INEX 2005 Evaluation Measures

Gabriella Kazai and Mounia Lalmas

Queen Mary, University of London,
Mile End Road, London, UK
{gabs, mounia}@dcs.qmul.ac.uk

Abstract. This paper describes the official measures of retrieval effectiveness employed in INEX 2005: the eXtended Cumulated Gain (XCG) measures. In addition, results of correlation analysis are reported, examining the correlation between the employed quantisation functions and the different measures for the INEX 2005 ad-hoc tasks.

1 Introduction

In INEX 2005, a new set of measures, the eXtended Cumulated Gain (XCG) measures, were introduced with the aim to provide a suitable evaluation framework, where the dependency among XML document components can be taken into account. In particular, two aspects of dependency were considered: 1.) near-misses, which are document components that are structurally related to relevant components, such as a neighbouring paragraph or a container section, and 2.) overlap, which regards the situation when the same text fragment is referenced multiple times, as in the case when a paragraph and its container section are both retrieved.

The XCG measures are an extension of the Cumulated Gain based measures proposed in [2]. These measures were chosen as they have been developed specifically for graded relevance values and with the aim to allow IR systems to be credited according to the retrieved documents' degree of relevance. The motivation for the XCG measures was to extend the CG metrics for the problem of content-oriented XML IR evaluation, where the dependency of XML elements is taken into account. The extension lies partly in the way the gain value for a given document component is calculated via the definition of so-called relevance value (RV) functions, and partly in the definition of the ideal recall-bases. The former allows to consider the dependency of result elements within a system's output, while the latter regards the dependency of elements within the test collection's recall-base[1].

The new measures aim to overcome the limitations of `inex-eval`, the previous official measure of INEX. One such issue is that `inex-eval` is not well-suited to handle multiple degrees of relevance. In addition, `inex-eval` has no mechanisms for both rewarding partial scores to near-misses and to handle overlap.

[1] The term recall-base refers to the collection of assessments within the test collection that forms the ground-truth for the evaluation experiments.

2 Definition of an ideal recall-base

As described in [6], in INEX 2005 relevance assessments were given according to two relevance dimensions: *exhaustivity* (e) and *specificity* (s). The relevance degree of an assessed component, given by the combined values of exhaustivity and specificity, is denoted as (e, s), where $e \in \{?, 0, 1, 2\}$ and $s \in [0, 1]$. The value of $e =?$ is used to denote elements judged as 'too small'. Within the evaluation, $e =?$ is equated to $e = 0$.

An important property of the exhaustivity dimension is its propagation effect, reflecting that if a component is relevant to a query, then all its ascendant elements will also be relevant. Due to this property, all nodes along a relevant path[2] are always relevant (with varying degrees of relevance), hence resulting in a recall-base comprised of sets of overlapping elements.

In order to evaluate tasks based on the Focussed retrieval strategy[3], where overlap is not allowed, it is necessary to remove overlap from the collected assessments in the recall-base. For this purpose, we define an ideal recall-base as a subset of the full recall-base, where overlap between relevant reference elements is removed so that the identified subset represents the set of ideal answers, i.e. those elements that should be returned to the user.

The selection of ideal nodes into the ideal recall-base is done through the definition of preference relations on the possible (e, s) pairs and a methodology for traversing an article's XML tree. The preference relations are given by quantisation functions, while the following methodology is adopted to traverse an XML tree and select the ideal nodes: Given any two components on a relevant path, the component with the higher quantised score is selected. In case two components' scores are equal, the one higher in the tree is chosen (i.e. parent/ascendant). The procedure is applied recursively to all overlapping pairs of components along a 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 shortest one.

The use of an ideal recall-base supports the evaluation viewpoint (needed for the Focussed strategy) whereby components in the ideal recall-base *should* be retrieved, while the retrieval of near-misses *could* be rewarded as partial successes, but other systems *need not* be penalised for not retrieving such near-misses.

