

Reliability Tests for the XCG and *inex-2002* Metrics

Gabriella Kazai¹, Mounia Lalmas¹, and Arjen de Vries²

¹ Dept. of Computer Science, Queen Mary University of London, London, UK

{gabs, mounia}@dcs.qmul.ac.uk

² CWI, Amsterdam, The Netherlands

arjen@acm.org

Abstract. In this paper we compare the effectiveness scores and system rankings obtained with the *inex-2002* metric, the official measure of INEX 2004, and the XCG metrics proposed in [4] and further developed here. For the comparisons, we use simulated runs as we can easily derive the desired system rankings that a reliable measure should produce based on a predefined set of user preferences. The results indicate that the XCG metrics are better suited for comparing systems for the INEX content-only (CO) task, where systems aim to return the highest scoring elements according to the user preferences reflected in a quantisation function, while also aiming to avoid returning overlapping components.

1 Introduction

The official metric of INEX 2004 is *inex_eval* or, as referred here, the *inex-2002* metric. This metric has been chosen by INEX as the official measure partly because at the time it was still not clear how much its known weaknesses would effect the overall system rankings and partly because alternative measures were not yet ready to take this role. Some of the known weaknesses were reported in [4, 5]. One such issue is that the metric does not take into account the overlap between result elements and hence produces better effectiveness scores for systems that return multiple nested components, e.g. a paragraph and its container section and article. At the INEX 2003 workshop, it was agreed that such a system behaviour should not be rewarded, but in fact should be penalised [4]. Furthermore, comments collected as part of the user studies run by the interactive track at INEX 2004 confirm that “searchers generally recognised overlapping components, and found them an undesirable ‘feature’ of the system” [9]. Another issue with the *inex-2002* metric is that it calculates recall based on the full recall-base, which also contains large amounts of overlapping components. This means that 100% recall can only be reached by systems that return all elements of the full recall-base including all overlapping components. An affect of the latter issue is that the precision scores of systems, that aim to avoid inundating users with overlapping, and hence redundant, elements, are plotted against lower recall values than merited [5].

An argument for the *inex-2002* metric is that it can produce reliable rankings of systems provided none of the systems retrieve overlapping result elements. Although the effectiveness scores would still reflect a pessimistic estimate of performance (due to the overlap amongst the reference elements in the full recall-base), the relative ranking of systems may provide a suitable reflection of performance.

However, most of the current systems at INEX output result lists, where high overlap ratios in the region of 70-80% are not uncommon. This then raises the question whether we can trust the scores obtained by the inex-2002 metric.

In this paper, we investigate this question by means of a basic reliability test. We refer to the reliability test of this study as basic, as we do not provide here a comprehensive survey of acceptable error rates, levels of significant differences in effectiveness scores and so on, but we concentrate only on evaluating “the metrics’ ability to rank a better system ahead of a worse system” [10]. We test the inex-2002 metric [2] and the XCG metrics proposed in [5] and further developed in this paper. For the comparisons, we use simulated runs instead of the actual INEX runs submitted by participants. The reason for this is that by controlling which elements and in what order should make up a run, we can get clearer conclusions regarding the behaviours of the metrics under evaluation.

In the following, we first give a quick overview of the two metrics (Section 2) and then describe the setup and results of our metric reliability test (Section 3). As an indication of the effect of overlap on the rankings of the official INEX 2004 runs, in Section 3.5 we give the ten highest scoring runs obtained for the two metrics. We close with conclusions in Section 4.

2 The Metrics

This section gives a brief summary of the inex-2002 (aka. *inex_eval*) [2] and the XCG metrics introduced in [5] and extended here.

