Structural Relevance in XML Retrieval Evaluation

M. S. Ali
University of Toronto
sali@cs.toronto.edu

Mariano P. Consens
University of Toronto
consens@cs.toronto.edu

Mounia Lalmas
Queen Mary, University of London
mounia@dcs.qmul.ac.uk

ABSTRACT

Determining the effectiveness of XML retrieval systems is crucial for improving information retrieval from XML document collections. Traditional effectiveness measures do not address the problem of overlap in the recall-base. At the Initiative for the Evaluation of XML retrieval (INEX), extended cumulated gain (XCG) was developed to address overlap. It works by comparing the cumulated score of a retrieval result to an ideal result. The use of XCG is contingent on being able to define an ideal recall-base for every topic.

This paper introduces an alternative approach called structural relevance (SR) which addresses overlap by extending relevance to overlapping, non-disjoint elements. SR models the user process of browsing overlapped elements in a ranked list using XML summaries (bisimilarity-based graph representations of the structure of a collection of XML documents) to describe the user process in terms of the structure of the collection. We show how SR is incorporated into traditional relevance-based measures and illustrate the behavior of SR in comparison to XCG. Our results suggest that SR can evaluate XML retrieval systems as effectively as XCG without requiring an ideal recall-base.

1. INTRODUCTION

The Initiative for the Evaluation of XML retrieval (INEX) is a collaborative, international effort to develop effective XML retrieval systems. At INEX, a recognized challenge in evaluating XML retrieval systems has been the overlap problem [11]. Overlap occurs when a user finds a ranked element more than once in the process of evaluating a ranked list of elements. Numerous proposals have been made to address the problem [2, 20, 12]. Overlap occurs because a user can access retrieval elements directly from either the ranked list or from following the structural paths between elements while browsing. Overlapped elements result in poor user satisfaction of retrieval systems [21] because the user perceives that the search results contain repetitive answers [5]. Moreover, overlap invalidates the use of traditional relevance-based effectiveness measures because it repudiates the basic information retrieval assumption of independence of retrieval elements when determining their relevance.

The official metric for evaluation of system effectiveness for INEX 2002 to 2004 was precall [7]. Precall is the expected precision of a result at a given recall level where the system is weakly ordered [18] or, in other words, where a user assesses tie-ranked elements in a random order. Precall does not address overlap in the recall-base and this has motivated the development of other measures which do. In 2005, INEX adopted the extended cumulated gain (XCG) as its official measure [10]. XCG is based on cumulated gain (CG) [8] where each element in a ranked list contributes to the overall relevance, or what is referred to as the gain, of the list. The cumulated gain is calculated by summing the scores of elements from the head of the list to a fixed rank position. An ideal recall-base defines the maximum possible cumulated gain for all ranks. The ratio of cumulated gain to ideal cumulated gain shows how closely a given ranked list compares to the ideal list. XCG extends CG by incorporating heuristics to model the effects of overlap on relevance scores. It is important to recognize XCG's dependency on the methodology used to build an ideal recall base (as shown in [9]).

We approach the problem of overlap in the recall-base by revisiting the notion of relevance for non-disjoint (*i.e.*, overlapping) elements. In this regard, we differentiate between the relevance of a single retrieval element and the relevance of a retrieval element as a member of a set of elements. A human judge assesses the relevance of an element to a topic with the assumption that the element is independent of all other elements; whereas, the relevance of an element in a set of non-disjoint elements is the result of how a human uses the set to fulfill their information need. In this regard, isolation, which is the probability of first encountering an element from a ranked list composed of a given set of elements, is a measure of the expected relevance of the element as a member of the given set. The overall relevance of a ranked list of structurally non-disjoint elements must be assessed in terms of a user model that describes both how the list is used and how relevant the list is to the user for answering a given query. We call this the *structural relevance* of the list. If we consider the ranked list as an affordance for a user to traverse through retrieval results in an orderly manner, then, in XML retrieval, the structural relevance is the expected number of relevant elements found using a weakly ordered list.

SIGIR 2007 Workshop on Focused Retrieval
July 27, 2007, Amsterdam, The Netherlands

In this paper, we present the definition of structural relevance, show how it is incorporated into the evaluation of XML retrieval systems, and provide experimental results to compare SR to XCG. We restricted our experiments here to comparing measures for three runs for top-10 results across many INEX topics for three reasons. The first reason was that we noticed during the development of structural relevance that our results were sometimes different from XCG for certain topics, so we restricted the runs to systems that have performed well over the years at INEX. We restricted our presented results here to a small k (in our case, k=10) over a large number of topics so that we could clearly show through simple examples the differences between what the XCG and SR measures capture. Thirdly, we restricted the runs to the thorough task in INEX because it allows overlap. In future work, we intend to extend the application of SR to all retrieval tasks in INEX. The next two sections of the paper introduce SR and describe a (summary-based) approximation technique for computing SR. Section 4 surveys and compares existing measures and Section 5 presents the experimental results. Conclusions and future work are discussed in Section 6.