The following quantisation functions are used in INEX 2005: $quant_{strict}$ (Equation 1), $quant_{gen}$ (Equation 2) and $quant_{genLifted}$ (Equation 3). The strict function models a user for whom only fully specific and highly exhaustive components are considered worthy. The generalised (gen) function credits document components according to their *degree* of relevance, hence allowing to model varying levels of user satisfaction gained from not fully specific and highly exhaustive,

[2] 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. $quant(e, s) > 0$) that has no or only non-relevant descendants.

[3] CO.Focussed, COS.Focussed, CO.FetchBrowse and COS.FetchBrowse

but still relevant components or near-misses. Both $quant_{strict}$ and $quant_{gen}$ functions ignore elements assessed as 'too small', since by default these are treated as $e = 0$. In order to consider too small elements within the evaluation, the $quant_{genLifted}$ quantisation function is introduced, which adds +1 to lift all values of exhaustivity[4]. The effect of this in the evaluation is that it allows the scoring of too small elements as near-misses.

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

$$quant_{gen}(e, s) := e \cdot s \tag{2}$$

$$quant_{genLifted}(e, s) := (e + 1) \cdot s \tag{3}$$

3 eXtended Cumulated Gain (XCG) measures

The XCG measures are a family of evaluation measures that are an extension of the cumulated gain (CG) based metrics of [2] and which aim to consider the dependency of XML elements (e.g. overlap and near-misses) within the evaluation. The XCG measures include the user-oriented measures of normalised extended cumulated gain ($nxCG$) and the system-oriented effort-precision/gain-recall measures (ep/gr).

3.1 Normalised *xCG* (*nxCG*)

We define xCG as a vector of accumulated gain. Given a ranked list of document components where the element IDs are replaced with their relevance scores, the cumulated gain at rank i, denoted as $xCG[i]$, is computed as the sum of the relevance scores up to that rank:

$$xCG[i] := \sum_{j=1}^{i} xG[j] \tag{4}$$

For each query, an ideal gain vector, xI, can be derived by filling the rank positions with the relevance scores of all documents in the recall-base (or as in the case of the Focussed strategy, with the relevance scores of all elements in the ideal recall-base) in decreasing order of their degree of relevance. The corresponding cumulated ideal gain vector is referred to as xCI.

A retrieval run's xCG vector can then be compared to this ideal ranking by plotting both the actual and ideal cumulated gain functions against the rank

[4] Note that this is only applied to relevant elements of the recall-base, hence non-relevant nodes remain as $e = 0$.

position. By dividing the xCG vectors of the retrieval runs by their corresponding ideal xCI vectors, we obtain the normalised xCG ($nxCG$) measure:

$$nxCG[i] := \frac{xCG[i]}{xCI[i]} \tag{5}$$

For a given rank i, the value of $nxCG[i]$ reflects the relative gain the user accumulated up to that rank, compared to the gain he/she could have attained if the system would have produced the optimum best ranking. For any rank, the normalised value of 1 represents ideal performance.

Systems may be compared at various cutoff values, e.g. $nxCG[1]$ or $nxCG[500]$. In addition, we may average $nxCG[i]$ scores up to a given rank as:

$$MAnxCG[i] := \frac{\sum_{j=1}^{i} nxCG[j]}{i} \tag{6}$$

An advantage of this latter measure is that it reflects on the quality of the ranking, whereas $nxCG$ reports a set-based value, measured at a single point in the ranking.

3.2 Calculating an element's relevance value

The definition of the $nxCG$ measure is based on the gain value, $xG[i]$, that a user obtains when examining a returned result component at a given rank i. In this section we detail how this gain is calculated.

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 or gain value to the user. The gain value will depend on the returned component's exhaustivity and specificity (i.e. its (e, s) values) as well as on how overlap and near-misses are handled. I.e. for the Focussed tasks overlap and near-misses are taken into account, whereas for the Thorough tasks the relevance value is a direct function of (e, s).