2.1 The inex-2002 Metric

The inex-2002 metric applies the measure of *precall* [8] to document components and computes the probability $P(\text{rel}|\text{retr})(x)$ that a component viewed by the user is relevant:

$$P(\text{rel}|\text{retr})(x) := \frac{x \cdot n}{x \cdot n + \text{esl}_{x \cdot n}} \quad (1)$$

where $\text{esl}_{x \cdot n}$ denotes the *expected search length* [1], 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 are first mapped to a single relevance scale by employing a quantisation function, $\mathbf{f}_{\text{quant}}(e, s) : ES \rightarrow [0, 1]$, where ES denotes the set of possible assessment pairs (e, s) :

$$ES = \{(0, 0), (1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}$$

There are a number of quantisation functions currently in use in INEX, e.g. strict or generalised (see Equations 2 and 3 in [4]), each representing a different set of user preferences. In this paper we concentrate on the “specificity-oriented generalised” (*sog*) quantisation function proposed in [5]:

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

The argument in [5] is that the relative ranking of assessment value pairs in the above formula better reflects the evaluation criterion for XML retrieval as defined within the CO task. According to this, specificity plays a more dominant role than exhaustivity. This is not the case for the generalised quantisation function, which shows slight preference towards exhaustivity, assigning high scores to exhaustive, but not necessarily specific components. Due to the propagation effect and the cumulative property of exhaustivity, such components are generally large, e.g. `bdy` or `article`, elements [6]. This means that relatively high effectiveness scores could be achieved with simple article runs, which contradicts the goal of the retrieval task. The *sog* mapping aims to overcome this bias.

Like all quantisation functions, the *sog* quantisation captures a relative ranking of exhaustivity-specificity value pairs reflecting user preferences, such that, e.g., $(e, s) = (3, 3)$ nodes are preferred to $(e, s) = (2, 3)$ nodes, which in turn are better than $(e, s) \in \{(1, 3), (3, 2)\}$ nodes and so on.

2.2 Cumulated Gain Based Metrics

The XCG metrics described in the next Section are extensions of the cumulated gain (CG) based metrics proposed by Järvelin and Kekäläinen in [3]. The motivation for the CG metrics was to develop a measure for multi-grade relevance values, i.e. to credit IR systems according to the retrieved documents' degree of relevance. The CG 'metrics-family' includes four measures: the Cumulated Gain (CG) measure, the Discounted Cumulated Gain (DCG) measure and their normalised versions, the normalised Cumulated Gain (nCG) and the normalised Discounted Cumulated Gain (nCG) measures. Here we only cover the CG and the nCG metrics as the discounting method can be implemented directly within our extensions rather than defining a separate measure.

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

$$\mathbf{CG}[i] := \sum_{j=1}^i G[j] \quad (3)$$

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

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 be compared to the ideal ranking by plotting the gain value of both the actual and ideal CG functions against the rank position. We obtain two monotonically increasing curves, levelling after no more relevant documents can be found.

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

2.3 The XCG Metrics: Cumulated Gain Based Metrics for XML Retrieval

The XCG metrics include extensions of both the CG and nCG measures: XCG and nXCG, respectively. The motivation for the XCG metrics was to extend the CG measures in such a way that the problem of overlapping result and reference elements can be addressed within the evaluation framework. The extension of the CG metrics to XML documents, and in particular to INEX, lies partly in the way the relevance score for a given document - or in this case document component - is calculated via the definition of so-called relevance value (RV) functions, and partly in the definition of ideal recall-bases, while the actual formula of calculating cumulated gain (Equation 3) is unchanged.

An ideal recall-base is a set of ideal result nodes selected from the full recall-base based on a given quantisation function and the following methodology. Given any two components on a relevant path¹, the component with the higher quantised score (as per chosen quantisation function) is selected. In case two components' scores are equal, the one deeper in the tree is chosen². 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 the selected ideal components, keeping from two overlapping ideal paths the shortest one.

