union of the metasearch and system pools. In order to keep their weighting in the combined pool equal, we take the same number of results from both. However, we won't be able to predict the size of the combined pool since that will depend on the overlap between the system and metasearch pools. If the overlap is large, the size of the metasearch pool will be close to K_s, the size of the system (and metasearch) pool. However, if the overlap is small, the size of the combined pool will be close to $2K_s$, double the size of the system pool. The value of K_s can be easily chosen by experimentation since the process is an automated one. We may choose a value to limit the assessment workload rather than choose an arbitrary value.

After assessment is complete we will be able to determine which pool (M or S) has the higher level of recall. This will tell us which pooling method is the superior. If the metasearch pooling is superior then we could continue to use it in future INEX Workshops.

2.3 Preliminary Experiment

Before using metasearch pooling at INEX one would want to verify the validity of the approach. Therefore, we conducted a preliminary experiment to compare the performance of the proposed metasearch pool with the existing system pool. We conducted the experiment using the INEX 2004 submissions and both set of INEX 2004 assessments sets, and followed the proposed steps 3a and 3b in Section 2.2. The pool depth was set to 50 for the system pool to give us approximately 50% of the results that would be in a system pool of depth 100. In theory any metasearch method could be used to derive the metasearch pool, but we used the Borda Count approach. The Borda Count only requires a ranked list of results from constituent systems (that is - no relevance score per result) and it does not require any training. Evaluation of the pools was conducted as follows: For each pool, we calculated the total recall and precision values for each of the topics; then, we averaged the values across all topics. These averages are presented in Table 1, along with the average number of results not assessed. To produce the metasearch we used a pool depth of 500 results. We tested several pool depths (between 250 and 1500 results), but found that they all perform similarly. These results indicate that the metasearch pool is slightly superior to the system pool.

However, it must be noted that the assessment set is possibly/probably biased towards the system pool. This is because there were some results selected by the metasearch pool, which were not included in the assessments. Since these results were not assessed by a human judge they were automatically scored as irrelevant, even though in reality they could be relevant. Of course, the only way to know if these results are in fact relevant is to assess them, in the manner that we propose for INEX 2005. At the very least, our preliminary experiment has shown that the metasearch pool is as good as the system pool, with the possibility of out-performing it.

3. WHO CONTRIBUTES RARE RELEVANT RESULTS?

In the pooling method used in INEX results are added to the pool regardless of their originating system, even though some poor performing systems contribute very few relevant results (either at the element or document level) to the pool. There are two justifications for including results from poor performing systems in the pool. First, it keeps the pool unbiased, and removes the possibility of a 'self-fulfilling prophecy', whereby systems perform poorly because their retrieved results are not assessed. Second, even poor performing systems *may* find rare relevant results that are useful when added to the pool. But do we know for certain that poor performing systems find unique relevant results? If not, and if we can somehow identify poor system without completing the detailed manual assessment, should we not include their results in the pool (and include more results from better systems)?

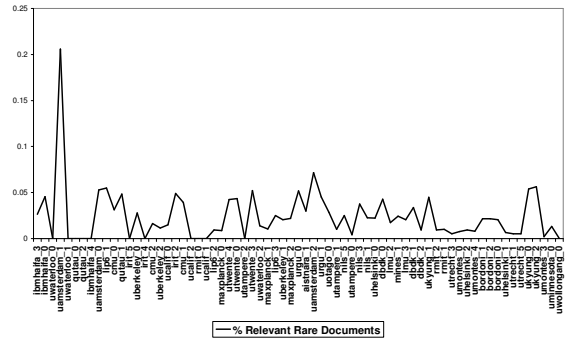


Figure 1: Percentage of Relevant Rare Results – CO – I

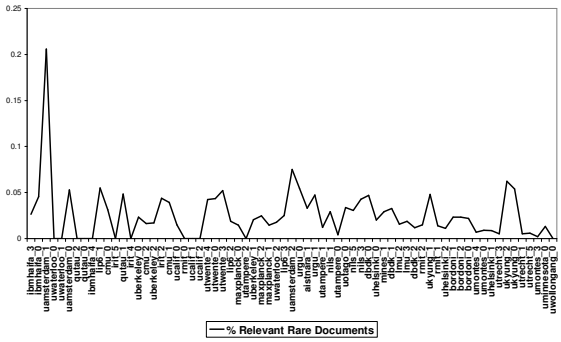


Figure 2: Percentage of Relevant Rare Results – CO - II

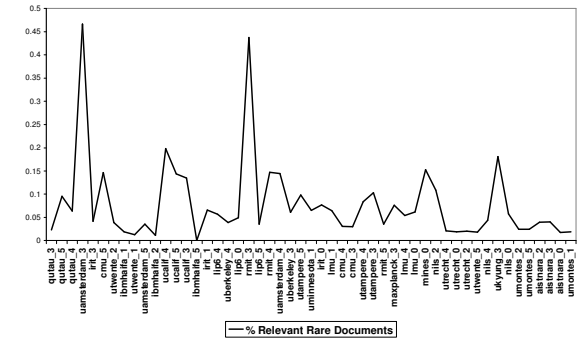


Figure 3: Percentage of Relevant Rare Results – VCAS - I

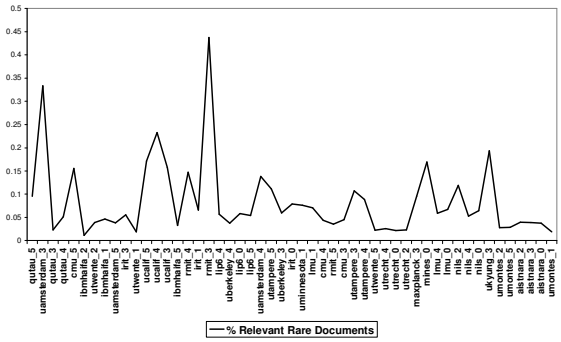


Figure 4: Percentage of Relevant Rare Results – VCAS - II

Here, we tested whether or not poor performing systems contribute a significant number of unique or rare relevant documents to the system pool. Originally we had planned to investigate the amount of unique results located by systems. However, the notion of "*uniqueness"* is clouded by the hierarchal nature of XML document, since we can not consider a result unique if its ancestor has been found by other system. For instance, imagine that a result **article[1]/sec[3]/p[5]** was only found by a one system (and therefore unique). However, if its parent node (**article[1]/p[3]**) was found by several or more systems can we then say that it is truly unique? We would say no, since the parent node obviously contains the child node. Furthermore, in INEX assessors when a relevant parent node is located assessors must judge each child node, arguably making the inclusion of the child node in the original pool moot. Hence, we concluded that for a system to have a unique result, neither the element nor any of its ancestors can a found by another system. And since the root ancestor node of all elements is the article node, in practice, this meant we where investigating the amount of unique articles/documents located by systems. However, after executing initial tests we realized that very few systems found unique document, therefore we extended our investigation to locate the amount of rare documents located by systems.

Out process was as follows: for each topic, we examined each system's top 100 results, examined their document name, and determined which documents were located by 5 or fewer systems. The number 5 was chosen as an estimate to what consists a rare document. This became our rare documents set (R). When formulating U, we only examined the top 100 results from each system because that corresponds to the pool depth used to derive the INEX system pool. We then classified each document in R as either relevant if it had a non-zero exhaustiveness or specificity value, and irrelevant otherwise. We conducted our experiments using both the CO and VCAS tasks and both 2004 assessments sets. Figure 1 – 4 are the plots, and for each system show the percentage of relevant rare documents. The systems are sorted according to each system's official INEX rank with the highest scoring systems on the left. As the results indicate there doesn't appear to be a correlation between a system's performance and the number of relevant rare documents. This indicates that it is valid to pool results from poor performing systems.

