The overlap problem in content-oriented XML retrieval evaluation

Gabriella Kazai
Queen Mary University of London
London, E1 4NS
UK
gabs@dcs.qmul.ac.uk

Mounia Lalmas
Queen Mary University of London
London, E1 4NS
UK
mounia@dcs.qmul.ac.uk

Arjen P. de Vries
CWI
PO Box 94079, 1090 GB
Amsterdam
The Netherlands
arjen@acm.org

ABSTRACT

Within the INitiative for the Evaluation of XML Retrieval (INEX) a number of metrics to evaluate the effectiveness of content-oriented XML retrieval approaches were developed. Although these metrics provide a solution towards addressing the problem of overlap among returned result elements, they do not consider the problem of overlapping reference components within the recall-base, hence leading to skewed effectiveness scores. We propose alternative metrics that aim to provide a solution to both overlap issues.

Keywords

XML retrieval, INEX, evaluation, metrics, overlap, overpopulated recall-base, cumulated gain

1. INTRODUCTION

The INitiative for the Evaluation of XML Retrieval (INEX) is a large-scale campaign for the evaluation of content-oriented XML retrieval systems. The motivation for INEX came with the widespread use of the eXtensible Markup Language (XML) in Digital Libraries and on the Web, which brought about an explosion in the development of XML tools, including systems to store and access XML content. While some of these systems address the data-centric issues of XML, content-oriented XML information retrieval (IR) systems take a document-centric view on XML and exploit the explicitly represented logical structure of documents, in order to retrieve document components (instead of whole documents) in response to a user query [1, 5]. INEX, which is now in its third year, deals with the evaluation of these content-oriented XML retrieval systems, focusing on the evaluation of their retrieval effectiveness.

During the first two years of INEX, with the collaborative effort of 41 participating groups, a test collection of XML documents with 126 user queries and graded relevance assessments has been created. The use of graded relevance assessments in INEX was deemed necessary due to the logical structure of the XML documents, which requires capturing the difference in relevance among related components. In order to provide a measure of retrieval effectiveness, a number of evaluation metrics have also been developed. These metrics provide mechanisms that consider the issue of multiple overlapping result elements being returned to the user when calculating the effectiveness scores. However, the use of graded assessments has also introduced the problem of overlapping reference components into the evaluation, which is not addressed by the current INEX metrics.

This paper examines the effect of this problem on the performance measures used in INEX. We show that the obtained recall-precision curves provide an over-pessimistic view of retrieval performance, whereby the recall axis is scaled by a factor > 1 (approximately 1.82 according to our estimates). We then propose the use of a new set of measures by extending the cumulated gain based metrics introduced in [7] for the evaluation of XML retrieval approaches. The extension of these metrics include means to solve both result and reference component overlap issues as well as the use of relevance value functions, which separate the model of user behaviour from the actual metric to be employed. The metrics are illustrated using a small number of example systems.

The paper is structured as follows. Section 2 describes the INEX test collection, and exemplifies the overlapping relevance assessments. Section 3 describes the current INEX metrics, and presents anomalies arising from the overlapping relevance assessments. Section 4 discusses the desired characteristics of metrics to overcome the overlap problem. Section 5 presents alternative metrics, together with illustrative results. The paper finishes with conclusions and a plan for future work.

2. THE INEX TEST COLLECTION

The document collection of the INEX test collection [8, 5] consists of 12 107 articles of the IEEE Computer Society's publications, from 1995 to 2002, totalling 494 megabytes. The collection contains over 18.5 million XML nodes including over 8.2 million element nodes of varying granularity, where the average depth of a node is 6.9. The overall structure of a typical article consists of a frontmatter (containing e.g. title, author, publication information and abstract), a body (consisting of e.g. sections, sub-sections, sub-sub-sections, paragraphs, tables, figures, lists, citations) and a backmatter (including bibliography and author information).

The topics of the test collection include typical IR queries where no constraints are formulated with respect to the structure of the retrieval results, and XML queries (in a modified XPath syntax [2]) that contain explicit references to the XML structure. Based on these two topic types, INEX defined three ad-hoc retrieval tasks:

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.
Copyright 2004 ACM 1-58113-881-4/04/0007 ...$5.00.

Table 1: Statistics on assessments for 31 INEX'03 CO topics

(e, s)	#	#/relevant article	#/relevant path
$(3, 3)$	1487	0.88	0.45
$(3, 2)$	723	0.43	0.45
$(3, 1)$	782	0.46	0.37
$(2, 3)$	2176	1.28	0.31
$(2, 2)$	1880	1.11	0.46
$(2, 1)$	1570	0.92	0.46
$(1, 3)$	5001	2.95	0.44
$(1, 2)$	3795	2.24	0.51
$(1, 1)$	8425	4.97	1.22
$(0, 0)$	49576	29.23	0
$(> 0, > 0)$	25839	15.24	4.67

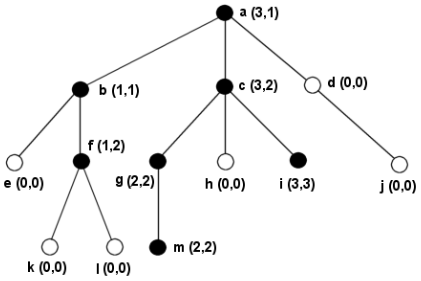


Figure 1: Example XML tree

the content-only (CO) task, which centres around the use of IR queries, where it is left to the retrieval system to identify the most appropriate granularity of information to return to the user; the strict content-and-structure (S-CAS) and the vague content-and-structure (V-CAS) tasks, which are based on the use of XML queries. In this paper we only focus on the CO task where the effect of the overlap problem on the evaluation presents a more dominant issue. In 2003, 36 CO topics were added to the test collection (making a total of 66).