Consider Figure 1 as an example. The figure shows the elements and their structural relationships within an article file (*co/2001/r7022.xml*) of the INEX test collection, where only the nodes that have been assessed as relevant (for topic 163) are included. For each node, the node name, the assigned assessment value pair (in the form of (e,s)), the size of the element in number of characters contained, and the size ratio of the node to its parent node is shown. According to the algorithm above, in the first step one ideal node is selected from each relevant path. For example, from the relevant path $\{\text{article1}, \text{bdy1}, \text{sec4}, \text{sec4/p2}\}$, *sec4* is selected as ideal as it obtains the highest score of 0.5 based on the *sog* quantisation function, with all other nodes on the path each scoring 0.25. From the relevant path $\{\text{article1}, \text{bdy1}, \text{sec4}, \text{sec4/p1}\}$, *sec4/p1* is selected as the ideal node with a score of 0.9. Once all relevant paths have

¹ A relevant path is defined as a path in the XML tree of a document, such as an article file in the INEX collection, whose root node is the *article* element and whose leaf node is a relevant component (i.e. $(e > 0, s > 0)$) that has no or only irrelevant descendants. E.g. in Figure 1 there are 6 relevant paths.

² We are also experimenting with the alternative option, i.e. selecting the node higher in the tree.

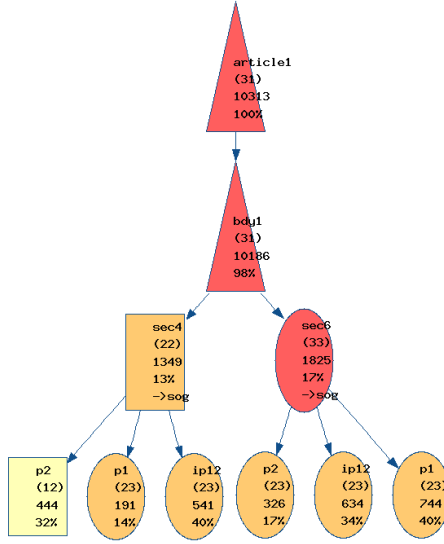


Fig. 1. Sample assessments showing only relevant nodes (i.e. $e > 0$ and $s > 0$) for topic 163 in the article file `co/2001/r7022.xml`. For each node, the node name, the assessment value pair (es), the size in characters and the size ratio to its parent node is shown

been processed, in the final step any remaining overlap is removed. For example, from the overlapping ideal nodes `sec4` and its descendant `sec4/p1` only the former is kept.

The resulting ideal recall-base contains the best elements to return to a user based on the assumptions that overlap between result nodes should be avoided and that the user’s preferences are reflected within the employed quantisation function. The derived ideal recall-bases then form the basis for the ideal gain vectors for each topic.

While I is derived from the ideal recall-base, the gain vectors, G , for the runs under evaluation are based on the full recall-base in order to enable the scoring of near-miss components³.

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

$$rv(c_i) = f(quant(assess(c_i))) \tag{4}$$

where $assess(c_i)$ is a function that returns the assessment value pair for the component c_i if given within the recall-base and $(e, s) = (0, 0)$ otherwise. The $rv(c_i)$ function then returns, for a not-yet-seen component c_i the quantised assessment value pair $quant(assess(c_i))$, where $quant$ is a chosen quantisation functions, e.g. *sog*. In this case $f(x) = x$. For a component, which has been previously fully seen by the user, we have $rv(c_i) = (1 - \alpha) \cdot quant(assess(c_i))$, i.e. $f(x) = (1 - \alpha) \cdot x$. With α set to

³ All relevant components of the full recall-base that are not included in the ideal recall-base are considered as near-misses.