Table 2: CO Rank Correlations – Official vs. Out-Of-Pool

Assessment/Correlation	Aggregate	Strict	Generalized	SO	E3S32	E3S321	S3E32	S3E321
I-Spearman-rho	0.996	0.997	0.989	0.986	0.996	0.997	0.996	0.993
II-Spearman	0.995	0.996	0.990	0.990	0.996	0.997	0.996	0.989
I-Kendall-tau	0.965	0.968	0.937	0.936	0.962	0.977	0.964	0.953
II-Kendall-tau	0.960	0.960	0.947	0.948	0.960	0.973	0.965	0.942

Table 3: 2004 VCAS Rank Correlations – Official vs. Out-Of-Pool

Assessment/Correlation	Aggregate	Strict	Generalized	SO	E3S32	E3S321	S3E32	S3E321
I-Spearman-rho	0.989	0.983	0.995	0.992	0.994	0.994	0.984	0.987
II-Spearman	0.990	0.993	0.996	0.989	0.996	0.992	0.987	0.987
I-Kendall-tau	0.942	0.923	0.969	0.950	0.961	0.961	0.923	0.927
II-Kendall-tau	0.950	0.951	0.970	0.939	0.962	0.956	0.933	0.936

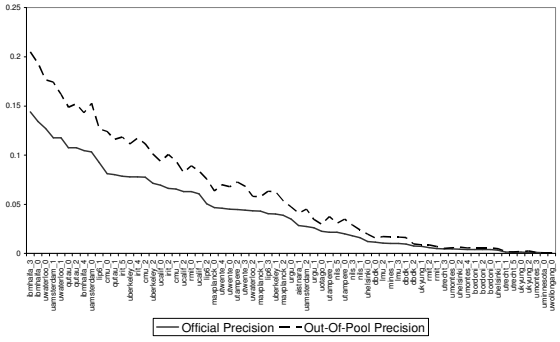


Figure 5: Official MAP vs. Out-Of-Pool MAP(Aggr) – CO – I

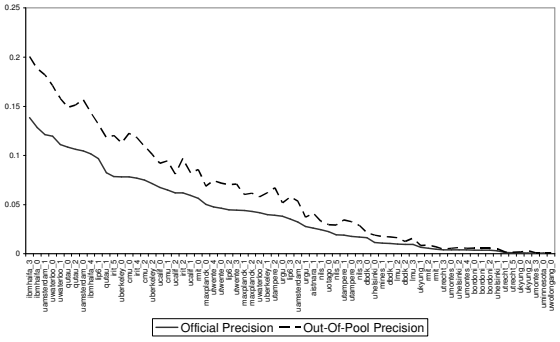


Figure 6: Official MAP vs. Out-Of-Pool MAP(Aggr) – CO - II

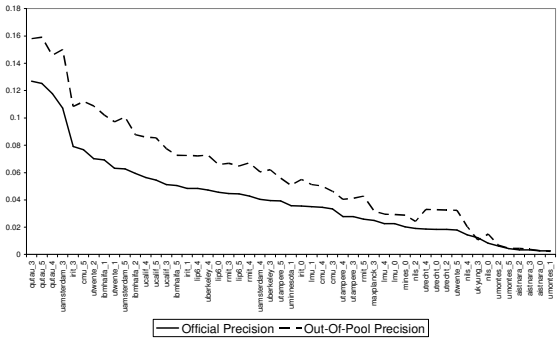


Figure 7: Official MAP vs. Out-Of-Pool MAP(Aggr) - VCAS - I

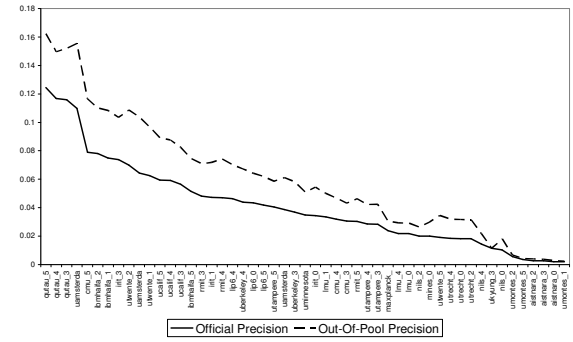


Figure 8: Official MAP vs Out-Of-Pool MAP(Aggr) - VCAS - II

4. ARE OUT-OF-POOL RESULTS USEFUL?

During the INEX evaluation phase, if a document contains one or more relevant results, the assessor must examine all the other elements in the document, and individually assess each for relevance. Assessors can also add incidental results that are picked up by inspection. These results are then added to the original pool as 'out-of-pool results'. The justification for this is two-fold. First, the results may have been identified by systems, but at below position 100 thus escaping the system pool. Second, it will help to 'future proof' the evaluation set since judges may locate results beyond the capabilities of current search engines, which may be found by future, more sophisticated search engines. We do not dispute the validity of these motivations; however, the process has one major drawback - it is very time consuming. We already know that the INEX evaluation process takes a lot longer than comparable workshops. We believe that by removing the evaluation of out-of-pool assessment, thereby having judges assess only returned results, we could greatly reduce the time required for assessment. However, there is a risk involved in not assessing all AAAelements: the rank of systems may significantly change when inferred results are not included in the assessment pool. Here, we investigate this hypothesis, by producing a rank of systems inferred results are not included in the assessment pool. Here, we investigate this hypothesis, by producing a rank of systems using the original pool, without the inclusion of 'out-of-pool' results. We then compare this systems ranking to the official systems ranking. We argue that if the system ranks are similar, then out-of-pool assessment may not be necessary.

We conducted our experiments in the following manner. First, we parsed the INEX 2004 assessments and removed all the out-of-pool results. This allowed us to have a set of results consisting of the original pool. Then we executed the inex_eval using the original pool assessments, and produced a ranked list of systems. Tables 2 and 3 present the correlation between the two. Table 2 is the correlation between ranks for the INEX 2004 CO Task and Table 3 is the correlation between ranks for the INEX 2004 VCAS Task. We compared systems using both evaluation sets and metrics used in INEX 2004. We used two correlation measures: Spearman-rho and Kendall-tau. Figures 9-12 plot the Aggregate Mean Average Precision (MAP) for participants in the CO and 2004 VCAS tasks, using both sets of assessments. The two systems rankings are very similar. This indicates that the assessment of out-of-pool results is not vital for accurately discriminating between XML-IR systems; this raises the possibility of having judges only assess results in the original Ad-hoc pool. However, even if we choose to keep out-of-pool results in the Ad-hoc task, current or future tasks/tracks may choose to eliminate the process without significantly impacting on the ranking of systems.

5. GRADED VS BINARY RELEVANCE ASSESSMENTS

The objective of XML-IR is two-fold. First, systems must find XML elements (results) that match the subject area specified in a user query. Second, systems must choose the most appropriately sized elements to return to the user, and rank accordingly. To correspond with this dual retrieval objective, INEX has extended the notion of relevance to cover two dimensions - exhaustivity and specificity**.** Each dimension is judged as one of four values from zero to three, where zero is judged as irrelevant. Also, an element cannot have a zero score in one dimension and a non-zero score in another. This produces nine possible levels of relevancy, plus a single non-relevant level. In contrast, most document-level evaluation methods classify documents as relevant or non-relevant.

In theory, INEX's use of two dimensions, and graded scaled makes sense, since we assume that as one propagates up an XML tree, the values for the two dimensions will change. The observation is that since ancestor nodes contain a larger amount of information, they tend to be more exhaustive than descendants. Conversely, relevant descendant nodes tend to be more specific than their ancestors, as they contain less irrelevant information. The graded INEX evaluation process is very time consuming and prone to great disagreement between multiple judges. However, it should be much easier and quicker to judge result relevancy on a binary scale (that is - as either relevant or irrelevant). Here, we investigate this hypothesis by producing a ranked list of systems evaluated using binary assessment, and comparing it with the official INEX systems rank. We propose that if the two systems rankings are similar, then quantized assessment may not be necessary.

We conducted our experiment in the following manner. First, we parsed the INEX 2004 assessments and changed the value of every non-zero score exhaustiveness or specificity score to 3/3. This allowed us to simulate binary relevance. Then we executed the INEX evaluation module (inex_eval) using the binary assessments, and produced a ranked list of systems. Tables 4 and 5 presents the correlation between the two systems ranks. Table 4 is the correlation between ranks for the INEX 2004 CO Task and Table 5 is the correlation between ranks for the INEX 2004 VCAS Task. We compared systems using both evaluation sets and metrics used in INEX 2004. We used two correlation measures: Spearman-rho and Kendall-tau. Figures 9-12 plot the Mean Average Precision (MAP) for participants in the CO and 2004 VCAS tasks, using both sets of assessments. The results show that the two systems are similar, but significantly different. This indicates that graded assessment is important for accurately discriminating between the performances of XML-IR systems. This validates INEX's choice of using graded results for its Ad-hoc task. However, since the systems ranks are reasonably similar, particularly for the Generalized and SO metrics, it raises the possibility of using binary relevance in situations were time is a major constraint (such as the interactive track).