For the construction of the relevance assessments, INEX'03 employed two relevance dimensions: *exhaustivity* and *specificity*. Exhaustivity is defined as a measure of how exhaustively a document component discusses the topic of request, while specificity is defined as a measure of how focused the component is on the topic of request (i.e. discusses no other, irrelevant topics). Both dimensions are measured on four-point scales with degrees of highly (3), fairly (2), marginally (1), and not (0) exhaustive/specific. The motivation for the use of multiple grades was the need 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 combination of the two dimensions is used to identify those relevant document components, which are both exhaustive and specific to the topic of request and hence represent the most appropriate unit of information to return to the user.

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

The assessment pools were created by pooling the top 100 results (out of 1 500) from the 56 submitted CO runs (from 23 groups), resulting in an average pool size of 1 063 elements per CO topic. The assessments were done either by the topic authors or by groups with expertise in the topic's subject area. The assessment procedure made explicit use of the nested XML structure to obtain assessments for each level of granularity, i.e. both ascendant (up to article element) and descendant elements of a relevant component had to be assessed (this was enforced by the assessment system).

Assessments were collected for 31 of the 36 topics, for a total of 15 637 files containing 102 651 assessed elements, of which 11 112 are at article level.[1] More statistics on the collected assessments for each (e, s) value pair are summarised in Table 1. The second column lists the number of elements assessed as (e, s), the third column shows the average number of (e, s) assessments within a relevant article file, and the last column shows the average number of (e, s) assessments on a *relevant path*. A relevant path is 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. Figure 1 illustrates an XML tree with 7 relevant nodes and 3 relevant paths: a-b-f, a-c-g-m and a-c-i. In the INEX assessments there is a total of 14 189 such relevant paths and 25 839 relevant elements in 1 696 relevant article files, where on average an article contains 15.24 relevant elements of which 4.67 are on the same relevant path.

An important property of the exhaustivity dimension is its cumulative nature and propagation effect, whereby the exhaustivity degree of a component is always equal to or greater than its subcomponents' exhaustiveness. Due to this property, all nodes along a relevant path are always relevant (with varying degrees of relevance), hence resulting in a recall-base comprised of a series of overlapping components. It is clear that some relevant nodes along these relevant paths have only been included in the recall-base as a direct result of the propagation effect of exhaustivity, leading to an increase in the size of the recall-base. For example, a single relevant paragraph in an article would generate as many relevant elements as its depth in the article's XML tree. To estimate the increase on the size of the recall-base, we assume that all nodes on a relevant path, except the leaf node, are propagated assessments. This assumption seems a reasonable estimation given that all relevant descendant elements of relevant components had to be assessed. Our estimated rate of increase is then $25\,839/14\,189 = 182\%$, reflecting that $(25\,839 - 14\,189)/25\,839 = 45\%$ of relevant elements in the recall-base are propagated assessments.

3. THE INEX METRICS

Retrieval effectiveness with respect to the CO task has been defined in INEX as a system's ability to retrieve relevant document components that are both exhaustive and specific to the topic of the request. Two metrics were developed to quantify a retrieval system's performance according to this evaluation criterion: inex-2002 (aka. inex_eval) [8] and inex-2003 (aka. inex_eval_ng) [6]. In this section we first provide a brief description of these metrics, then examine the produced recall-precision graphs highlighting anomalies arising from the overlapping reference elements in the recall-base.

3.1 Brief description

Both INEX metrics have been defined with the aim to evaluate systems based on the notions of recall and precision, while also taking into account the relevance degree of retrieved components. In addition, while assuming that users inspect a ranked result list in linear order, both metrics allow the modelling of different user be-

[1] All statistics in this paper are based on the `inex-1.4` collection and the INEX'03 `assessments-2.4` judgements.

haviours. For example, reflecting a more lenient user, near misses (i.e. when components near desired elements are retrieved) may be considered partial successes. Although in both metrics only exact matches between result and reference components are considered a hit, the scoring of near misses is possible based on the collected assessments resulting from the elaborate assessment process.

The inex-2002 metric applies the measure of *precall* [10] to document components and computes the probability $P(rel|retr)$ that a component viewed by the user is relevant:

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

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 components with respect to a given topic.

To apply the above metric, the two relevance dimensions were first mapped to a single relevance scale by employing a quantisation function, $\mathbf{f}_{quant}(e,s)\colon ES \to [0,1]$. The principal idea behind the quantisation is that different functions can be selected according to possible user models. For INEX, two functions were used: $\mathbf{f}_{strict}$ (Equation 2) and $\mathbf{f}_{gen}$ (Equation 3). The former is used to evaluate retrieval methods with respect to their capability of retrieving highly exhaustive and highly specific components. This function models a user for whom only purely relevant components (i.e. those that contain no or only minimal irrelevant information), which are also highly exhaustive are considered worthwhile. The generalised function credits document components according to their *degree* of relevance, hence allowing to model varying levels of user satisfaction gained from not perfect, but still relevant components or near misses.

$$\mathbf{f}_{strict}(e,s) := \begin{cases} 1 & \text{if} \quad e=3 \quad \text{and} \quad s=3, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

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

A side-effect of the generalised quantisation function in Equation 3 is that it shows slight preference towards the exhaustivity dimension, as it assigns higher scores to exhaustive, but not necessarily specific components, where such components are generally large, e.g. article or section elements. As a result, relatively high effectiveness can be achieved with simple article runs, contradicting the goal of the retrieval task. This can be overcome by employing alternative quantisation, such as the one we propose in section 5.1.