1, the RV function returns 0 for a fully seen, hence redundant, component, reflecting that it represents no value to the user any more. Finally, if c_i has been seen only in part before (i.e. some descendant nodes have already been retrieved earlier in the ranking), then $rv(c_i)$ is calculated as:

$$rv(c_i) = \alpha \cdot \frac{\sum_{j=1}^m (rv(c_j) \cdot |c_j|)}{|c_i|} + (1 - \alpha) \cdot quant(assess(c_i)) \quad (5)$$

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

In addition to the above, the final RV score is obtained by applying a normalisation function, which ensures that the total score for any group of descendant nodes of an ideal result element cannot exceed the score achievable if retrieving the ideal node itself. For example, in Figure 1 the two ideal result nodes within the ideal recall-base for the quantisation function *sog* are *sec4* and *sec6*. Since these results represent the best nodes for the user, a system returning these should be ranked above others. However, if another system retrieved all the leaf nodes, it may achieve a better overall score if the total RV score for these nodes exceeds that of the ideal nodes. For example, in Figure 1, the ideal node *sec4* has a score of 0.5, but the total score of its three child nodes is 2.05. The following normalisation function safeguards against this by ensuring that for any $c_j \in S$:

$$\sum_{c \in S} rv(c) \leq rv(c_{ideal}) \quad (6)$$

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

3 Evaluation Setup

3.1 What to Evaluate?

The evaluation of a metric requires a number of tests. Voorhees in [10] identifies two aspects to qualify an evaluation: fidelity and reliability. Fidelity reflects the extent to which an evaluation metric measures what it is intended to measure, while reliability is the extent to which the evaluation results can be trusted. In this paper we concentrate on the latter test. We take the viewpoint of [10] that in a comparative evaluation setting, reliability reflects ‘‘a metric’s ability to rank a better system ahead of a worse system’’. This is, of course, highly dependent on a definition on what makes a system better or worse than another. The basis for such a decision lies within the user satisfaction criterion defined within the given retrieval task.

This criterion in the INEX CO track is (largely) defined by the task definition. According to this, within the CO task, the aim of an XML retrieval system is to point users to the specific relevant portions of documents, where the user’s query contains no structural hints regarding what the most appropriate granularity of relevant XML elements may be. The evaluation of a system’s effectiveness should hence provide a measure with respect to the system’s ability in retrieving such components.

But what exactly are these “most appropriate” components? At the moment, we don’t actually have an exact answer to this in INEX. Intuition dictates that users would prefer elements that contain as much relevant information and as little irrelevant information as possible. Therefore, given a set of possible retrievable components in an arbitrary document (such as an article in INEX), the best elements to return to the user should be those that are “most” exhaustive and “most” specific to the user’s request⁴. However, given two relevant components, one highly exhaustive but only fairly specific ($(e, s) = (3, 2)$) and another which is only fairly exhaustive but highly specific ($(e, s) = (2, 3)$), which one should be regarded as better? The quantisation functions provide a flexible means for addressing this issue by allowing to model various sets of possible preferences, which can be adjusted according a given user model. Based on these preferences, it is then possible to identify the “best” components as those elements that score highest. For example, a user whose preferences are described by the *sog* quantisation function would prefer a $(e, s) = (2, 3)$ component to a $(e, s) = (3, 2)$.

Overall, systems should then rank these “best” components in decreasing order of their quantised scores, i.e. highest scoring elements should be ranked first. In addition, we reason that users do not want to be returned overlapping redundant elements, so systems should be either penalised or at least not rewarded for such redundancy.

Given a user satisfaction criterion, a simple method to evaluate a metric’s reliability is to construct appropriate test data for which we can derive expectations as to what the metric’s outcome should be and check if the expected output is indeed obtained. The expected output is in the form of rankings of systems that meets user expectations. A system ranking is simply an ordered list of runs sorted by decreasing value of effectiveness. For example, according to the user preference that non-overlapping results are preferred to overlapping ones, we would expect that from two systems producing respective result rankings, the former would be regarded as the better system by a reliable evaluation metric.

For our test data, we constructed a number of simulated runs, which are described next.