Table 4: CO Rank Correlations – Official vs. Binary

Assessment/Correlation	Aggregate	Strict	Generalized	SO	E3S32	E3S321	S3E32	S3E321
I-Spearman-rho	0.957	0.882	0.988	0.973	0.928	0.917	0.893	0.909
II-Spearman	0.950	0.862	0.985	0.970	0.937	0.932	0.902	0.928
I-Kendall-tau	0.875	0.788	0.940	0.901	0.837	0.820	0.789	0.812
II-Kendall-tau	0.862	0.788	0.933	0.893	0.850	0.837	0.790	0.819

Table 5: VCAS Rank Correlation – Official vs. Binary

Assessment/Correlation	Aggregate	Strict	Generalized	SO	E3S32	E3S321	S3E32	S3E321
I-Spearman-rho	0.961	0.914	0.996	0.983	0.900	0.901	0.960	0.969
II-Spearman-rho	0.957	0.888	0.986	0.986	0.877	0.874	0.952	0.962
I-Kendall-tau	0.875	0.811	0.963	0.921	0.796	0.803	0.867	0.889
II-Kendall-tau	0.862	0.778	0.925	0.926	0.765	0.760	0.858	0.879

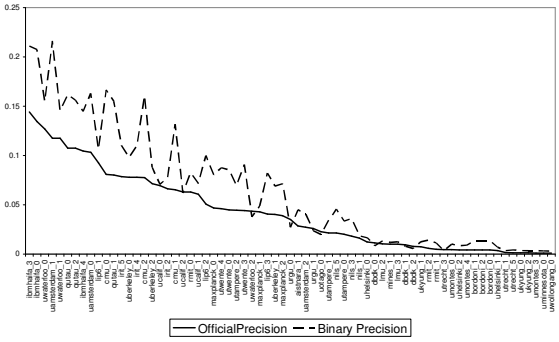


Figure 9: Official MAP vs. Binary MAP – CO –I

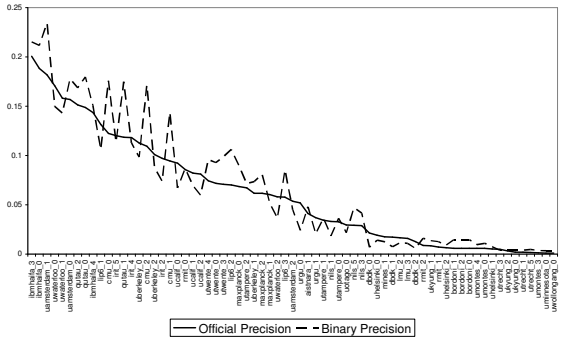


Figure 10: Official MAP vs. Binary MAP – CO -II

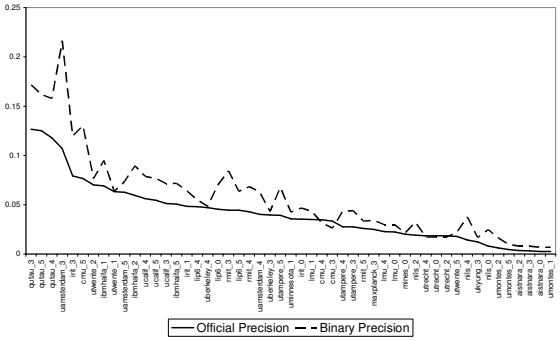


Figure 11: Official MAP vs. Binary MAP – VCAS -I

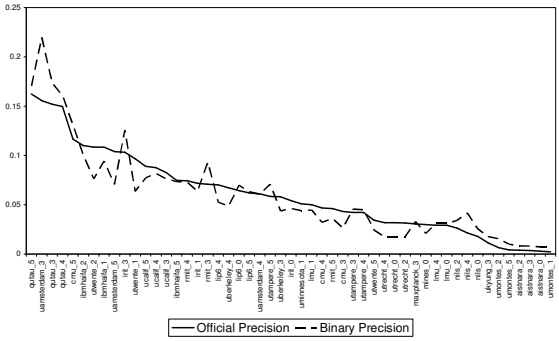


Figure 12: Official MAP vs. Binary MAP – VCAS -II

6. MANUAL RUNS

Even if we are successful in expanding the proportion of relevant results in the results pool through metasearch, it is still limited by the ability of the search engines to automatically find results. It is possible to increase the size of the results pool by including the results of manual runs to the pool. Manual run results can be performed either through a semi-automated relevance feedback process, or through more elaborate manual intervention of assembling relevant result sets. In Semi-automated mode participants evaluate results and provide relevance scores for the top N results. The search engine then automatically utilizes the feedback to modify the search strategy (for example - by adding/removing keywords), by changing the ranking strategy (for example - by re-ranking results through a change in the scoring parameters), or both. In the more elaborate manual mode users can change the query in any way desired through iterative use of the search engine and by manually eliminating irrelevant results and re-ordering results (that is - manual ranking).

There are two ways in which evaluation of systems can then take place. The evaluation of automatic runs can be carried out as in previous INEX workshops. The important contribution of manual runs is in providing a result pool that is closer to the 'absolute' results pool, or a baseline, against which to compare automatic results. Then there is also benefit in comparing the average performance of automatic runs with the average performance of manual runs. This average performance can be computed on the best N performing systems, or by comparison that is based on a metasearch pool that is obtained as described in section 2. There is another comparison that could be made – between manual runs - but there is a problem: the quality of the manual submissions depends critically on the competence of the persons who use the search engines. Although it is possible to conduct controlled experiments that will attempt to eliminate this problem, there seems to be no simple enough way other then by involving numerous users, each of whom will be required to use several different systems. This seems infeasible under the INEX mode of operation and resource constraints.

7. RE-USING TOPICS

Past topics can be very useful for at least two reasons. The most obvious reason is of course the reduced assessment load. The second reason is the ability to quantify the improvement in search engine technology over time. Re-used topics should lead to result pools that include additional relevant results. Of course there is no need to re-assess the entire result pool of a re-used topic. Previously assessed results can be assigned the known scores. This leaves a smaller residual result pool for assessment. There is always a risk, when re-using topics, that search engines that were designed with the use of past assessments, are over-fitted to those assessments. However, this can be tested by comparing the performance of systems over re-used topics with the respective performance over new topics. This evaluation can reveal whether this is a problem that is specific to some systems, to all systems, or perhaps to none. If we discover that over-fitting does not occur at significant levels then we can re-use topics with confidence in future workshops.

8. XML Structure Changes

There are several generic XML DTD changes that we would like to propose and that we believe will enrich the INEX collection, the type of tasks that may be pursued, and possibly improve the results that can be obtained. The changes may also assist in result assessment and run evaluation processes. These proposed changes are discussed below

8.1 Text Segmentation.

Segmentation can always be applied to text type elements. For instance, we may wrap sentences within XML tags <s> ... </s>. This can be useful in several ways. Question answering tasks that require highly specific responses can benefit from the ability to pinpoint relevant sentences. Furthermore, in evaluation it may be useful to be able to assess individual sentences as relevant. For instance, sometimes only a small part of a long paragraph is relevant and at present there is no way to assess at below the paragraph level.

8.2 Part of Speech Tagging (POS)

Part of Speech (POS) tags can be added with fairly high accuracy to the collection. State of the art POS taggers are claimed to operate with accuracy of better than 95%. Apart from being useful in supporting NLP functionality POS tagging can be very useful in facilitating very simple selectivity in term searching and can probably assist in improving overall accuracy – merely by adding some elementary semantics to index terms. By adding POS tags attributes of sentences in the collection all participants at INEX will be able to use such information – or ignore it – and it should foster greater interest in using POS techniques in IR.

8.3 XPointers and XLinks

A standard mechanism of referencing, namely XPointer and XLink, exist in XML but are not used within the INEX collection. It is possible to convert the collection to support XLinks and XPointers and we propose to pre-process the INEX collection and augment it as follows:

- Replace references within the article body, pointing to bibliography entries in the References section, by XPointers. Furthermore, insert XPointers in each bibliography entry pointing back to each element in the article that references that bibliography entry.
- Replace references within the article body, pointing to figures in the article, by XPointers. Furthermore, insert XPointers in each Figure pointing back to each element in the article that references that figure.
- Some bibliography entries in the INEX collection refer to other articles within the INEX collection itself. Insert

XLinks in each bibliography entry that makes such a reference.

These additions can simplify processing of common operations such as composing a response for a query from multiple relevant components that are interlinked. For instance, it would be possible to easily support queries that in the past were excluded at INEX, such as "Get figures of the CORBA architecture together with the relevant text that explains it".

8.4 Element Size attributes

We could augment each element with a set of size attributes. These attributes could facilitate various operations in searching and in ranking results, as well as in assessing results. The following size elements could be considered:

- Number of children (C).
- Number of descendents (D)
- Number of Sentences (S)
- Number of words (W)

For instance, <sec C="7" D="43" S="234" W="1432"> ... </sec> indicating that the section has 7 children elements and 43 descendents, 234 sentences, and 1432 words in total. This information can be omitted from very small elements as it is unlikely to be useful.

This information can be used, for instance, in determining the size ratio of relevant to irrelevant parts of a given XML component. It could be used in evaluating the specificity of a result element for the purpose of ranking and also for the purpose of automating an assessment tool for INEX.

9. ADDITIONAL TASKS

The performance of systems over the INEX Ad-hoc task varies greatly. Some of the performance differences can be attributed to specific system characteristics. It is usually impossible to assess precisely which properties of a given system are responsible for its performance (or lack thereof…). Contributing factors can be superior indexing structures, the use of insightful heuristics, rigorous analysis, a user model that is faithful to some general traits of assessors, and so on. Many of these are embedded deeply and implicitly within search engines and therefore the ranking of entire systems is the only way for us, as a community, to assess the merits of individual approaches. We would like to explore particular aspects of search engines in isolation. In the following we propose a few tasks that might help us achieve this. Importantly, none of these tasks require any additional assessments – evaluations can be fully automated using the standard ad-hoc track assessments.