A problem with the inex-2002 metric is that it ignores possible overlaps between result elements and rewards the retrieval of a relevant component regardless if it has already been seen by the user either fully or in part. For example, a system Sys_A that returns a relevant section and also one of its relevant paragraphs achieves the same performance as a system Sys_B, which returns two non-overlapping relevant elements. It can be argued that from the user's perspective Sys_B performs better, given that the redundant information of the paragraph returned by Sys_A can be considered as a waste of the user's time and effort.

The inex-2003 metric aims to provide a solution to this problem by incorporating component size and overlap within the definition of recall and precision (Equation 4). (For the derivation of the formulae based on an interpretation of the relevance dimensions within an ideal concept space [11] refer to [6].) 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, 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.

$$\text{recall}_o = \frac{\sum_{i=1}^{k} e\left(c_i\right) \cdot \frac{|c'_i|}{|c_i|}}{\sum_{i=1}^{N} e\left(c_i\right)} \qquad \text{precision}_o = \frac{\sum_{i=1}^{k} s\left(c_i\right) \cdot |c'_i|}{\sum_{i=1}^{k} |c'_i|} \tag{4}$$

Components $c_1, \ldots, c_k$ in Equation 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, which, given a representation, such as a set of (term, position) pairs, can be calculated as:

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

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

Since the inex-2003 metric treats the two relevance dimensions separately, the quantisation functions were also redefined to provide a separate mapping for exhaustivity, $\mathbf{f}'_{quant}(e)\colon E \to [0,1]$ and specificity, $\mathbf{f}'_{quant}(s)\colon S \to [0,1]$, where $E = \{0,1,2,3\}$ and $S = \{0,1,2,3\}$. For the strict case, the result of the quantisation was 1 if $e = 3$ or $s = 3$, respectively, and 0 otherwise. For the generalised case, the quantisation function was defined as $\mathbf{f}'_{gen}(e) = e/3$ and $\mathbf{f}'_{gen}(s) = s/3$.

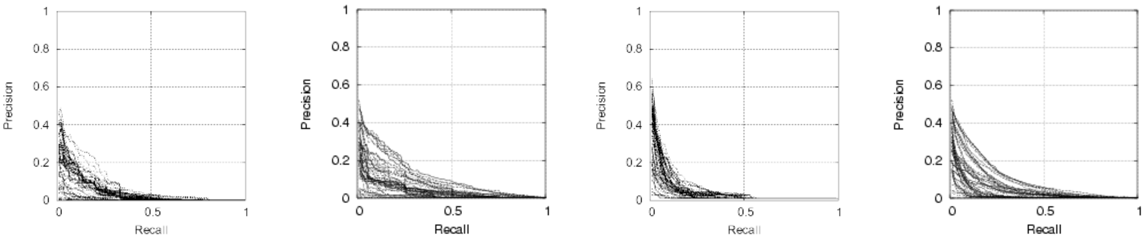
Given our two example systems above, using the inex-2003 metric, Sys_A is now only credited for the returned section element since the user would have already read the contents of the paragraph. As a result, it achieves worse performance than Sys_B. Note that a system Sys_C, which ranks the paragraph first and the section second, would obtain full score for the paragraph and some partial score for the section proportional to the not-yet-seen information contained within it.

3.2 Overpopulated and varying recall base

Both metrics were applied to all 56 CO runs submitted in INEX 2003. Figure 2 summarises the recall/precision graphs for both metrics using their respective quantisation functions.

Although we cannot compare the resulting recall-precision graphs to those obtained for other retrieval tasks, e.g. ad-hoc document retrieval, it is evident that the level of a typical curve in INEX is much lower than what has been presented, e.g. in the TREC conferences. There can be several reasons for this. The task in itself may be considered more challenging given that retrieval is done at the level of document components, where systems not only need to locate relevant information, but also have to decide about the appropriate level of granularity to return to the user. Furthermore, the evaluation in INEX is based on graded scales of two dimensions of relevance, whereas, TREC typically uses a binary relevance scale with a low threshold for relevance. In INEX, when evaluating systems using strict quantisation, in order to perform well, systems are required to retrieve only a very small portion of relevant components (i.e. $(3,3)$-assessments represent 5.75% of all relevant components in the recall-base and 0.018% of all components in the collection).

Although these arguments may go some way towards explaining the 'unusually' low effectiveness results, a further look reveals



2002 2003 2002 2003

strict quantisation generalised quantisation

Figure 2: Summary of recall/precision curves for all 56 INEX'03 CO submissions

deeper problems within the evaluation.