Focussed tasks. Focussed tasks are evaluated based on the assumption that returned overlapping components represent only as much gain as the amount of new relevant information they contain and that the retrieval of near-misses is considered useful to the user. The evaluation parameter that represents this setup is referred to as "overlap=on".

Overlap: Following the assumption that any already viewed components become irrelevant to the user, we define the following result-list dependent relevance

value (RV) function, $rv(c_i)$:

$$rv(c_i) := \begin{cases} quant(assess(c_i)) & \text{if} \quad c_i \text{ has not yet been seen,} \\ (1-\alpha) \cdot quant(assess(c_i)) & \text{if} \quad c_i \text{ has been fully seen,} \\ \alpha \cdot \frac{\sum_{j=1}^{m} (rv(c_j) \cdot |c_j|)}{|c_i|} & \\ \quad +(1-\alpha) \cdot quant(assess(c_i)) & \text{if} \quad c_i \text{ has been partially seen before.} \end{cases} \tag{7}$$

where $assess(c_i)$ is a function that returns the assessment value pair (e, s) for the i-th component in the ranking if it is given within the recall-base and $(0, 0)$ otherwise. The function $quant(\cdot)$ is a quantisation function, m is the number of c_i's child nodes and $|\cdot|$ is the length of an element (in characters or words). The $\alpha \in [0, 1]$ weighting factor reflects a user's intolerance to being returned redundant components or component-parts. The higher the α value, the less value a redundant relevant component represents to the user.

According to the above equation, for a not-yet-seen component, the component's relevance value is only dependent on the component's quantised assessment value: $quant(assess(c_i))$. For a component that has been already fully seen by the user, the component's quantised assessment value, $quant(assess(c_i))$, is weighted by $(1 - \alpha)$. For example, using $\alpha = 1$, which represents a user who does not tolerate already viewed components, we obtain an RV score of 0 for a fully seen component, reflecting that it represents no value to the user any more. Finally, if a component has been seen only in part before, then its relevance value is calculated recursively based on the relevance values of its descendant nodes, combined with its own quantised assessment value. The intolerance weighting factor of α is again used to modify the value attributed to already seen components. For example, using $\alpha = 1$ means that only not-yet-seen sub-components will be scored, while using $\alpha = 0$ will return the unmodified quantised score of the component, regardless how much of it the user has seen already.

Near-misses: To consider the retrieval of near-misses within the evaluation, we reward a partial score for the retrieval of non-ideal elements that are structurally related to ideal components. This year, only those relevant elements (as per quantisation function) of the full recall-base were considered near-misses which were not included in the ideal recall-base.

Given this set of near-misses and the ideal recall-base, the XCG measures are applied such that the ideal gain vector of a query, xI, is derived from the ideal recall-base, and the gain vectors, xG, corresponding to the system runs under evaluation are based on the full recall-base. The relevance score of a near-miss component is calculated by Equation 7.

Before the final gain value can be assigned to $xG[i]$, we apply a dependency normalisation function, which ensures that the total score for any sub-tree of an ideal node cannot exceed the maximum score achievable when the ideal node itself is retrieved. For example, an ideal node may have a large number of relevant

child nodes whose total RV score may exceed that of the ideal node. The following dependency normalisation function, rv_{norm}, safeguards against this by ensuring that for any $c_j \in S$, $rv(c_i) + \sum^{S} rv(c_j) \leq rv(c_{ideal})$ holds:

$$rv_{norm}(c_i) = min(rv(c_i), rv(c_{ideal}) - \sum^{S} rv(c_j)) \tag{8}$$

where c_{ideal} is the ideal node that is on the same relevant path as c_i, S is the set of nodes in the ideal node's sub-tree that have already been retrieved (before c_i).

The final gain value: The final gain value of a result element in a ranked output list of an XML IR system, taking into account near-misses and overlaps, is given by the normalised relevance score of:

$$xG[i] := rv_{norm}(c_i) \tag{9}$$

where $rv_{norm}(c_i)$ is defined in Equation 8, $rv(c_i)$ is given in Equation 7.