9.1 Query Expansion sub-task

As a pre-text to our suggestion, we would like to briefly look at the Natural Language Query (NLQ) task. One of the sub-tasks is the translation of a description element into a NEXI title expression. The idea is to evaluate all the NLQ approaches using one or more baseline search engines. In this manner, any performance variation can be attributed to superior translation of a query into a NEXI specification, rather than to any inherent property of a particular search engine. It is possible to isolate the NLQ contribution from the contribution of the implementation of searching and ranking.

We would like to propose the extension of this approach to query expansion. One of the critical success factors in query evaluation is query expansion. Most queries are expanded by the addition of terms, and in the case of CAS queries by the addition of tags, to the original query. In order to isolate the contribution of query expansion, from the contribution of the searching and ranking processes that follow, we propose the following task. In similar manner to the NLQ task, given a set of INEX topics, produce a set of expanded topics. Each NEXI topic in the original set is expanded – transformed into a new NEXI expression. The submission thus consists of a set of new topics rather than the submission of results from retrieval runs. All the expanded sets of topics will then be run on one or more of the better performing search engines and ranked through their indirect performance with the baseline search engines. The relative ranking can thus be attributed to query expansion rather than to the underlying search engines.

9.2 Ranking-Only sub-task

A natural extension to this approach is a ranking-only sub-task. Given a topic and a bag of results (that is - an unordered set, possibly derived using a metasearch technique as discussed in Section 2), the task is to rank the results. The task here is solely the ordering of results. It requires scoring and ranking only and therefore does not require the implementation of a search engine. This task may provide greater insights into which ranking and scoring strategies work better, in isolation from query expansion and indexing/searching strategies. It is not obvious how to separate the functions, but if it is possible than this would be a worthwhile investigation.

9.3 Ontology Mining

Currently, most search engines accept a list of terms, or reduce a natural language sentence to a list of terms by 'cleaning up' noise words. Search engines typically use query expansion techniques (for example - addition of synonyms or related terms) to explicitly augment the implicit correlation between query terms. It is difficult to do this in a user-free context since different users may benefit from expanded queries in different ways, depending on individual interests and contexts. Furthermore, query expansion may be context dependent in itself. The same query terms may be closely associated with one set of terms in one context and with a completely different set of terms in relation to another context. The WordNet ontology is perhaps the best known example; however, it is very generally language oriented rather than collection specific.

Identifying sets of terms for distinct contexts is a difficult problem. The term 'Ontology' is understood in this context to mean a thesaurus that can identify the use of related terms in different specific contexts. Unlike an ordinary thesaurus, which is language based, very general, and not context sensitive, ontology has higher granularity and is context sensitive. In this task we would like to study techniques for mining ontologies from XML document collections. The aim is to automatically construct and maintain ontologies that capture the possible semantic information in XML documents, including term taxonomy, 'interesting topics', frequent terms and phrases, associations between terms and phrases, and so on.

Of course it is impractical to generate domain ontologies manually. So the trick is to take a large collection and to perform data mining operations to discover associations, co-occurrences, similar uses, and so on. This is not new - there is a lot of research in ontology mining. However, the XML collection potentially offers us much richer semantics to create associations with. Rather than merely word proximity we now have terms appearing together in <keywords> elements, or in <abstract> , <author>, <biography>, <theorem>, and so on. It should be possible then to take advantage of this rich semantics in mining ontologies. But how can we identify and quantify any potential improvement?

We propose to study Ontology Mining in XML collections in the INEX context. The task that we propose is closely related to the task described in section 9.1, Query Expansion.

Given the INEX collection of 18 Journals and Magazines:

- Automatically generate a comprehensive ontology from the XML collection
- Given a set of topics (queries) expand the queries with related terms derived from the ontology – that is, produce an augmented set of queries

The idea is to evaluate ontology mining systems through their utility in query expansion - we use a set of standard search engines to evaluate the original and the expanded queries and we measure the improvement (if any). The baseline measurement is the performance of the standard search engines with the original queries. The expanded queries are also executed by the same baseline search engines. If the ontology is accurate, and if there is an advantage to query expansion by obtaining more comprehensive and accurate results, then we can rank approaches to ontology mining by the amount of improvement.

10. SUMMARY

Here we presented several ideas that that could be incorporated into INEX. Some proposals relate to new tasks or extended functionality and others to different assessment procedures. We believe that it is possible to obtain an increase in evaluation efficiency by trading off evaluation effectiveness. Regardless, we feel that the proposal will lead to spirited discussion and debate at the INEX Workshop on Element Retrieval Methodology and with respect to IR in XML in general.

11. ACKNOWLEDGMENTS

We wish to acknowledge the direct and indirect contribution of all the participants of the INEX workshops and the various INEX mailing lists for raising and discussing many of the issues that we address in this document.

12. REFERENCES

[1] G. V . Cormack, C. R. Palmer, and C. L. A. Clark, "Efficient Construction of Large Test Collections", In *Proceedings of The 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* ACM Press, Melbourne, Australia, 1998

[2] M. Sanderson, H. Joho, "Forming Test Collections with no System Pooling", In *Proceedings of The 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Sheffield, Great Britain, 2004, pp. 33-40.

[3] Sparck-Jones, and C. J. Van Rijsbergen, "Report on the Design Study for the 'Ideal' Information Retrieval Test Collection", *British Library Research and Development Report 5428*, Computer Laboratory, University of Cambridge, 1975.

[4] E. M. Voorhees, "Variations in Relevance Judgements and the Measurement of Retrieval Effectiveness", In *Proceedings of The 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Melbourne, Australia, 1998, pp. 315-323.

[5] J. Zobel, "How Reliable are the Results of Large Scale Information Retrieval Experiments", In *Proceedings of The 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Melbourne, Australia. 1998.

Query Formulation for XML Retrieval with Bricks

Roelof van Zwol, Jeroen Baas, Herre van Oostendorp, and Frans Wiering

Centre for Content and Knowledge Engineering
Utrecht University
Utrecht, the Netherlands
roelof, ajbaas, herre, fransw}@cs.uu.nl

ABSTRACT

XML retrieval, also referred to as Structured Document Retrieval is a discipline of information retrieval that focusses on the retrieval of relevant document fragments for a given information need that contains both structural and textual components.

In this article we will focus on the theory behind Bricks, a visual query formulation technique for XML retrieval that aims at reducing the complexity of the query formulation process and required knowledge of the underlying document structure for the user, while maintaining full expression power, as offered by the NEXI query language for XML retrieval.

In addition, we present the outcome of a large scale usability experiment, which compared Bricks to a keyword-based and a NEXI-based interface. The results showed that participants were more successful at completing a number of search assignments using Bricks or NEXI. Furthermore, we observed that the participants were also able to successfully complete their assignments in a significantly shorter period of time, when using the Bricks interface.

1. INTRODUCTION

The recent popularity of XML retrieval, or Structured Document Retrieval, is caused by the widespread use of the eXtensible Markup Language (XML) in digital libraries, intranet environments. Contrary to the well-know Internet search engines, XML retrieval systems try to exploit the structure of the documents during the retrieval process.

For XML retrieval systems to work in practice, it is crucial that users are capable to adequately used the structure of a document in all facets of the retrieval process. Not only do we need good retrieval strategies, but the offered functionality should correspond to the user's need.

In this article the focus is on the query formulation process for XML retrieval. The large scale search engines that are available on the Internet allow easy access to large quantities of information that is available on-line. Using a few keywords a user can formulate his information need and retrieve a list of relevant documents, which needs to be browsed for the relevant information. This approach is satisfactory for most users, but for digital libraries and large intranets, where the information need is usually more specific and large amounts of information on the subject is available more sophisticated query formulation techniques are desired.

Current approaches in XML retrieval allow a user to either specify his information need using keywords (content only), or by using a combination of structural constraints and keywords. This is formalized in the NEXI query language [14], where a user can specify his information request through an XPath-like expression [4], that combines both the structural and content-based aspects of the user information need.

Using such a query language for the retrieval provides powerful expression mechanisms, but also has its impact on the query formulation process. The user should be able to express his information need using the syntax of the query language, and in addition the user should have knowledge of the structure of the document.

Consider the information need of Example 1, where a user visiting the Lonely Planet Web-site wants to:

Example 1
Find historical information about ***revolutions*** *for destinations with a* ***constitutional monarchy*** *as government.*

Using a (NEXI-CO) content-only approach, the user is likely to use the following keyword combination to formulate his information need:

history revolutions destination government "constitutional monarchy"

Without any path directives in the information request a XML retrieval system can literally retrieve any document fragment that contains one or more of the given terms. For example, this can be a piece of text that is emphasized, or the entire document.

