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.

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

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 [9, 2]: 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 these personal decision 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 or 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.

2.1 Relevance dimensions

In INEX relevance assessments, two relevance dimensions are used for evaluating XML elements: exhaustivity and specificity. 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 ($P(rel|retr)$):

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

where $esl_{x \cdot n}$ denote 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:

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

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 $f_{generalized}$ (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.

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

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

As can be seen in the definition of generalized quantisation function, this function favors exhaustivity over specificity. The question is does this follows 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?

[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.

The features of 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 its part or if it has been seen entirely. 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 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: $f'_{generalized}(e) = e/3$ and $f'_{generalized}(s) = s/3$.

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 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:

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

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:

$$CG[i] = \sum_{j=1}^{i} G[j] \tag{13}$$

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:

$$rv(c_i) = f(quant(assess(c_i))) \tag{14}$$

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:

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

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).

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

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

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

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.

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.

	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 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.

	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`

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`

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.

	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`

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

[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.

	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	0.217	0.621	23	0.152	0.851	7

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

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.

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.

	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

The run in Table 7 (utwente2, LMM-VCAS-Relax-0.35) shows different behaviour. First, the overlap in this run never exceeds 30%. 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.

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.

	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

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

	cut-off at 10			cut-off at 100		
run id	precision	overlap	rank	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

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`).

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, 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 also 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. In *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.