Thorough tasks. Thorough tasks were evaluated using the full recall-base as the basis for deriving the ideal gain vectors. The evaluation parameter that represents this setup is referred to as "overlap=off".

For the Thorough tasks, systems obtain a score for returning as many of the relevant reference elements as possible, including all overlapping nodes. The gain value of a result element in a ranked output list is calculated as:

$$xG[i] := rv(c_i) := quant(assess(c_i)) \tag{10}$$

where $assess(c_i)$ is a function that returns the assessment value pair (e, s) for the i-th component in the ranking if it is given within the recall-base and $(0, 0)$ otherwise. The function $quant(\cdot)$ is a quantisation function.

3.3 Effort-precision and gain-recall: *ep/gr*

The cumulated gain based measures described so far provide a recall-oriented view of effectiveness at fixed rank positions. Next we want to measure the amount of effort required of the user to reach a given level of cumulated gain when scanning a given ranking compared to an ideal ranking. The horizontal line drawn at the cumulated gain value of r, shown in Figure 1, illustrates this view. Based on this, we define effort-precision ep as:

$$ep[r] := \frac{i_{ideal}}{i_{run}} \tag{11}$$

where i_{ideal} is the rank position at which the cumulated gain of r is reached by the ideal curve and i_{run} is the rank position at which the cumulated gain of r is reached by the system run. A score of 1 reflects ideal performance, i.e. when the user needs to spend the minimum necessary effort to reach a given level of gain.

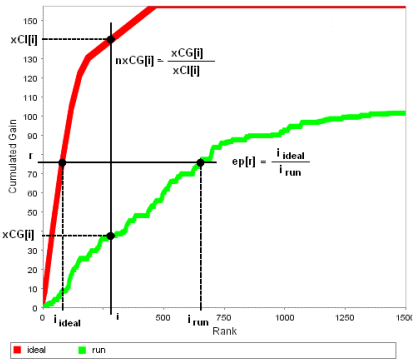

Fig. 1. Calculation of $nxCG$ and effort-precision (ep)

Effort-precision can be calculated at arbitrary gain-recall points, where gain-recall is calculated as the cumulated gain value divided by the total achievable cumulated gain [5]:

$$gr[i] := \frac{xCG[i]}{xCI[n]} = \frac{\sum_{j=1}^{i} xG[j]}{\sum_{j=1}^{n} xI[j]} \tag{12}$$

where n is the total number of documents in the recall-base.

The meaning of effort-precision at a given gain-recall value is the amount of relative effort (where effort is measured in terms of number of visited ranks) that the user is required to spend when scanning a system's output ranking compared to the effort an ideal ranking would take in order to reach a given level of gain relative to the total gain that can be obtained.

This method follows the same viewpoint as standard precision/recall, where recall is the control variable and precision the dependent variable. As with precision/recall, interpolation techniques are necessary to estimate effort-precision values at non-natural gain-recall points (e.g. when calculating effort-precision at standard recall points). We adopted linear interpolation for estimating values between two natural recall points, i.e. using straight lines ($y = ax + b$).

As with standard precision/recall, the non-interpolated mean average effort-precision, denoted as $MAep$, is calculated by averaging the effort-precision values measured at natural recall-point, i.e. whenever a relevant XML element is found in the ranking. For non-retrieved relevant elements the score of 0 is used. Note that calculating $MAep$ still requires interpolation over the ideal curve as natural recall points of a run may not coincide with natural recall points of the ideal ranking.

We also calculate an average over the interpolated effort-precision values at standard recall points, i.e. $[0.01, 0.02, ..., 1]$, which we refer to as $iMAep$.

Analogue to recall/precision graphs, we plot effort-precision against gain-recall and obtain a detailed summary of a system's overall performance.

3.4 The Q and R measures

A criticism of the $nxCG$ measures is that they do not average well across topics [3] (in [8]). The reason for this is that as the total number of relevant documents differs across topics, so does the upper bound performance at fixed ranks.