One of the main problems is that of an *overpopulated recall-base*. We use the term 'overpopulated' as it captures the nature of the problem, that the recall-base contains more reference elements than an ideal system should in fact retrieve. The root of this problem lies in the high ratio of overlap among reference components due to the inclusion of multiple nested components in the recall-base (section 2). As a result, perfect recall can only be reached by systems that return all the relevant reference components of the recall-base, including all the overlapping elements. Such retrieval behaviour is, however, contradictory to the definition of an effective XML retrieval system. The aim of returning XML fragments instead of whole documents is to reduce the user effort required in viewing large texts by allowing users to focus only on the parts relevant to their query. However, the inclusion of multiple nested components within the recall-base and the evaluation of systems using metrics that directly rely on the size of this recall-base encourages systems to overwhelm users with redundant information, while also leading to skewed and misleading effectiveness results. This effect can be seen on the recall-precision graphs in Figure 2, where all curves show a sharp decline of precision values at low recall points with almost disappearing effectiveness results after 0.5 recall. In fact, the graphs appear as if precision was plotted against a recall axis of $[0, 0.5+]$. This finding seems to correlate with the high percentage of overlap found in the recall-base due to the propagation effect of the exhaustivity dimension (resulting in an estimated increase of 182% in the size of the recall-base). The precision values of retrieval systems that aim to avoid the retrieval of overlapping components have therefore been plotted against lower recall values than merited according to the task definition, while higher precision values were rewarded for systems that return overlapping elements.

To further demonstrate this effect of the overpopulated recall-base, we constructed an ideal run consisting of only those relevant components that represent the most desirable units to return to the user. These components have been selected from the recall-base, ensuring that all overlap is removed (the exact procedure is described in section 4.1). Figure 3 shows the result of the two INEX metrics (with both strict and generalised quantisation) applied to our ideal run using the original recall-base as ground truth (continuous line). The dotted line shows the expected result that a perfect run should produce, i.e. straight line at the precision value of 1 for the strict case and a slightly sloping line for the generalised case (due to graded relevance). However, as it can be seen, when evaluating our perfect run against the overpopulated recall-base, we get a curve showing far from perfect performance. These graphs clearly demonstrate the distorting effect of the overpopulated recall-base.

A partial solution to this problem has been provided by the strict quantisation function, which allows creating a recall-base with minimised overlap and hence lessening the effect of overlap on the evaluation. However, strict quantisation does not remove overlap completely (on average 30% of $(3, 3)$ assessments overlap), nor does it provide an ideal solution since it only reflects a very limited view based on a rather specific user model (section 3.1). For the generalised case, the overlap of reference components constitutes a crucial issue.

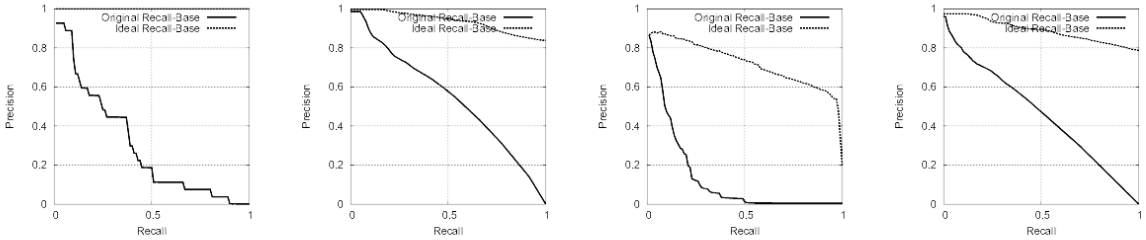
In addition to the problem of overpopulated recall-base, the inex-2003 metric presents another anomaly. As mentioned before, to discourage systems from returning multiple nested components, the inex-2003 metric defined a mechanism to discount the score rewarded for results that have already been seen by the user either fully or in part. The effect of this discounting is of course lower precision values and more steeply descending graphs, which is in fact the intended result. However, via the discounting mechanism, the measure indirectly alters the recall-base that the evaluation is based upon. For example, after a relevant element is retrieved at a given rank, all its relevant sub-components are subsequently rendered irrelevant by the metric. This dynamic removal of some reference components from the recall-base ultimately leads to a situation where no common ground truth exists in the evaluation, but each run is assessed against its own recall-base of varying size.

4. DESIDERATA FOR XML RETRIEVAL

Before we can propose appropriate solutions to address the problem of overlap in the evaluation of XML retrieval systems, both with respect to the overlap of result elements (i.e. varying recall-base) and the overlap of reference elements (i.e. overpopulated recall-base), we first need to consider what the desired results and rankings for XML retrieval should be.

4.1 Ideal retrieval results

We introduce the notion of an ideal retrieval result, which represents the most desirable retrieval unit from a set of overlapping possible results for a given user and user request. Based on the assumed preferences of users of XML retrieval systems, as defined within the CO retrieval task, we can identify an ideal result by selecting it from a set of overlapping relevant reference elements in the recall-base using the following procedure. Given any two components on a relevant path, the component with the higher specificity degree is selected. In case two components' specificity levels are equal, the one with the higher exhaustivity degree is chosen. If both specificity and exhaustivity values are the same, the descendant component is selected. The procedure is applied recursively to all overlapping pairs of components along the relevant path until one element remains. 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



2002 2003 2002 2003

strict quantisation generalised quantisation

Figure 3: Recall/precision curves for constructed ideal run

shortest one. For example, for the XML tree in Figure 1, we obtain the ideal results of f and c.

The validation of the above decisions lies within the definition of the retrieval task. As mentioned before, XML retrieval systems aim to retrieve those relevant components that focus on the user's information need. Such components are expected to contain purely relevant information, or only small amount of irrelevant information. In INEX, these components are said to be highly specific to the topic of the request. Furthermore, from the supposed user's point of view, highly relevant components are preferred to fairly or marginally relevant ones. Finally the descendant is selected from two overlapping reference components with equal exhaustivity and specificity degrees in order to minimise the propagation effect of exhaustivity.

By applying the above procedure to the test collection's recall-base, we effectively define an 'ideal recall-base', where the overlap between reference elements is completely removed. We then consider all relevant components of the original recall-base, other than those included in the ideal recall-base, 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 matches between retrieved 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.