3.2 Simulated Runs

Each simulated run is populated with components derived from the full recall-base⁵ of all INEX 2004 CO topics, where the selection and ordering of the result components in a run is according to the assumed set of user preferences as defined by the *sog* quantisation function (Equation 2). We constructed the following simulated runs:

irb: a ranked result list derived from the ideal recall-base containing only ideal results selected according to the quantisation function *sog*, where the ordering of the components within the ranking is also according to *sog*. The selection of the ideal results follows the procedure described in Section 2.3. For example, if the relevant nodes in

⁴ Note that most exhaustive and specific here **does not** equate to $(e, s) = (3, 3)$ nodes, but refers to the nodes with the highest available exhaustivity and specificity score. For example, it may be that amongst all the possible retrievable components in an article, the most exhaustive node is $e = 1$, or the most specific node is $s = 2$.

⁵ We used v3.0 of the assessments file for INEX 2004: assessments-3.0.tar.gz.

Figure 1 would form our imaginary full recall-base, then we obtain the following result ranking for our *irb* run: $\{\text{sec6}, \text{sec4}\}$.

frb: a run that contains all relevant components of the full recall-base, where the components are ordered by decreasing quantised value according to the quantisation function *sog*. E.g. for Figure 1, all shown nodes will be included as follows: $\{\text{sec6}, \text{sec6/p1}, \text{sec6/p2}, \text{sec6/ip12}, \text{sec4/p1}, \text{sec4/ip12}, \text{sec4}, \text{sec4/p2}, \text{bdy1}, \text{article1}\}$.

ia: contains all ideal results and all their relevant ascendant nodes ordered by *sog*. E.g. from Figure 1, we obtain: $\{\text{sec6}, \text{sec4}, \text{bdy1}, \text{article1}\}$.

id: contains all ideal results and all their relevant descendant nodes ordered by *sog*. E.g. from Figure 1, we get: $\{\text{sec6}, \text{sec6/p1}, \text{sec6/p2}, \text{sec6/ip12}, \text{sec4/p1}, \text{sec4/ip12}, \text{sec4}, \text{sec4/p2}\}$.

lo: contains all relevant leaf nodes ordered by *sog*. E.g. from Figure 1, we get: $\{\text{sec6/p1}, \text{sec6/p2}, \text{sec6/ip12}, \text{sec4/p1}, \text{sec4/ip12}, \text{sec4/p2}\}$. Note that a leaf node here refers to leaf nodes on the relevant paths within an article, which may be non-leaf nodes within the article file itself (e.g. $\text{sec}[6]/\text{p}[2]$ may have a number of irrelevant descendant nodes).

ao: contains only relevant article nodes ordered by *sog*. E.g. from Figure 1, we get: $\{\text{article1}\}$.

3.3 Expected System Rankings

Based on the user preferences captured by the *sog* quantisation function, systems that return the best components (i.e. highest quantised-scoring elements) should be ranked above others. In addition, based on the intuition that users do not want to be inundated with multiple redundant, nested components, systems that return minimum amount of overlapping results would be preferred.

From these two assumptions, we can derive a relative ranking of simulated runs that should then be matched by a metric if it is to be proved reliable. From the latter assumption, we can reason that the runs should be ranked as follows:

$$irb (0\%) \succeq lo (0\%) \succeq ao (0\%) \succ id (57.9\%) \succ ia (70.7\%) \succ frb (77.5\%)$$

where the percentage values show the overlap ratio among result elements contained in a run and where $a \succ b$ signals that ‘run *a* performs better than run *b*’.

Based on the quantisation function, a metric should rank those systems first that are able to return the best components. Since, the best components are defined by the quantised score of the quantisation function, without looking at the document collection and the actual relevance assessments we cannot safely predict much more than:

$$irb \succeq r$$

where $r \in \{lo, ao, id, ia, frb\}$. This is because the best scoring elements could be, e.g., the leaf or article nodes within a collection (depending entirely on the judgements of the assessor) in which case these runs (*lo*, *ao*) could achieve the same performance as *irb*.

With respect to the runs *id*, *ia*, *frb*, since *irb* is a subset of all these runs, the expectation is that they could possibly produce as good results as the ideal run, but never better.