A solution has been suggested in [8] in the form of the following measures. Here the explicit incorporation of the rank position in the denominator ensures that performance is calculated against an always increasing ideal value:

$$Q - measure = \frac{1}{R} \sum_{j=1}^{i} isrel(d_j) \frac{cbg(j)}{cig(j) + j} \tag{13}$$

where R is the total number of relevant documents, d_j is the document retrieved at rank j, $isrel(\cdot)$ is a binary function that returns 1 if the document is relevant (to any degree) and 0 otherwise. The function $cbg(\cdot)$ is a so-called cumulated bonus gain function, which is defined as $cbg(i) := bg(i) + cbg(i-1)$, where $bg(i) := g(i) + 1$ if $g(i) > 0$ and $bg(i) := 0$ otherwise, and $g(i)$ is the gain value at rank i. The function $cig(\cdot)$ is the cumulated bonus gain derived for the ideal vector (analogue to $cbg(\cdot)$).

$$R - measure = \frac{cbg(R)}{cbg(R) + R} \tag{14}$$

We employ extended versions of the above measures, adapted to XML through the definition of $g(i) := xG[i]$. We refer to these measures as Q and R.

4 Results reported in INEX 2005

The results of the following measures were reported in INEX 2005:

- Effort-precision/gain-recall (ep/gr) graphs.
- Non-interpolated mean average effort-precision ($MAep$).
- Interpolated mean average effort-precision ($iMAep$).
- Normalised xCG ($nxCG$) graphs, plotting the $nxCG$ value obtained at $[1, 2, ..., 100]\%$ of the length of the output list, i.e. 1500.
- Normalised xCG ($nxCG$) at various fixed ranks (e.g. $nxCG[25]$).
- Mean average $nxCG$ at various fixed ranks (e.g. $MAnxCG[50]$).
- Q and R.

The official system-oriented evaluation was based on the ep/gr measures, with $MAep$ being the main overall performance indicator. The official user-oriented evaluation was based on the $nxCG[10]$, $nxCG[25]$ and $nxCG[50]$ performance indicators. All results are accessible on the INEX 2005 website[5].

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

5 EvalJ

All measures have been implemented within a single Java project, the EvalJ evaluation package, which can be downloaded from SourceForge.net[6]. Instruction for how to download the project are at https://sourceforge.net/cvs/?group_id=136430. Alternatively, installer files can be accessed from http://evalj.sourceforge.net/. There is a README included within EvalJ, detailing how to get going and how to run the various evaluation measures.

6 Evaluating different tasks

The XCG measures in EvalJ take several parameters, which define how e.g. overlap is to be handled. These parameters are read at run time from a config file. A config file, `inex2005.prop` is provided within EvalJ, containing the official parameter settings for INEX 2005. These are detailed below.

Note that the difference between the CO and COS evaluations was that the former was based on all assessed CO+S topics (29 topics), whereas the latter was evaluated using only those assessed topics that contained a $<castitle>$ element (19 topics). In this case the assessment pool IDs were given within the `POOL` parameter to filter the total set of assessments. The pool IDs represent the following topics: 202, 203, 205, 207, 208, 210, 212, 216, 219, 222, 223, 228, 229, 230, 232, 233, 234, 236, 239.

6.1 CO.Focussed and COS.Focussed

These tasks were evaluated using the "overlap=on" option, which means that overlap and near-misses are considered within the evaluation. The ideal recall-base is generated automatically within the evaluation based on the selected quantisation function and the methodology described in section 2.

```
TASK: CO.Focussed
METRICS: nxCG, ep/gr, q
ALPHA: 1.0
OVERLAP: on
QUANT_FUNCTIONS: gen, strict, genLifted
DCV: 10, 25, 50
ASSESSMENTS_DIR: .../adhoc2005/official/CO+S/*/
QUERY_TYPE: CO+S
SUBMISSIONRUNS_DIR: .../inex2005_runs/*/

TASK: COS.Focussed
METRICS: nxCG, ep/gr, q
ALPHA: 1.0
OVERLAP: on
QUANT_FUNCTIONS: gen, strict, genLifted
```