Taking a closer look at the information need, we can see that the objective is to retrieve historical information. Furthermore suppose that the user is familiar with the (semanti-

cal) structure of the document collection, he is then able to identify the structural conditions of the information need. In Example 1 the structural conditions of the information request are underlined, while the emphasized terms form the content-based aspects of the information need. If we make the transition of the information need to a formal specification, we will end up with the following NEXI content and structure (NEXI-CAS) query:

//destination[about(.//government, "constitutional monarchy")]//history[about(., revolutions)]

This NEXI query consists of two parts, a request query and a support query. The request query specifies the type of document fragment that should be returned by the system:

//destination//history[about(., revolutions)],

while the support query is used to specify additional conditions that should be met:

//destination[about(.//government, "constitutional monarchy")]

A NEXI-CAS query always consists of a request query that has a request path and a filter with one or more *about*-clauses. The request path specifies the desired element of retrieval, while the filter is used to specify the structural and textual conditions. Each about-clause has two arguments, a path directive and a list of terms. The path directive specifies where within the request path to search for the specified terms. Similarly the support query consists of a support path and filter that can contain one or more about-clauses.

The NEXI query language provides exactly the necessary expression power for XML retrieval. Although the syntax of the NEXI query language is relatively simple, a user needs to learn the syntactical features. This makes it hard, if not impossible, for the average user to express their information need in NEXI.

To overcome these limitations we have developed *Bricks*, a visual query formulation technique for XML retrieval that aims at:

1. Reducing the complexity of the query formulation process.
2. Reducing the required knowledge of the document structure.
3. Maintaining maximum expression power, as offered by the NEXI query language.

To realize this, Bricks uses a graphical approach that allows the user to specify his information need using small building blocks ('bricks'), starting with the specification of the desired element of retrieval. As a result, Bricks is guiding the user in a more natural way through the query formulation process. Not only does it solve the syntactical formulation issues, it also prevents possible information overload, when the document structure is large and complex. This is realized by using a priority for the different document elements. Elements with a low priority are not visible for the user early in the query formulation process. In Figure 1 the information need of Example 1 is expressed with Bricks.

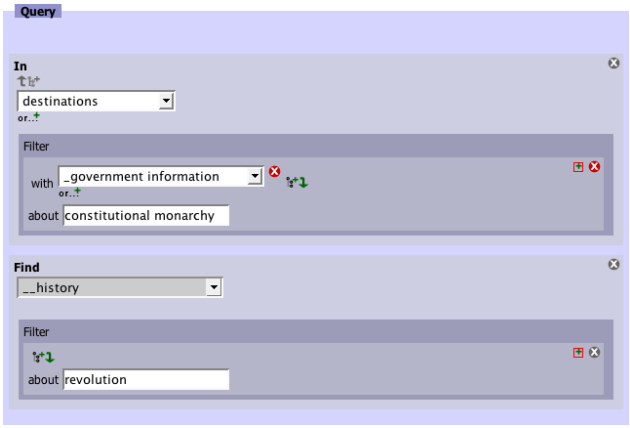


Figure 1: Example information request in Bricks

To validate our ideas, we have designed and implemented the Bricks interface on top of the XML retrieval system that was developed for participation in INEX, the INitiative for the Evaluation of XML retrieval [6]. INEX provides an international platform for the evaluation of XML retrieval strategies, allowing researchers to measure the retrieval performance of their system.

Finally we have set-up and performed a usability experiment to evaluate our ideas. In the experiment, we have compared Bricks with a keyword-based (NEXI-CO) approach and a 'content and structure'-based (NEXI-CAS) approach. We will briefly discuss the outcome of the experiment in terms of effectivity and efficiency.

1.1 Organization

In Section 2 we discuss the related work on structured document retrieval, and in particular on query formulation techniques. It also provides additional background information on the experimental setting that was used to evaluate our theory on structured query formulation. The theoretical foundation for Bricks is then discussed in Section 3. We then present the outcome of the usability experiment in Section 4. Finally, we will come to the conclusions and discuss future research in Section 5.

2. RELATED WORK

In general an information retrieval system consists of three components: a query formulation interface, a retrieval strategy (engine), and an interface for the result presentation. Below we will discuss the impact of and various research approaches on structured document retrieval for each of the three components.

2.1 Query formulation

The research on query formulation, presented in this article is using the NEXI query language as a starting point. NEXI [15, 14] is an XPath-based query language that primarily focusses on the extraction of relevant information,

using a combination of path directives and content-based filters. This makes NEXI an excellent query language for XML retrieval, providing a powerful expression mechanism to the user.

Alternative XML query languages, such as XQuery[3] and XSLT[1], do not focus on the retrieval task. They provide additional functionality that lays outside the scope of structured document retrieval, like for example transformations on the extracted XML document structure.

A more trivial query formulation technique is adopted by Lucene [7]. Information that is found within a specific field, i.e. an XML element, can be specifically targeted, like:

government: "constitutional monarchy"

The downside of this approach is that it is not possible to retrieve anything other than the document containing the requested field and content, or to specify more complex paths.

Of course one should not neglect the power of keyword-based information retrieval. It is still the driving force behind all popular search engines, and allows literally anyone to specify his information need with just a simple keyword combination. The NEXI query language therefore allows for the specification of keyword combinations, including the usage of phrases. This is referred to as NEXI-CO (Content Only), while a NEXI query that contains a path specification is referred to as a NEXI-CAS query (Content and Structure).

The Bricks query formulation technique uses a graphical approach. In [12] a method for replacing a complex command language syntax is discussed, called direct manipulation. By using a graphical interface the syntactical formulation of the query is represented by graphical items. This allows the user to successfully execute complex queries on a data structure.

In our prior research on schema-based structured document retrieval we have developed the Webspace method [16, 17]. There, we have shown that the retrieval performance can be improved by presenting the user a graphical interface that visualizes a (database-oriented) schema representing the semantical structure of the document collection. The user then formulates his information need in a materialized view on the schema. The approach followed for Bricks is schemaless and uses path directives to derive the requested information.

2.2 Retrieval strategy

The success of a XML retrieval system also depends heavily on the retrieval strategy. It executes the (structural) information request and derives a ranked list of relevant document fragments. In INEX, the INitiative for the Evaluation of XML Retrieval [6] the retrieval performance of XML retrieval strategies is evaluated. Participating in INEX allowed us to develop, evaluate, and improve various retrieval strategies [19] for XML retrieval. For the evaluation of Bricks and the other query formulation techniques discussed here, we have used our best-performing retrieval strategy.

Within INEX a number of user-related issues are being discussed. With respect to query formulation it is the question whether the structural conditions of the information request should be strictly interpreted, or whether these conditions should be seen as merely hints of where the user expects to find the relevant information. This is also referred to as the vague interpretation [9]. For our experiment, we have used a semi-strict interpretation of the path directives, which penalizes relevant document fragments that do not exactly fulfill the structural conditions of the information request.

Another issue within INEX, refers to result presentation. It deals with the question what the most specific and exhaustive element of retrieval is for a given information need. As a result it is possible that the list of document fragments returned by the retrieval strategy contains overlapping results [8]. When an XML fragment is considered relevant, its parent is by definition also relevant, and probably more exhaustive. From a user perspective, however, it is undesirable to have redundancy in the ranking of the document fragments.

2.3 Result presentation

Since relatively small document fragments are derived by the system, it is possible to use alternative techniques to present the retrieved information to the user. This is also the scope of the INEX interactive track [13]. There the interaction of the user with a result presentation interface for structured document retrieval has been evaluated. Using a content-only approach for query formulation, they were able to analyze user behavior with the presentation interface.

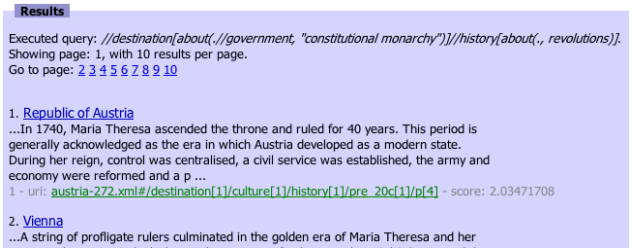


Figure 2: Snapshot of result presentation interface

For our research we use a commonly accepted presentation technique that provides a link to the relevant fragment, a short summary of the fragments content, and some additional statistical information that help the users to judge the relevancy of the retrieved information. Figure 2 shows a snapshot of the presentation technique used for our research. Nearly all main search engines available use this presentation format, therefore we can safely assume that the result presentation is not of significant influence to the result of our experiment.

3. THEORETICAL FOUNDATION

The theoretical foundation for Bricks can derived from the three objectives that are identified:

1. Minimize the complexity of the query formulation process.

2. Minimize the required knowledge of the document structure.

3. Maximize the expression power as provided by the NEXI query language.

Based on these objectives the follow design principles can be obtained that together form the theoretical basis of Bricks.

3.1 A graphical approach

The use of a graphical interface reduces the burden of syntactical formulation issues that are related to the NEXI query language. Although NEXI uses a relatively simple syntax based on XPath, it still allows users to submit malformed queries to the retrieval system. Apart from incomplete queries, this is not possible with the graphical approach adopted by Bricks. This is referred to as direct manipulation of the query language [12].

Furthermore, the underlying structure that is present in the document collection can be integrated into the query interface. Several approaches are thinkable, but for Bricks we have chosen to work with pull-down lists, allowing the user to select structural elements into the query. Alternatively a tree-based approach can be used to visualize the structure to the user. However, this is a more complex structure that needs to be interpreted by the user.