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 *should not* be penalised for not retrieving such near misses.

Once an ideal recall-base has been built, an ideal run can be created by ordering the components of the ideal recall-base in decreasing value of their quantised relevance score.

4.2 How to rank systems?

The possibility to reward systems partial scores for near misses presents several issues regarding the question of how to score systems with overlapping result elements in their ranked result list.

Consider an (imaginary) test collection in which the only two reference elements are the parent component a and its sub-component b, sharing the relevant path of a/b. Say that the recall-base is $\{< /a, (3,1) >, < /a/b, (3,3) >\}$, where b represents the ideal result. Assume now that we have seven systems with the ranked system results shown in Table 2.

Table 2: Example system results; '-' denotes irrelevant node

Sys$_1$	$< b, -, - >$	Sys$_5$	$< a, -, b >$
Sys$_2$	$< b, a, - >$	Sys$_6$	$< a, -, - >$
Sys$_3$	$< b, -, a >$	Sys$_7$	$< -, -, - >$
Sys$_4$	$< a, b, - >$		

Looking at these rankings it is clear that Sys$_1$ performs best on the XML retrieval task of identifying only the most relevant information in the collection, and Sys$_7$ performs worst as it does not retrieve any relevant information. We postulate Sys$_1 \succ$ Sys$_7$, meaning that 'Sys$_1$ performs better than Sys$_7$'. Because Sys$_6$ retrieves a but not b, it follows directly from the task definition that Sys$_1 \succ$ Sys$_6 \succ$ Sys$_7$. Ordering the remaining systems is however less obvious. Is retrieving ascendant a still useful once b has already been shown to the user; or is it even worse than retrieving any other irrelevant element (e.g. more frustrating for the user)? And, if we already retrieved a, is it still useful to point the user at b? Is it sensible to distinguish between the performance of systems Sys$_2$ and Sys$_3$; similarly, should we regard Sys$_4$ and Sys$_5$ of equal performance? A decision on these questions (based on a user model) may lead to a possible desired ranking such as:

$$\mathsf{Sys}_1 \succeq \mathsf{Sys}_2 \succeq \mathsf{Sys}_3 \succ \mathsf{Sys}_4 \succeq \mathsf{Sys}_5 \succeq \mathsf{Sys}_6 \succ \mathsf{Sys}_7$$

However, what should happen if the assessments include another relevant element c on the path $/a/c$, such that the recall-base is $\{< /a, (3,1) >, < /a/b, (3,3) >, < /a/c, (3,3) >\}$? This new recall-base affects the desired ranking of the system results, because systems that return both a and b could now be rewarded somehow for implicitly pointing the user to c, that none of the systems retrieved. In the given system results, Sys$_2$ may hence perform (arguably) better than Sys$_1$, because Sys$_1$ does not give any information about the existence of c in the collection. On the other hand, ordering systems Sys$_2$ and Sys$_4$ cannot be resolved without taking into account the structure of the sub-tree of node a. If a has a large number of child nodes, then it may be preferable if b is returned first. If, however, a has just the two children, b and c, then we could argue that Sys$_4$ is preferable over Sys$_2$.

In order to answer any of the above questions, appropriate models of user behaviours would be required. Since the evaluation of XML retrieval systems is still in its infancy, no such elaborate user models exist. In INEX, the definition of the retrieval task itself provides clues about the assumed user preferences regarding retrieved components, but the definition of a desired ranking is left largely unsolved, mainly due to the expectation that systems should not return overlapping result elements. However, the shortage of proven user models only highlights the need for employing evaluation measures, which are flexible enough and can be appropriately tuned to reflect different user behaviours. The use of the quantisation functions in the current INEX metrics go some way towards this goal, but are burdened with problems caused by how overlap is handled within the traditional recall-precision framework.

5. PROPOSED SOLUTIONS

This section proposes to resolve the problems that graded relevance and overlap introduce into the evaluation by adopting the cumulated gain based measures of [7]. We first introduce the notion of a relevance value function to separate a model of user behaviour from the actual metric to be employed. We then explain how to apply cumulated gain based evaluation measures to the evaluation of content-oriented XML retrieval.

5.1 Relevance value functions

As described in section 3.1, the current INEX metrics employ quantisation functions in order to map the two relevance dimensions to a single relevance scale. Any such mapping serves as a model of user behaviour and is used as a parameter of the evaluation. Extending this notion, we define a *relevance value (RV) function*, $r(c_i)$, as a function that returns a value in $[0,1]$ for a component c_i in a ranked result list representing the component's relevance to the user. The meaning of such a relevance value within the evaluation may be compared to the notion of utility, where the result of the RV function reflects the worth that a retrieved component represents to the user (0 reflects no relevance or utility, 1 is highest relevance score and values in between represent various relevance levels). The calculation of the relevance value can be based on parameters such as the component's assigned exhaustivity and specificity (e,s) values, the ratio of already viewed parts, or even possibly viewed parts (e.g. assuming that a retrieved component is presented as an entry point into an XML document, it is likely that a user may discover additional relevant information just by reading on [4]). Given the richness of the parameters that contribute to a model of a user, a wide variety of RV functions may be defined, each capturing a different type of user behaviour. Here we concentrate on two classes, result-list independent and result-list dependent RV functions. They share the assumption that users view the ranked result list in linear order, from top rank down.