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

```
DCV: 10, 25, 50
ASSESSMENTS_DIR: .../adhoc2005/official/CO+S/*/
QUERY_TYPE: CO+S
SUBMISSIONRUNS_DIR: .../inex2005_runs/*/
POOLS: 275,297,273,353,300,283,309,327,292,319,321, 325,301,315,265,361,
       364,349,279
```

6.2 CO.Thorough and COS.Thorough

These tasks were evaluated using the "overlap=off" option, which means that overlap is tolerated within the evaluation. Therefore, no ideal recall-base is generated and the gain value of a component is only a function of its exhaustivity and specificity values, regardless if it overlaps or not with a previously returned element.

```
TASK: CO.Thorough
METRICS: nxCG, ep/gr, q
ALPHA: 1.0
OVERLAP: off
QUANT_FUNCTIONS: gen, strict, genLifted
DCV: 10, 25, 50
ASSESSMENTS_DIR: .../adhoc2005/official/CO+S/*/
QUERY_TYPE: CO+S
SUBMISSIONRUNS_DIR: .../inex2005_runs/*/

TASK: COS.Thorough
METRICS: nxCG, ep/gr, q
ALPHA: 1.0
OVERLAP: off
QUANT_FUNCTIONS: gen, strict, genLifted
DCV: 10, 25, 50
ASSESSMENTS_DIR: .../adhoc2005/official/CO+S/*/
QUERY_TYPE: CO+S
SUBMISSIONRUNS_DIR: .../inex2005_runs/*/
POOLS: 275,297,273,353,300,283,309,327,292,319,321,325,301,315,265,361,
       364,349,279
```

6.3 CO.FetchBrowse and COS.FetchBrowse

The evaluation methodology for these task is different from all other tasks in that two separate evaluation scores were calculated: an article-level and an element-level score.

The article-level score regards a system's ability to find relevant documents in the first place. To obtain this score, we first filter the recall-base to contain only those article nodes that have at least one relevant element according to the chosen quantisation. E.g. for strict quantisation, only those articles are kept that contain at least one highly exhaustive and fully specific element. The ideal gain vector is obtained by sorting the filtered set by quantised score. Since articles

do not overlap, the process is the same for both overlap=on and off modes. We compare the obtained ideal gain vector to the list of article nodes that is *derived* from a system run. To derive the list of articles from a run, we reduce each XML element in the run to its article root and keep from any two duplicate entries the first occurrence (e.g. from $<a1/e1, a1/e2, a2>$ we derive $<a1, a2>$).

The element-level score reflects a system's ability to locate relevant elements within an article. Each cluster of an article and its contained elements are examined individually. The order of clusters is ignored here (it was already considered at the article-level), but the order of elements within a cluster is taken into account. The recall-base for a given cluster consists of the relevant elements within the given article (as per quantisation and overlap setting), where elements are ordered in decreasing quantised value. The list of elements within a cluster of a run is then compared against the cluster's recall-base directly. The individual cluster-scores are then averaged over all clusters and then over all queries.

We only report effort-precision/gain-recall measures for the FetchBrowse tasks, as the selection of an appropriate document cutoff value for $nxCG$ is an open question (due to the small number of relevant elements within each cluster-recall-base).