The combination of the two user satisfaction criteria, if producing conflicting rankings, is currently an open question. It may be solved by defining the relative importance of the two aspects, which may be a parameter of a given user’s model for XML retrieval. Within the XCG metrics, this issue is addressed by considering the overlap of result elements directly within the way the relevance scores are calculated (RV function) for a run.

3.4 Metric Reliability Tests

We evaluated each of the simulated runs using the two metrics⁶. The resulting graphs are shown in Figures 2 and 3.

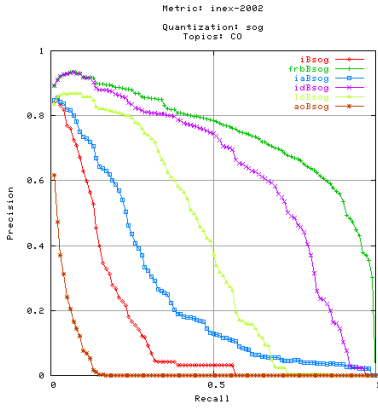


Fig. 2. Results of the inex-2002 metric

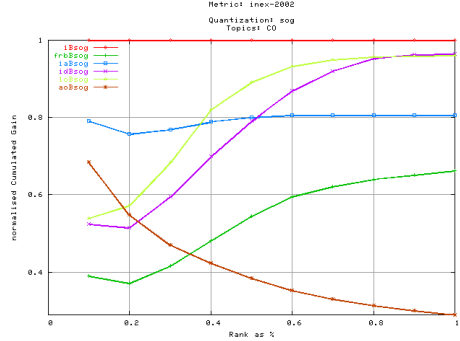


Fig. 3. Results of the nXCG metric

In order to obtain an overall ranking, we use the MAP measure for the inex-2002 metric and the mean average of the normalised cumulated gain values for nXCG (averaged over 10 rank % points). Table 1 summarises the results. The numbers in brackets show the achieved system (run) ranks.

For the inex-2002 metric, both the graph and the MAP results clearly illustrate that better effectiveness is achieved by systems that return not only the most desired components, but also their ascendant (*ia*) or descendant (*id*) elements, hence inundating users with redundant components. In fact, according to this measure only the article-only run (*ao*) has worse performance than the ideal run (*irb*). Best performance is achieved by the run that returns the full recall-base (*frb*).

Looking at the results for the nXCG metric, we can see that best performance is achieved by the ideal run (*irb*), which is registered at a constant 1 normalised cumulated

⁶ Throughout the paper, we used $\alpha = 1$ within the XCG metrics.

gain value (Figure 3). The worse performer is the article-only run (*ao*), followed by the full recall-base run (*frb*). The performance of the remaining runs (*ia*, *id* and *lo*) is evaluated as worse than the ideal, but better than the full recall-base run. Their relative performance to each other will vary depending on the collection and the recall-base.

Table 1. System Rankings produced by inex-2002 and nXCG, based on performance over all INEX 2004 CO topics. For inex-2002 the MAP is shown, for nXCG the mean average normalised cumulated gain (averaged over 10 rank % points) is shown. The numbers in brackets indicate the achieved rank

	inex-2002	nXCG
<i>irb</i>	0.1430 (5)	1.0000 (1)
<i>frb</i>	0.7437 (1)	0.5369 (5)
<i>ia</i>	0.2567 (4)	0.7936 (3)
<i>id</i>	0.6195 (2)	0.7790 (4)
<i>lo</i>	0.3944 (3)	0.8269 (2)
<i>ao</i>	0.0296 (6)	0.4096 (6)

What is clear from the above is that the inex-2002 metric cannot reflect true performance differences when systems return overlapping elements as these can artificially raise the performance indicator. The nXCG metric's ranking of systems, on the other hand, corresponds to the user satisfaction criterion: the retrieval of ideal nodes representing the best nodes for the user (in accordance with a given set of user preferences expressed within a chosen quantisation function) is rewarded, while the retrieval of near-misses is also considered. Systems that retrieve such near-misses can achieve good performances, but cannot surpass an ideal system's score.