Result-list independent RV functions

A result-list independent RV function returns a relevance value based only on the (e,s) pair assigned to a component c_i:

$$r(c_i) = f\,(assess(c_i)) \tag{6}$$

where f is a mapping function such that $f(\cdot)\colon ES \rightarrow [0,1]$, and $assess(c_i)$ is a function which returns the (e,s) assessment value pair for the component c_i, provided that one exists in the recall-base and otherwise it returns $(0,0)$.

A binary function of f assigns only the values of 0 (irrelevant) or 1 (relevant) to a given (e,s) pair. The strict quantisation function in Equation 2 is an example of such a binary mapping.

Non-binary mappings order the (e,s) pairs according to their sought value for the user as returned retrieval units (given an evaluation criterion). E.g., INEX assumes that highly exhaustive and highly specific components are of most value to the user, such that $assess(c_i) = (3,3) \wedge assess(c_j) \neq (3,3) \Rightarrow r(c_i) > r(c_j)$. We denote the ordering of assessment pairs as O. The number of ranks in O is $1 \leq |O| \leq |ES|$, where each rank represents a level of relevance (worth) to the user. An example is the generalised quantisation function in Equation 3, implementing a weak ordering of assessment pairs. Here, the actual relevance values corresponding to the ranks in O are assigned by a linear scoring function, resulting in an ordinal relevance scale with a fixed increment between the relevance values (i.e. $1/|O|$). Alternatively, we can choose a non-linear scoring function resulting in a ratio scale, to express e.g. the evaluation criterion that retrieving a highly exhaustive and specific element is 10 times more valuable to the user than a fairly relevant and marginally specific component. This is implemented in an RV function where $r(c_i) = 1$ if $assess(c_i) = (3,3)$ and $r(c_i) = 0.1$ if $assess(c_i) = (2,1)$.

In section 4.1 we defined a relative ranking of assessment value pairs that closely reflects the evaluation criterion for XML retrieval defined in INEX. According to this, specificity plays a more dominant role than exhaustivity (contrary to Equation 3). Based on this criterion, we propose a 'specificity-oriented generalised' mapping:

$$\mathbf{f}_{sog}(e,s) := \begin{cases} 1 & \text{if} \quad (e,s) = (3,3), \\ 0.9 & \text{if} \quad (e,s) = (2,3), \\ 0.75 & \text{if} \quad (e,s) \in \{(1,3),(3,2)\}, \\ 0.5 & \text{if} \quad (e,s) = (2,2), \\ 0.25 & \text{if} \quad (e,s) \in \{(1,2),(3,1)\}, \\ 0.1 & \text{if} \quad (e,s) \in \{(2,1),(1,1)\}, \\ 0 & \text{if} \quad (e,s) = (0,0). \end{cases} \tag{7}$$

Result-list dependent RV functions

The second class of relevance functions takes into account the full ranked result list. Our goal is to discourage systems from retrieving overlapping result elements by rewarding them only once for each portion of retrieved information. This reflects the viewpoint of a user for whom any already viewed components become irrelevant, and the relevance of components seen in part is reduced.

The inex-2003 metric penalises systems for overlapping results by normalising the quantised score of a component with the ratio of the size of the 'not-yet-seen' part of the component and the total size of the component (see Equation 4). To achieve the same goal, we define the following RV function:

$$r(c_i) = \frac{f(assess\,(c_i)) \cdot \,|c_i'|}{|c_i|} \tag{8}$$

where $|c_i|$ is the total size of component c_i, and $|c_i'|$ is the size of the not-yet-seen part of c_i (see Equation 5).

An advantage of implementing this scoring strategy within our RV function instead of directly within the effectiveness measure is that it provides a clearer separation of the user-specific parameters from the actual metric to be used. Another benefit of this approach over the inex-2003 metric is that both dimensions of relevance are incorporated within the resulting relevance score. While it is acceptable to consider the two dimensions separately in the ideal concept space interpretation, the inex-2003 metric has been criticised for separating exhaustivity and specificity since their combination is required to identify the most desirable retrieval components.

As mentioned already, the above formulation assumes that relevant information is uniformly distributed within the component, hence providing only a rough estimate of the actual relevance value that a component-part may represent to a user. For example, given a relevant section s_1, assessed as $(3,1)$, with one relevant paragraph, p_1 $(3,3)$, and nine irrelevant paragraphs, $p_2 \ldots p_{10}$, it would seem less reasonable to reward partial score for the retrieval of s_1 after p_1 has already been seen by the user (Equation 4 would credit a score of 1 for p_1 and then $0.3 \cdot 90\% = 0.27$ for s_1). To provide an alternative estimate of the relevance value of a component-part, we define a new assessment function $assess'(\cdot)$, which uses the collected assessments for its estimation. Like $assess(\cdot)$, it returns for a not-yet-seen component c_i, the (e,s) assessment value pair given within the recall-base and $(0,0)$ otherwise, i.e. $assess'(c_i) = assess(c_i)$. For a component, which has been fully seen by the user previously, $assess'(\cdot)$ returns $(0,0)$. However, if the input parameter is a component-part c'_i (i.e. one which has in part already been seen by the user), then the function calculates an estimate for its assessment value pair:[2]

$$assess'(c'_i) = \alpha \cdot \frac{\sum_{j=1}^{m} (assess'(c^*_j) \cdot |c^*_j|)}{\sum_{j=1}^{m} |c^*_j|} + (1-\alpha) \cdot assess(c_i) \tag{9}$$

where $|\cdot|$ denotes component size, m is the number of child nodes of c_i, and c^*_j may be a not yet seen child node, a component-part or a component already seen in full by the user. The weighting factor of α represents how much frustration we assume a user may suffer from accessing redundant component-parts. For example, setting $\alpha = 1$ reflects a user who does not tolerate already viewed components. Setting $\alpha = 0$ gives our result-list independent RV function (Equation 6).