FetchBrowse tasks have been evaluated both with "overlap=on" and "overlap=off" options. The "overlap=off" option evaluates systems according to the Thorough strategy assumption. The "overlap=on" option evaluates systems according to the Focussed strategy assumption.

```
TASK: CO.FetchBrowse
METRICS: ep/gr
ALPHA: 1.0
OVERLAP: on, off
QUANT_FUNCTIONS: gen, strict, genLifted
DCV: 10, 25, 50
ASSESSMENTS_DIR: .../adhoc2005/official/CO+S/*/
QUERY_TYPE: CO+S
SUBMISSIONRUNS_DIR: .../inex2005_runs/*/

TASK: COS.FetchBrowse
METRICS: ep/gr
ALPHA: 1.0
OVERLAP: on, off
QUANT_FUNCTIONS: gen, strict, genLifted
DCV: 10, 25, 50
ASSESSMENTS_DIR: .../adhoc2005/official/CO+S/*/
QUERY_TYPE: CO+S
SUBMISSIONRUNS_DIR: .../inex2005_runs/*/
POOLS: 275,297,273,353,300,283,309,327,292,319,321,325,301,315,265,361,
       364,349,279
```

6.4 SSCAS, SVCAS, VSCAS and VVCAS

These tasks are evaluated based on the Thorough task assumption, with "overlap=off".

```
TASK: SSCAS #or SVCAS, VSCAS, VVCAS
METRICS: nxCG, ep/gr, q
ALPHA: 1.0
OVERLAP: off
QUANT_FUNCTIONS: gen, strict, genLifted
DCV: 10, 25, 50
ASSESSMENTS_DIR: .../official/SSCAS/ #or .../SVCAS/, .../VSCAS/, .../VVCAS/
QUERY_TYPE: CAS
SUBMISSIONRUNS_DIR: .../inex2005_runs/*/
```

7 Correlation analysis of results

7.1 Correlation of XCG measures

We examined correlation among the different XCG measures by calculating the Kendall τ correlation [1] between their resulting respective system rankings.

The correlation measure of Kendall's τ is a nonparametric measure of the agreement between two rankings. It computes the distance between two rankings as the minimum pair-wise adjacent swaps necessary to turn one ranking into the other. The distance is normalised by the number of items being ranked such that two identical rankings produce a correlation of 1 and two rankings that are a perfect inverse of each other produces a score of -1. The expected correlation of two rankings chosen at random is 0. Previous work has considered all rankings with correlations greater than 0.9 as equivalent and rankings with correlation less than 0.8 as containing noticeable differences [9].

Table 1 shows the averaged correlation values over the following ad-hoc tasks: CO.Focussed, CO.Thorough, COS.Focussed, COS.Thorough, SSCAS, SVCAS, VSCAS, VVCAS. This means that the average correlation for each measure was calculated over 24 correlation scores: 8 tasks, each having three variants for the three quantisation functions. Although different tasks and different quantisations resulted in somewhat different correlation values amongst the different measures, the overall general trend is reflected within this table.

The low levels of correlation between the overall performance measures, e.g. *MAep*, and the fixed cutoff measures, e.g. $nxCG$[25], show that these measures reflect different aspects of a system's performance and that systems which perform well according to one criterion may not do so well according to another. However, since *MAep*, *iMAep* and Q are highly correlated, it may be enough to report only one of these measures in the future. $MAnxCG$ and $nxCG$ at the various cutoffs also report fairly similar results, and hence the evaluation could focus just on one or two of these measures.

7.2 Correlation of quantisation functions

Next, we examined correlation among the different quantisation functions by calculating the Kendall τ correlation between the resulting respective system rankings.

Table 1. Averaged correlation of the XCG measures over 8 ad-hoc sub-tasks

	MAnxCG			nxCG			MAep	iMAep	Q
	[10]	[25]	[50]	[10]	[25]	[50]			
MAnxCG[25]	0.85								
MAnxCG[50]	0.77	**0.90**							
nxCG[10]	0.83	**0.91**	0.86						
nxCG[25]	0.71	0.85	**0.92**	0.82					
nxCG[50]	0.64	0.76	0.85	0.74	0.85				
MAep	0.64	0.72	0.75	0.70	0.74	0.76			
iMAep	0.63	0.71	0.74	0.70	0.73	0.75	**0.93**		
Q	0.60	0.69	0.72	0.68	0.71	0.73	**0.93**	0.87	
R	0.58	0.66	0.69	0.65	0.69	0.71	0.83	0.78	0.85

Table 2 shows the averaged correlation values over the following ad-hoc tasks: CO.Focussed, CO.Thorough, COS.Focussed, COS.Thorough, SSCAS, SVCAS, VSCAS, VVCAS. This means that the average correlation for each measure was calculated over 80 correlation scores: 8 tasks, each having each having results reported for ten measures. Although different tasks and different measures resulted in somewhat different correlation values amongst the different quantisation functions, the overall general trend is reflected within this table.