3.2 Intuition of a mental model for query formulation

When formulating a specific information request, the user has a mental model of the information he is looking for. Research on information seeking behavior [10, 11] has shown that users develop such a mental model, and that the effectivity of the task performance can be increased if the interface and offered functionality is closely related to the mental model of the user. When focussing on query formulation for structured document retrieval the task is more complex, since the user has to specify what the structural and content-based conditions of his information need are. If a user is asked to express his information need in natural language, he is likely to formulate a sentence like: "*Find historical information about revolutions, for destinations ...*".

A logical first step is to specify the requested element of retrieval, "*Find historical information*". From there a limited number of iterative steps are possible. The user either specifies a content-based constraint, "*about revolutions*", using the filter that is associated with the request path, or adds additional path directives to the request path, "*, for destinations*". If needed the user can add one more content-based filter, and simultaneously introduce a support path to the information request. This allows the user enough 'freedom' to follow his intuition, and to perform intermediate checks on the specified information request.

3.3 Step-by-step formulation of the information need: the building blocks

Bricks uses small building blocks to formulate the information request (query). Each block represents a small step in the formulation process, that needs to be completed, before another block is added to the query. After specifying the requested element of retrieval, the user can add an *about* clause to the request filter, or specify additional path directives to the request path. Adding an about clause allows the user to specify a content-based constraint, and to descend further down the document structure.

Adding an additional path directive allows the user to go up in the tree, this is referred to as the support path. Another block is added to the query for each step that is taken by the user. Based on the document structure and the syntax of the NEXI query language, the possible actions are controlled by the NEXI interface. This prevents the specification of malformed and unmeaningful (with respect to the document structure) queries. On the other hand we aim at preserving full expression power, as offered by the NEXI query language. In the next section, we will show how a nested object structure of a NEXI query is constructed with Bricks, to prove that we are able to achieve this.

3.4 Avoiding information overload

It is important that the user is not overwhelmed with options and possible next steps. In a sense, the intuition of a mental model is one approach to avoid information overload. Using a wizard-based approach, is a proven technique to reduce the learning curve of a task that needs to be accomplished. However expert users can experience a limitation in the provided functionality, causing them to get frustrated [5, 18]. In our case, we are not focussing on the high-end experts, such as programmers and database administrators, but on users with a complex information need that goes beyond the average profile of a user on the Internet. Although Bricks is more flexible than a wizard-based approach, the aim is similar: by reducing the number of options that are available, it becomes easier to complete (more efficient) the query formulation task.

In an attempt to reduce the required knowledge of the document structure, Bricks provides lists of structural elements that allow the user to select path elements into their query. However, the Lonely Planet XML document collection contains 271 unique element and attribute names. This can easily cause an information overload for the user, and cause the efficiency of the task performance to drop. When inspecting the structural elements, it becomes apparent that not all elements are meaningful from a retrieval perspective. For instance, the retrieval of a highlighted (italic) text fragment, containing just a few keywords, will probably not satisfy the user's information need, since all context is missing.

In general, it is possible to define a structure for the document collection that consists of three layers, as is presented in Figure 3. The top layer is formed by a semantical markup that provides a high level description of the content that is contained. The middle layer provides a logical markup, containing elements that have a logical function/meaning to the user. I.e. a chapter and its sections form logical containers of information. At the bottom layer the presentation markup is found, which is used for visual layout and presentation of the content. Any XML document can be seen as a tree. When using such a three-layer structure, the semantical element will naturally appear in the top of the tree, while the presentation element as usually found near the leafs to the tree. The mid-section of the XML document will then

contain the logical elements.

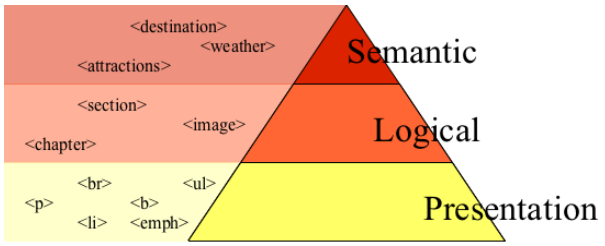


Figure 3: Three layer structure for XML document collections: semantical-, logical-, and presentation markup.

Bricks exploits this three layer structure in the retrieval process by adding a priority to each of the structural elements. Semantical elements will receive a high priority, followed by the logical elements, while the presentation elements are given a low priority. Early in the query formulation process, only the high priority elements can be selected in the query. Elements with a lower priority will become available once the user has made a first selection of the elements that should be retrieved. In a sense the user is traversing down the tree structure of the document collection, and narrowing down the possible elements that can be added to the query.

In practice a threshold of 20 elements is used, limiting the number of structural elements that can be presented to the user at once. Alternative presentation techniques with sublists are possible, to allow the user to explore a larger set of structural elements that can be included in the query.

4. USABILITY EXPERIMENT

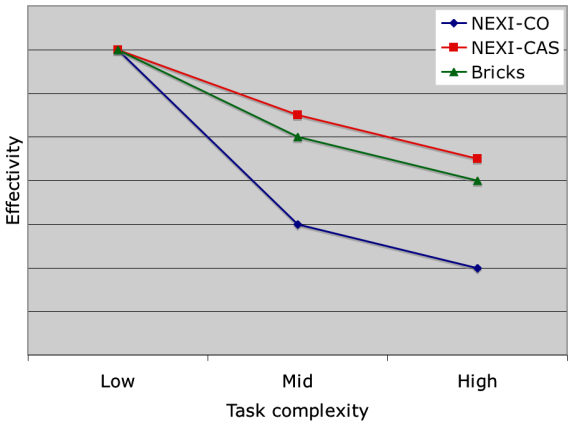
In this section we will briefly discuss the outcome of the usability experiment that was performed to evaluate three different query formulation techniques: keyword-based (NEXI-CO), content and structure-based (NEXI-CAS), and Bricks. First we will discuss the hypotheses that were formulated for the experiment, then present the setup and methodology used for the experiment and finally discuss the results and some observations. A more detailed discussion of the experiment, including the results of a retrieval performance experiment can be found in [2].

4.1 Hypotheses

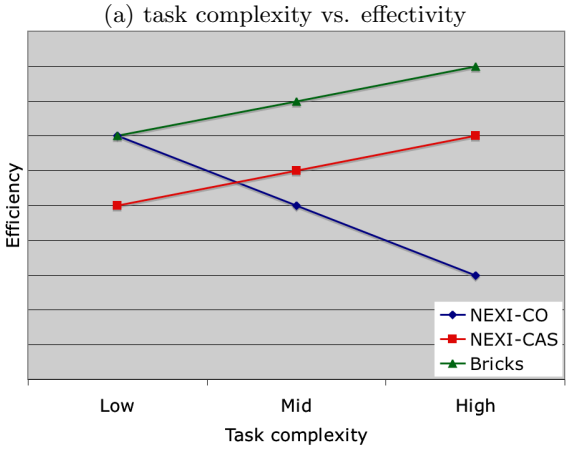
In this article we have formulated three objectives that are important for query formulation in structured document retrieval: (1) minimize the complexity of the query formulation process to the user, (2) minimize the required knowledge of the structure of the document collection, and (3) provide maximum expression power to the user, allowing him to express his (complex) information need. Based on these objectives, we have formulated three hypotheses.

Hypothesis 1
Use of sophisticated query formulation techniques will lead to a higher effectiveness of the task performance.

The intuition behind Hypothesis 1 is that if a user can add structural conditions to the information request, by using



(a) task complexity vs. effectivity



(b) task complexity vs. efficiency

Figure 4: Expected performance, including task complexity

either NEXI-CAS of Bricks, the user is more successful in completing a given search task. Furthermore we designed Bricks to provide similar expression power as is available in NEXI-CAS. Therefore we expect that regardless of the task complexity the effectivity of NEXI-CAS and Bricks will almost be equal, but significantly higher than for NEXI-CO.

When taking the task complexity into account, we expect to find that for tasks with a low complexity the three approaches will have a similar performance, however if the task complexity increase, the effectivity of NEXI-CO will drop. The effectivity of NEXI-CAS and Bricks should remain more or less constant, or slightly decrease. This is depicted in Figure 4.a.

To measure our expectations we have introduced the following effectivity measure:

$$effectivity = \frac{\sum_{fragment} relevant_{fragment} \times \frac{relevant\_retrieved_{fragment}}{rank_{fragment}}}{\mid fragment \mid} \quad (1)$$

$$, \; scale : [0..1]$$

The effectivity measures assigns a score between 0 and 1 to a query that is submitted by a user for a particular information need. It takes into account the number ($relevant\_retrieved_{fragment}$) and position ($rank_{fragment}$) of the relevant fragments ($relevant_{fragment} : [0, 1]$) that are retrieved in the top 10 results. The measure balances the effectivity score, if less then ten results are retrieved for a given user query ($\mid fragment \mid$).