In both graphs, the comparison of the article-only (*ao*), the leaf-only (*lo*) and the ideal (*irb*) runs gives an indication of the metrics' capabilities for ranking systems whose output contains no overlapping results. The article-only run achieves worst performance in both cases. With the inex-2002 metric, the ideal run is scored lower than the leaf-only run, while with the nXCG metric, although the leaf-only run's performance is the second best, it never beats the ideal run's effectiveness. This suggests that the inex-2002 metric is able to rank systems when no overlapping results are returned. However, the fact that the leaf-only run seems to perform better than the ideal run points to the need that a similar score-normalisation function to that described in Section 2.3 would be required.

3.5 Top Ten INEX CO Runs

In this section, we list the top ten INEX 2004 runs for both metrics as an indication of how the rankings are effected, see Tables 2 and 3. As it can be seen, amongst the top ten only the run by Carnegie Mellon University appears in both tables, although the University of Amsterdam and the University of Waterloo also appear in both, but with different runs (and one run by Queensland University of Technology is actually ranked 11th by nXCG).

It can be observed that runs with less overlap score better in nXCG, but overlap-free runs are not a sufficient condition for obtaining high values, but relevant nodes

still need to be found. In fact, from the submitted 69 runs, a total of 18 runs have 0% overlap (while several other runs also have minimal overlap, e.g. 1% or less), but their average rank is only 41. There are a couple of runs, which nicely reflect the effect of overlap on the nXCG scores: e.g. the University of Amsterdam submitted these runs:

Table 2. Top ten INEX 2004 runs according to the inex-2002 metric, quant: sog, task: CO. The rank numbers in brackets indicate the rank obtained based on the nXCG metric

rank	Run	MAP	overlap %
1(27)	IBM Haifa Research Lab (CO-0.5-LAREFIENMENT)	0.1327	80.89
2(21)	IBM Haifa Research Lab (CO-0.5)	0.1274	81.46
3(18)	University of Amsterdam (UAMS-CO-T-FBack)	0.1060	81.85
4(10)	LTI, Carnegie Mellon University (Lemur_CO_KStem_Mix02_Shrink01)	0.0941	73.02
5(28)	IBM Haifa Research Lab (CO-0.5-Clustering)	0.0923	81.10
6(26)	LTI, Carnegie Mellon University (Lemur_CO_NoStem_Mix02_Shrink01)	0.0879	74.82
7(11)	Queensland University of Technology (CO_PS_Stop50K_049_025)	0.0839	71.06
8(31)	Queensland University of Technology (CO_PS_099_049)	0.0803	76.81
9(29)	Queensland University of Technology (CO_PS_Stop50K_099_049)	0.0784	75.89
10(25)	University of Waterloo (Waterloo-Baseline)	0.0781	76.32

Table 3. Top ten INEX 2004 runs according to the nXCG metric, quant: sog, task: CO The rank numbers in brackets indicate the rank obtained based on the inex-2002 metric

rank	Run	MAncG	overlap %
1(40)	University of Tampere (UTampere_CO_average)	0.3725	0
2(42)	University of Tampere (UTampere_CO_fuzzy)	0.3699	0
3(41)	University of Amsterdam (UAMS-CO-T-FBack-NoOverl)	0.3519	0
4(34)	Oslo University College (4-par-co)	0.3418	0.07
5(25)	University of Tampere (UTampere_CO_overlap)	0.3328	39.58
6(27)	LIP6 (bn-m1-eqt-porder-eul-o.df.t-parameters-00700)	0.3247	74.31
7(23)	University of California, Berkeley (Berkeley_CO_FUS_T_CMBZ_FDBK)	0.3182	50.22
8(43)	University of Waterloo (Waterloo-Filtered)	0.3181	13.35
9(22)	LIP6 (bn-m2-eqt-porder-o.df.t-parameters-00195)	0.3098	64.2
10(4)	LTI, Carnegie Mellon University (Lemur_CO_KStem_Mix02_Shrink01)	0.2953	73.02