The low correlation between the strict and both versions of the generalised quantisation functions indicates that these quantisations result in very noticeable result differences among systems. It is clear that systems that perform well according to the strict quantisation may not suit a user represented by the generalised quantisation functions. On the other hand, the two generalised quantisation functions show rather similar behaviour, although their averaged correlation is still below 0.9. This suggests that 'too small' elements do have some effect on system performance. To reflect this, future INEX evaluations may limit the number of quantisation functions to the use of the $quant_{strict}$ and $quant_{genLifted}$ functions.

Table 2. Averaged correlation of the quantisation functions over 8 ad-hoc sub-tasks

	strict	gen
gen	0.56	
genLifted	0.60	0.89

8 Conclusions

INEX 2005 introduced a new set of measures, the XCG measures, with the aim to address limitations of the previous official measure (`inex-eval`) and provide a suitable evaluation framework, where the dependency among XML document components can be taken into account.

Future work on the XCG measures will aim at 1.) exploring alternative methods for deriving ideal recall-bases, 2.) incorporating new relevance value functions that allow scoring near-misses that are not already included in the full recall-base, e.g. non-relevant sibling nodes, and 3.) investigating various discounting functions. In addition, we are aiming to conduct further studies into the measures' reliability and sensitivity, extending on previous work in [4].

Furthermore, other quantisation functions are currently also being investigated, including weighted versions of the harmonic mean function [7].

9 Acknowledgments

We would like to thank many participants of the INEX Workshop on Element Retrieval Methodology, Glasgow, July 2005 for their comments and suggestions, which essentially led to the use of the XCG framework at INEX 2005. We are also and especially grateful to Saadia Malik, Benjamin Piwowarski and Arjen de Vries for their invaluable help with hands-on work as well as for the many useful comments and email discussions. We thank Shlomo Geva and Jovan Pehcevski for their helpful comments and questions. Finally, many thanks to all members of the organisers mailing list for their contributions to many of the email discussions.

References

1. W. Conover. *Practical Non-Parametric Statistics, 2nd edn.* John Wiley & Sons, Inc., New York, NY, USA, 1980.
2. K. Järvelin and J. Kekäläinen. Cumulated Gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (ACM TOIS)*, 20(4):422–446, 2002.
3. N. Kando, K. Kuriyama, and M. Yoshioka. Information retrieval system evaluation using multi-grade relevance judgements - discussion on averageable single-numbered measures (in japanese). Technical report, 2001.
4. G. Kazai and M. Lalmas. eXtended Cumulated Gain Measures for the Evaluation of Content-oriented XML Retrieval. *ACM Transactions on Information Systems (ACM TOIS)*, To appear.
5. 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.
6. S. Malik, G. Kazai, M. Lalmas, and N. Fuhr. Overview of inex 2005. In N. Fuhr, M. Lalmas, S. Malik, and G. Kazai, editors, *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005), Schloss Dagstuhl, 28-30 November 2005*, volume 3977 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 2006.
7. C. J. V. Rijsbergen. *Information Retrieval.* Butterworth-Heinemann, Newton, MA, USA, 1979. Available at http://www.dcs.glasgow.ac.uk/Keith/Preface.html.
8. T. Sakai. New performance metrics based on multigrade relevance: Their application to question answering. In *NTCIR Workshop 4 Meeting Working Notes*, June 2004.
9. E. M. Voorhees. Evaluation by highly relevant documents. In *SIGIR'01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82, New York, NY, USA, 2001. ACM Press.