Hypothesis 2
The Bricks approach for query formulation will increase the efficiency of the user for a given task.

When taking the time factor into account, we expect to see a different picture, if our assumptions for Bricks are correct, a user should be able to successfully complete a search task in a shorter period of time, compared to NEXI-CAS. It is hard to predict how this will relate to NEXI-CO, because we expect that the user behavior is more focussing on query refinement and a quick scan of the list with retrieved document fragments. This corresponds with normal search behavior of users on the Internet [20]. Figure 4.b illustrates the expected efficiency for the different systems, when task complexity is taking into account.

We will measure the efficiency of the systems using the following formula:

$$efficiency = \frac{effectivity}{100\ seconds} \quad (2)$$

Hypothesis 3
Bricks will achieve a higher overall satisfaction among users that perform a (complex) search, when compared to the NEXI-based approaches.

We expect that: users, which are offered sophisticated query formulation techniques (Bricks and NEXI-CAS) will be more satisfied, than those users working with a keyword-based interface. In addition we expect that reduction of both the syntactical and structural problems at the user interface will also have a positive influence on the user satisfaction.

We will measure the user satisfaction through a survey, directly after the experiment, using 7-point Lickert scales.

4.2 Setup of the experiment

For this experiment, we used TERS, the Testbed for the Evaluation of Retrieval Systems [20]. TERS provides an experimental environment for the support of various evaluation tasks. It hosts two types of experiments, a usability experiment and a retrieval performance experiment, where the aim of TERS is to investigate the correlation between both experiments. We conducted both experiments, but limit ourselves here to the results of the usability experiment. For the usability experiment we have used the following setup:

Document Collection

For the experiment we used Lonely Planet destinations material, which consists of XML documents with interesting facts and background information about destinations on our planet.

Systems

Three systems were prepared for the experiment: NEXI-CO, NEXI-CAS, and Bricks. To eliminate undesirable side-effects all three systems used the same retrieval engine and result presentation technique.

Users

For the experiment we used a pool of 54 students, that participated in the course 'Multimedia Information Retrieval'. During this course they were taught the basic principles of structured document retrieval, and they followed a lecture on the NEXI query language. Prior to the experiment, they had to complete an assignment where they were asked to create both NEXI-CO and NEXI-CAS queries for fifteen representative information needs, based on the Lonely Planet.

Topics

For the usability we have used 27 topics, i.e. search assignments, representing specific information needs of travelers that are doing a background search, for instance to plan their next holiday. The topics can be sorted in three complexity groups, ranging from low to high complexity. To sort the topics, we have counted the syntactical and structural elements of the ideal NEXI-CAS query that represents the information need expressed in the topic.

Survey

Prior to and directly after the experiment we have presented the participants a list of questions to examine their level of expertise and experiences with the retrieval system.

Experience

In the first 30 minutes of the experiment, participants tried out the TERS interface, and played with the interface of the systems, to become familiar with the setup of the experiment and to reduce the learning effects.

4.3 Results of the usability experiment

In this section, the results of the usability experiment are presented. First we will give a brief overview of the overall results, and then discuss the influence of tasks complexity to the performance.

4.3.1 Overall results

In Table 1 the overall results of the experiment are presented for the three systems based on the measures that were used for the experiment: *time*, *effectivity*, *efficiency*, and *satisfaction*. The effectivity measure, which is used to test Hypothesis 1 shows that a significant difference ($p < .000$) is found between the systems, where both NEXI-CAS and Bricks were more effective than NEXI-CO. This indicates that the use of sophisticated query formulation techniques has a positive influence on the task performance.

Overall performance

System	Time	Effectivity	Efficiency	Satisf.
NEXI-CO	***198***	0.27	0.15	4.1
NEXI-CAS	245	***0.34***	0.14	***4.7***
Bricks	214	0.32	***0.16***	4.6

Table 1: Overall performance for the three systems based on time, effectivity, efficiency and satisfaction.

Effectivity

System	1	2	3
NEXI-CO	0.45	0.35	0.14
NEXI-CAS	***0.48***	***0.48***	***0.21***
Bricks	0.47	0.47	0.18

Time (sec.)

System	1	2	3
NEXI-CO	136	***154***	***246***
NEXI-CAS	160	189	311
Bricks	***134***	160	277

Efficiency (effectivity/100 sec.)

System	1	2	3
NEXI-CO	0.33	0.23	0.06
NEXI-CAS	0.30	0.25	***0.07***
Bricks	***0.35***	***0.30***	***0.07***

Table 2: Experiment results, including task complexity (tabular overview)

When the time factor is taken into account, it becomes apparent that users need significantly ($p < .000$) more time to formulate their information need in NEXI-CAS expressions. As a result, Bricks becomes the most efficient approach ($p < 0.04$) of the three systems, which confirms Hypothesis 2.

Inspection of the outcome of the experiment for user satisfaction, shows that users appreciate the additional query formulation power, but the did not rule in favor of Bricks. The highest satisfaction was achieved with NEXI-CAS, followed at a minimal distance by Bricks. Given that the satisfaction scale goes from 1 to 7, we can conclude that the users were content with both the Bricks and NEXI systems. However, we will have to drop Hypothesis 3.

4.4 Including task complexity

A more detailed insight in the results can be obtained when task complexity is also considered an influencing factor. Table 2 shows the raw results of the experiment for the measures *effectivity*, *time*, and *efficiency*, when task complexity is taken into account.

Effectivity

Figure 5.a shows the influence of task complexity on the effectivity of the performance. When comparing the results with our expectation, as shown in Figure 4.a, we see a sudden drop in effectivity for Bricks and NEXI-CAS, which was not predicted. The overall picture however, supports our expectations.

Time

When comparing the task complexity with respect to the average time needed to complete a task, we see that time is increasing with the task complexity, regardless of the system. This is illustrated in Figure 5.b. However, on average the users need more time to formulate NEXI-CAS queries.

Efficiency

Figure 5.c illustrates the combination of effectivity and time into the efficiency measure. It shows how Bricks is outperforming the other systems for tasks with a low and mid complexity, but that the efficiency for the three systems for highly complex tasks is almost at an equal low point, due to the extra time needed to complete the search assignment. Comparing the results for efficiency with our expectations, as depicted in Figure 4.b, we are mildly positive with the outcome. We had not anticipated the non-linear increase in time needed to complete highly complex tasks.

4.5 Observations

At this point we also want to discuss some of the observations that were made during the experiment. The search behavior of the users working with the different systems was entirely different. Users working with the NEXI-CO interface used many iteration steps to formulate a query and inspect the top of the ranking. If the results were unsatisfactory, they refined their query and tried again.

The participants working with the NEXI-CAS interface show a different strategy: they constructed the NEXI query in several steps. After each step, they submitted the query, to check the syntax and the intermediate results. Then continued extending the query, until the were satisfied with the results. Manual inspection of the submitted queries, showed numerous syntax errors, and misinterpretation of the document structure.

Finally, we observed that the participants working with Bricks hardly used any refinement steps. They continued working until they fully created a representation of the information need in Bricks, and only then inspected the results.

5. CONCLUSIONS

Structure document retrieval gained its popularity due to the use of XML in digital libraries, intranet environments, and large structured web-sites, where users have a specific and often complex information need. For structured document retrieval to work in practice, it is important that users are capable to adequately used the structure of a document in all facets of the retrieval process.

In this article we have identified three aspects that are of influence on the query formulation process for structured document retrieval: (1) adequate expression power, (2) syntactical complexity of the query formulation, and (3) required knowledge of the document structure. Using a keyword-based approach will not provide the user sufficient expression power, as is for instance provided by the NEXI query language. The NEXI query language allows a user to specify both the structural and content-based aspects of the information need, but also burdens the user with syntactical issues during the query formulation process. In addition, the

user has to be familiar with the structure of the document collection to avoid the specification of ill-formed structural paths.

Based on these aspects we have introduced Bricks, the building blocks to tackle query formulation issues in XML retrieval. The objective of Bricks is (1) to reduce the syntactical complexity of the query formulation process, (2) to minimize the required knowledge of the document structure, while (3) maintaining maximum expression power. We have explained how the objectives are used to form the theoretical foundation of Bricks. By using a graphical approach and the intuition of a mental model for query formulation, Bricks allows the user to step-by-step formulate his information need, while avoiding a possible information overload.

Finally we have discussed the outcome of a large scale usability experiment that evaluated the performance of Bricks, with respect to a keyword-based and NEXI-CAS system. Based on the results, we can concluded that sophisticated query formulation techniques, such as offered by Bricks and NEXI-CAS, will increase the success rate of the task performance in terms of effectivity. Furthermore, we can conclude that Bricks is more efficient, since will allow users to successfully complete a given task in a shorter period of time, compared to the keyword-based and NEXI-CAS approaches.