UAMS-CO-T-FBack (81.85% overlap) and UAMS-CO-T-FBack-NoOverl (0% overlap), which are scored as 0.2636 and 0.3521, respectively. Another example may be the runs submitted by the University of Tampere (see Table 3). On the other hand, the runs submitted by RMIT: Hybrid_CRE (82.12% overlap), Hybrid_CRE_specific (0% overlap) and Hybrid_CRE_general (0% overlap) achieve scores of 0.2791, 0.2576 and 0.2540, respectively. The drop in effectiveness score suggests that when reducing overlap, the higher scoring nodes were actually removed from the ranking, leaving lower scoring nodes in the ranking and (presumably) filling the rest of the ranks with irrelevant nodes (provided the three runs are produced from the same baseline, of course).

Note that a detailed analysis of the system rankings produced by the two metrics is planned for a separate paper.

4 Conclusions

In this paper we investigated how closely the output of the two metrics, inex-2002 and nXCG, reflect the user satisfaction criteria defined within the INEX CO task. The results confirm the weaknesses of the inex-2002 metric reported in [4, 5], but show that with an appropriate quantisation function (e.g. when leaf nodes represent the best nodes) or with an arbitrary quantisation function when combined with a normalisation method, the inex-2002 metric is able to produce system rankings that match the evaluation criteria, provided no overlapping results are returned by the systems.

We also described and further developed the XCG metrics, which produced promising results in our metric reliability test. A weakness of the XCG metrics, however, is that they produce effectiveness scores for a given rank (or rank %) and not for recall. To address this issue, Gabriella Kazai is currently working on a version of the metric that is able to give recall related performance indicators. She is also working on an extension of the generalised Precision and Recall measures introduced in [7]. These will be published in the near future.

In the future, we also hope to be able to derive better user models and hence arrive at more accurate user satisfaction criteria based on the outcome of the INEX 2004 interactive track.

References

1. 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.
2. N. Gövert and G. Kazai. Overview of the INitiative for the Evaluation of XML Retrieval (INEX) 2002. In N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas, editors, *Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX)*. Dagstuhl, Germany, December 8–11, 2002, ERCIM Workshop Proceedings, pages 1–17, Sophia Antipolis, France, March 2003. ERCIM. <http://www.ercim.org/publication/ws-proceedings/INEX2002.pdf>.
3. K. Järvelin and J. Kekäläinen. Cumulated Gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (ACM TOIS)*, 20(4):422–446, 2002.
4. G. Kazai. Report of the inex 2003 metrics working group. In N. Fuhr, M. Lalmas, and S. Malik, editors, *Proceedings of the 2nd Workshop of the INitiative for the Evaluation of XML retrieval (INEX)*, Dagstuhl, Germany, December 2003, pages 184–190, April 2004.
5. G. Kazai, M. Lalmas, and A. de Vries. The overlap problem in content-oriented XML retrieval evaluation. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, 2004.*, pages 72–79. ACM, July 2004.
6. G. Kazai, S. Masood, and M. Lalmas. A study of the assessment of relevance for the inex'02 test collection. In *Advances in Information Retrieval, Proceedings of the 26th European Conference on IR Research (ECIR), Sunderland, UK*, Lecture Notes in Computer Science. Springer, April 2004.

7. 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.
8. V. Raghavan, P. Bollmann, and G. Jung. A critical investigation of recall and precision. *ACM Transactions on Information Systems*, 7(3):205–229, 1989.
9. T. Tombros, B. Larsen, and S. Malik. The interactive track at INEX 2004. In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlavik, editors, *Proceedings of the 3rd Workshop of the INitiative for the Evaluation of XML retrieval (INEX), Dagstuhl, Germany, December 2004*, 2005.
10. E. M. Voorhees. Overview of the TREC 2003 question answering track. In *Text REtrieval Conference, Gaithersburg*, 2003.