Since the computation in Equation 9 may return any value $(e,s) \in \mathrm{IR} \times \mathrm{IR}$, the mapping function f, e.g. Equation 7, has to be adapted to cater for such input and return 1 if $(e,s) \in \{(x,y)|2 < x \leq 3 \wedge 2 < y \leq 3\}$, etc. Denoting this adaptation of f to regions as f', we define the following result-list dependent RV function:

$$r(c_i) = f'(assess'(c_i)) \tag{10}$$

Based on Equation 10 and using $f' = f'_{sog}$ and $\alpha = 0.9$, we obtain the following relevance values for our example retrieval run containing p_1 and s_1: $r(c_{p_1}) = 1$ and $r(c_{s_1}) = 0.9 \cdot 0 + (1-0.9) \cdot 0.25 = 0.025$. With $\alpha = 1$ we get $r(c_{p_1}) = 1$ and $r(c_{s_1}) = 0$.

5.2 Cumulated Gain

Järvelin and Kekäläinen proposed a set of cumulated gain based metrics [7] in order to credit IR systems according to the retrieved documents' degree of relevance. These metrics provide an alternative evaluation approach to those that extend traditional evaluation methods, i.e. generalised recall and precision [9]. Their aim was to combine a measure of document rank and degree of relevance in a coherent way resulting in metrics that are not heavily influenced by outliers (relevant documents found late in the ranking).

They proposed three novel metrics that compute the cumulated gain the user obtains by examining the retrieval results up to a given rank. The first metric, cumulated gain (CG), accumulates the relevance scores of retrieved documents along the ranked list. Given a ranked document 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:

$$\mathbf{CG[i]} := \sum_{j=1}^{i} G[j] \tag{11}$$

For example, based on a four-point relevance scale with relevance degrees of $\{0,1,2,3\}$, the ranking $G = <3,2,3,0,1,2>$ produces the cumulated gain vector of $CG = <3,5,8,8,9,11>$.

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). The area between the two curves then shows the user effort wasted due to the imperfect retrieval order.

Their second metric, discounted cumulated gain (DCG), extends CG with a discount factor on the relevance scores in order to devalue late-retrieved documents based on the idea that the further down the rank a document, the less likely that it will be examined by the user. In our approach, this discounting can be implemented by defining a result-list dependent RV function. This also provides additional flexibility in experimenting with different user models, without having to define a new metric and evaluation tools.

Their final metric is the normalised (D)CG measure (n(D)CG), where the (D)CG vectors of the retrieval runs are divided by their corresponding ideal (D)CG vectors. This way, 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.

5.3 Cumulated Gain for XML

In order to adapt these metrics for INEX, we need to consider a number of issues due to the fact that the metrics were originally developed for a single dimension of relevance and for the evaluation of document, not XML, retrieval systems. The first issue requires a mapping of the assessment pairs to a single relevance scale. This is provided by both classes of RV functions (section 5.1). The latter problem requires the consideration of both overlapping result and reference components. Regarding the overlap of result elements, the result-list dependent RV function provides a solution. Our proposal, hence, is to employ the RV function of Equation 10 to derive the relevance score of a component in a result ranking, G, achieving a solution to both the issues of multiple relevance scales and overlapping result components. Regarding the overlap of reference elements, the construction of an ideal recall-base, such as the one proposed in section 4.1, presents a solution.

There are, however, two important consequences of the use of the ideal recall-base for the construction of the ideal CG vectors. First, it may contain less elements than the number of ranks in an actual retrieval run (i.e. due to systems retrieving multiple overlapping components). Second, it has a maximum gain value, which may be exceeded by a retrieval run when the relevance score of a retrieved element in G is calculated based on the full recall-base (i.e. when counting near misses as partial successes). To address the first issue, we extend the ideal gain vectors with additional irrelevant components. This does not alter the evaluation since with no additional relevant elements, no additional gain is cumulated. A solution to the second problem is to maximise the gain value that a retrieval run can accumulate as the maximum of the ideal vector's gain. This way, an actual CG curve is forced to level after meeting the ideal CG curve.

Based on our extended cumulated gain based functions, the evaluation of the seven example systems introduced in section 4.2, based on the recall-base of $\{< /a,(3,1) >, < /a/b,(3,3) >\}$, can be done as follows. We employ the RV function of Equation 10 (with f'_{sog} and $\alpha = 1$) in order to produce the relevance score of a retrieved component taking into account both its assessment value and rank position. We obtain $G_{Sys_1} = G_{Sys_2} = G_{Sys_3} = <1,0,0>$, $G_{Sys_4} = G_{Sys_5} = G_{Sys_6} = <0.25,0,0>$ and $G_{Sys_7} = <0,0,0>$. Applying Equation 11, we calculate the cumulated gain for each rank position: $CG_{Sys_1} = CG_{Sys_2} = CG_{Sys_3} = <1,1,1>$, $CG_{Sys_4} = CG_{Sys_5} = CG_{Sys_6} = <0.25,0.25,0.25>$ and $CG_{Sys_7} = <0,0,0>$, and finally the nCG vectors, which in this case, given that $I = <1,0,0>$ and $CG_I = <1,1,1>$, match the CG vectors. As a result we obtain

[2] The assessment value pairs (e,s) are treated as 2-D vectors in a Euclidean vector space.

the ranking of:

$$\mathsf{Sys}_1 = \mathsf{Sys}_2 = \mathsf{Sys}_3 \succ \mathsf{Sys}_4 = \mathsf{Sys}_5 = \mathsf{Sys}_6 \succ \mathsf{Sys}_7$$

This ranking reflects user preference towards more specific components and considers already viewed components as irrelevant.