When taking task complexity into account, we found that the effectivity will decrease when the task complexity increases, but that NEXI-CAS and Bricks are more effective for the mid and highly complex search tasks. Increase in task complexity, will also lead to a non-linear increase in time needed to complete the task, causing the efficiency to drop significantly for all systems for the tasks with a high complexity.

Future research

For our future research we will work on alternative query formulation techniques that exploit the tree-based nature of XML documents. Furthermore, we are investigating how user-profiling can be used to enhance keyword-based query formulation for XML retrieval. With respect to result presentation, we will work on sophisticated techniques that use a query driven navigation, allowing the user to inspect the various structural and textual conditions of the information request. Finally we will continue to improve our retrieval engine, by participation in INEX.

Acknowledgments

At this place we would like to thank the Lonely Planet organization. They provided the XML document collection, based on the destinations material, that is used for our experiments. This allowed us to validate our ideas on query formulation issues in structured document retrieval. Furthermore we would like to thank the students that participated in the experiment, and provided us valuable information and new insights.

6. REFERENCES

[1] S. Adler, A. Berglund, J Caruso, S Deach, T. Graham, P. Grosso, E. Gutentag, A. Milowski, S. Parnell, J. Richman, and S. Zilles. Extensible stylesheet language (XSL). W3c recommendation, W3C: World-Wide-Web Consortium, October 2001.

[2] A.J. Baas. Structured document retrieval from a user perspective. Master's thesis, Center for Content and Knowledge Engineering, Institute for Information and Computing Sciences, Utrecht University, Utrecht, the Netherlands, July 2005.

[3] S Boag, D Chamberlin, M.F. Fernandez, D. Florescu, J Robie, and J. Simeon. Xquery 1.0: An XML query language. Working draft, W3C: World-Wide-Web Consortium, April 2005.

[4] J. Clark and S. DeRose. XML Path Language (XPath). Technical report, World-Wide-Web Consortium (W3C), 1999.

[5] D.C. Dryer. Wizards, guides, and beyond: Rational end empirical methods. In *proceedings of the International Conference on Intelligent User Interfaces*, pages 265–286, New York, NY, ASU, 1997. ACM Press.

[6] N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors. *Advances in XML Information Retrieval*, number 3493 in LNCS, Schloss Dagstuhl, Germany, June 2005. INitiative for the Evaluation of XML Retrieval, Springer.

[7] E. Hatcher and O. Gospodnetic. *Lucene in Action*. ISBN: 1932394281. Manning Publications Co., Januari 2005.

[8] G. Kazai, M. Lalmas, and A. de Vries. The overlap problem in content-oriented XML retrieval evaluation. In *proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, number ISBN:1-58113-881-4, pages 72 – 79, Sheffield, UK, 2004. ACM Press.

[9] M. Lalmas and T. Roelleke. Modelling vague content and structure querying in XML retrieval with a probabilistic object-relational framework. In *FQAS, 6th International Conference On Flexible Query Answering Systems*, Lyon, France, June 2004.

[10] L. Meho and H. Tobbo. Modelling the information-seeking behaviour of social scientists; Elly's study revisited. *journal of American Society for Information Science and Technology*, 4(6):570–587, 2003.

[11] J Muramatsu and W. Pratt. Transparent queries: Investigating users' mental models of search engines. In *proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 217–224. ACM Press, 2001.

[12] J. Preece, Y. Rogers, H. Sharp, D Benyon, S. Holland, and T. Carey. *Human Computer Interaction: Concepts and Design*. Number ISBN: 0201627698. Addison Wesley, 1994.

[13] A. Tombros, B. Larsen, and S. Malik. The interactive track at INEX 2004. In N. Fuhr, M Lalmas, S. Malik, and Szlavik Z., editors, *Advances in XML Information Retrieval. Third Workshop of the INitiative for the Evaluation of XML Retrieval*, number ISBN: 3-540-26166-4 in LNCS 3493, pages 410–423. DELOS - Network of Excellence on Digital Libraries, Springer, 2005.

[14] A Trotman and B. Sigurbjörnsson. Narrowed extended xpath i (NEXI). In N. Fuhr, M Lalmas, S. Malik, and Szlavik Z., editors, *Advances in XML Information Retrieval. Third Workshop of the INitiative for the Evaluation of XML Retrieval*, number ISBN: 3-540-26166-4 in LNCS 3493, pages 16–40. DELOS - Network of Excellence on Digital Libraries, Springer, 2005.

[15] A Trotman and B. Sigurbjörnsson. NEXI, now and next. In N. Fuhr, M Lalmas, S. Malik, and Szlavik Z., editors, *Advances in XML Information Retrieval. Third Workshop of the INitiative for the Evaluation of XML Retrieval*, number ISBN: 3-540-26166-4 in LNCS 3493, pages 42–53. DELOS - Network of Excellence on Digital Libraries, Springer, 2005.

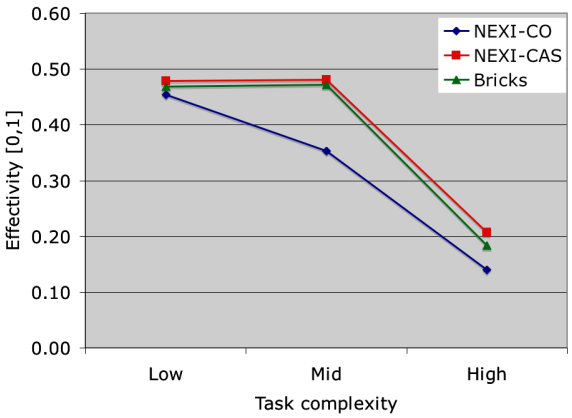
[16] R. van Zwol. *Modelling and Searching Web-based Document Collections*. Ctit ph.d. thesis series 02-40, Centre for Telematics and Information Technology (CTIT), Enschede, the Netherlands, April 2002.

[17] R. van Zwol and P.M.G. Apers. Complex query formulation with the webspace method. In *proceedings of the Sixth World Multi Conference on Systematics, Cybernetics, and Informatics*, number ISBN: 980-07-8150-1, pages 200–208, Orlando, Florida, USA, July 2002. IIIS.

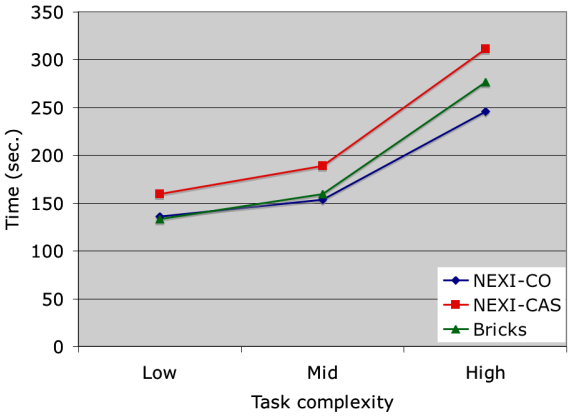
[18] R. van Zwol, A. Callista, J. Molenaar, and F. Wiering. Content authoring in an XML-based and author-friendly environment: Fact or fiction? In *proceedings of the third International Conference on Computer Science and its Applications (ICCSA'05)*, San Diego (CA), USA, June 2005. US Education Services.

[19] van R. Zwol, V. Dignum, and F. Wiering. The Utrecht Blend: Basic ingredients for an XML retrieval system. In N. Fuhr, M Lalmas, S. Malik, and Szlavik Z., editors, *Advances in XML Information Retrieval. Third Workshop of the INitiative for the Evaluation of XML Retrieval*, number ISBN: 3-540-26166-4 in LNCS 3493, pages 140–152. DELOS - Network of Excellence on Digital Libraries, Springer, 2005.

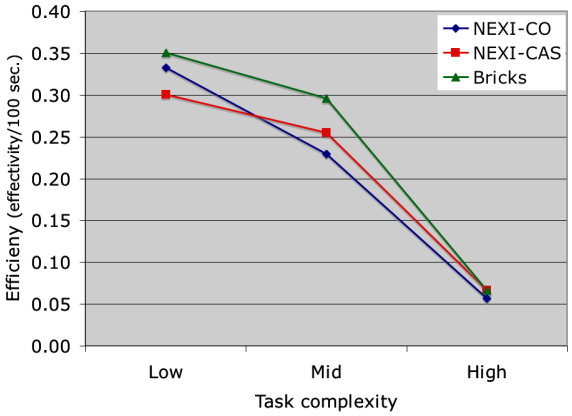
[20] van R. Zwol and H. van Oostendorp. Google's "I'm feeling lucky", truly a gamble? In Xiaofang Zhou, Stanley Su, Mike P. Papazoglou, X Maria Zhou, S. Su, M.P. Papazoglou, M.E. Orlowska, and K.G. Jeffery, editors, *Web Information Systems - WISE 2004, Proceedings of the 5th International Conference on Web Information Systems engineering*, number 3-540-23894-8, pages 378–390, Brisbane, Australia, November 2004. Springer.



(a) task complexity vs. effectivity



(b) task complexity vs. time



(c) task complexity vs. efficiency

Figure 5: Experimental results, including task complexity

Author Index