When considering the extended recall-base, which includes the assessment $< /a/c, (3, 3) >$, where a has 5 child nodes, all of equal length, and $\alpha = 1$, we obtain $G_{Sys_1} = < 1, 0, 0 >$, $G_{Sys_2} = < 1, 0.2, 0 >$, $G_{Sys_3} = < 1, 0, 0.2 >$, G_{Sys_4}, G_{Sys_5} and G_{Sys_6} remains $< 0.25, 0, 0 >$, and $G_{Sys_7} = < 0, 0, 0 >$. The system ranking in this case (based on nCG where $CG_I = < 1, 2, 2 >$) is:

$$\mathsf{Sys}_2 \succ \mathsf{Sys}_3 \succ \mathsf{Sys}_1 \succ \mathsf{Sys}_4 = \mathsf{Sys}_5 = \mathsf{Sys}_6 \succ \mathsf{Sys}_7$$

This corresponds to a user who does not tolerate redundancy in the retrieved results, but appreciates the retrieval of component a because it does lead to finding c in the end.

Of course, numerous possible scenarios can be evaluated using the combination of different RV functions with different parameter settings. The idea here is only to show that given a model of user behaviour, appropriate settings can be chosen to provide an evaluation platform according to which the different approaches to XML retrieval can be compared.

The advantages of employing the adapted cumulated gain based metrics is that they provide a solution to the problems that graded assessments and overlap introduce into the evaluation when traditional recall-precision based measures are employed (section 3.2). The problem of overpopulated recall-base is solved with the use of an ideal recall-base, upon which the ideal CG vectors are calculated. Unlike with recall-precision metrics, this does not restrict the evaluation to only perfect matches, but it remains flexible to allow partial successes to be taken into account (by matching results against the full recall-base). The only limitation is that it imposes a maximum 'document cut-off value' or rank (i.e. where the actual and ideal CG curves meet), after which the interpretation of the evaluation results requires further investigation. The problem of varying recall-base is addressed by defining CG_I on the ideal recall-base (i.e. independent of a retrieval run) and hence removing the direct dependence of the evaluation on the size of the recall-base.

6. CONCLUSIONS

This paper reviewed the approaches employed within the INEX evaluation initiative for the evaluation of content-oriented XML retrieval systems. We pointed out some non-trivial problems with the current evaluation, and provided an analysis of the effect of these issues. Our results explain how the skewed effectiveness results are a side-effect of the high overlap of reference components within the recall-base.

Our main contribution is a general framework for the definition of evaluation metrics suitable for XML retrieval, integrating the advantages of existing proposals within the INEX metrics with the cumulated gain based approach to the evaluation of IR systems. The use of measures based on cumulated gain overcomes the problem of overlapping reference elements through the definition of an ideal recall-base that distinguishes ideal components that *should* be retrieved by XML systems from near misses that *could* be rewarded as partial successes. The overlap of result elements has an effect on user satisfaction, so appropriate metrics should be based on a model of users of XML retrieval systems. We did not, however, intend to define such a user model here. Rather, our approach was to develop evaluation measures that facilitate the instantiation of different possible user models via the definition of different relevance value functions.

As part of our future research, we aim to apply our metrics to the actual retrieval runs submitted to INEX. For this we will need to obtain realistic models of user behaviour in XML retrieval, which is the aim of the interactive track at INEX 2004 currently being set up. Our framework provides the flexibility to experiment with various (existing or newly defined) relevance value functions, to capture the user behaviour observed. Our proposed extended cumulated gain based metric will then rank systems corresponding to the actual user preferences identified.

In the larger context of IR, the problem of overlap between result items also exists when systems are requested to identify relevant information in a document collection without predefined retrieval units, e.g. passage retrieval and video retrieval [4]. The evaluation framework proposed here is therefore also applicable in the evaluation of retrieval tasks where the elements to be retrieved have not been defined a priori.

7. REFERENCES

[1] H. M. Blanken, T. Grabs, H.-J. Schek, R. Schenkel, and G. Weikum, editors. *Intelligent Search on XML Data, Applications, Languages, Models, Implementations, and Benchmarks*, volume 2818 of *LNCS*. Springer, 2003.

[2] J. Clark and S. DeRose. XML Path Language (XPath) version 1.0. W3C Recommendation. http://www.w3.org/TR/xpath. Technical Report REC-xpath-19991116, WWW Consortium, Nov. 1999.

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

[5] N. Fuhr, M. Lalmas, and S. Malik, editors. *Proceedings of the Second Workshop of the INitiative for the Evaluation of XML Retrieval (INEX). Dagstuhl, Germany, December 15–17, 2003*, 2004. http://inex.is.informatik.uni-duisburg.de:2003/proceedings.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. Cumulated Gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (ACM TOIS)*, 20(4):422–446, 2002.

[8] G. Kazai, M. Lalmas, N. Fuhr, and N. Gövert. A report on the first year of the INitiative for the Evaluation of XML retrieval (INEX'02), European research letter. *Journal of the American Society for Information Science and Technology (JASIST)*, 55(6):551–556, Apr. 2004.

[9] 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.

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

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