2. STRUCTURAL RELEVANCE

In this section we derive the measure of structural relevance based on isolation and overlap, then we show how structural relevance is used to modify measures such as precision and precall, and, finally, we derive a general expression for calculating isolation for weakly ordered ranked lists.

Structural relevance is a measure of the relevance of a group of overlapped elements. The problem of overlap occurs in a ranked list when a user finds some ranked element while browsing a different ranked element. This will occur because the element is reachable either directly from the ranked list or indirectly via structural paths from a different element in the list. We define structural relevance as the expected number of relevant elements that are found by the user while browsing a ranked list of elements for the first time. If there is no overlap in a group of elements, then structural relevance reduces to the number of relevant elements in the set. The following theorem shows how to calculate structural relevance.

THEOREM 2.1. *Measure of Structural Relevance*
The expected number of relevant elements up to some given element u in the ordered set of elements R where $p(e;R)$ denotes the probability of the user first encountering element e from the ranked list R is

$$E[n_R(u)] = \sum_{e \in R[u]} rel(e) \cdot p(e; R[u]) \quad (1)$$

PROOF. The number of relevant elements of a ranked list R can be written as $n_R = \sum_{e \in R} rel(e)$, where $rel(e)$ is the binary relevance of element e to some given topic such that, if e is relevant then $rel(e) = 1$, and $rel(e) = 0$ otherwise. Binary relevance is used here for simplicity and there is no reason that other ways of measuring relevance could not be used.

For recall-precision calculations, the number of relevant elements are calculated at given ranks. Consider element $u \in R$ where $R[u]$ is the ordered subset (the ranking) of elements from R that contains elements up to an element u from the weakly ordered list R. We can rewrite n_R as a function of the element u in list R as,

$$n_R(u) = \sum_{e \in R[u]} rel(e)$$

Assume that system evaluation is conducted in multiple, independent trials for each returned element. So, in top-k search there will be k trials to determine the relevance of all returned elements. Each trial is done in order to determine the relevance of the element e in the ranked list R. The trials are conducted in descending order by rank of elements. Given that k trials are conducted for k ranked elements, it is certain that all elements will have known relevance (*i.e.*, total probability for any element having known relevance after evaluation is 1). Consider the element e, its relevance is determined by either first encountering it from the ranked list, or by first encountering it from an higher or tie ranked element. For element e we get the probabilistic relationship of evaluation,

$$\begin{aligned} 1 = {} & P(\text{encounter } e \text{ first from ranked list}) + \\ & P(\text{encounter } e \text{ first from an overlapped element }) \end{aligned}$$

Let $p(e;R)$ denote the probability of encountering e first from the ranked list. Let $q(e;R)$ denote the probability of encountering e first from an overlapped element so, $1 = p(e;R) + q(e;R)$. We refer to $p(e;R)$ as the *isolation* of e in the ranked list R. We refer to $q(e;R)$ as the *overlap* of e in the ranked list R. To find the relevance up to some element $u \in R$, we take the expectation of relevance on the isolation of elements in $n_R(u)$, and thus we get our desired result,

$$E[n_R(u)] = \sum_{e \in R[u]} rel(e) \cdot p(e; R[u])$$

□

2.1 Precision and Precall with SR

The measure of structural relevance (SR) can be substituted into an evaluation measure for the number of relevant elements in a ranked list. SR makes the measure sensitive to structural overlap among relevant retrieval elements, assuming a user model of ranked traversal and weak ordering. For instance, in precision, $precision = n_R/k$, we substitute $E[n_R]$ from equation 1 for the number of relevant elements to get SR in precision as $SRP = E[n_R]/k$.

Consider now precall, which is the expected precision [18] of weakly ordered ranked elements, where tied elements are assessed in a random order. Based on the expected search length [4], precall estimates the length of the ranked list that would contain a desired number of relevant elements, s, in terms of the expected number of irrelevant elements in the list. Precall calculates precision using the ratio of relevant elements r to irrelevant elements irr in the list to find the expected number of irrelevant elements $\frac{s \cdot irr}{r}$ to achieve the information need of the desired number of relevant elements s, so $precall = n_R/(n_R + [s \cdot irr/r])$.

For SR in precall, we replace n_R with $E[n_R]$ for the number of relevant elements, but *not* the number of relevant elements r used in the *esl* calculation. Substituting for r would

invalidate the *esl* calculation because the search length is derived with the assumption of atomic elements. Substituting SR into precall, we get the SR in precall (SRPL) as

$$SRPL = \frac{E[n_R]}{E[n_R] + \frac{s \cdot irr}{r}} \quad (2)$$

2.2 Isolation of Elements

We now consider some ranked list R with a user model of browsing R as presented in precall [18]. In SR, the user model defines the set of traversal permutations of the ranked list. We use Raghavan's precall model here, but as will be seen, other user models could be similarly incorporated into structural relevance.

Let m_R denote the number of ranks in the ranked list R. Let R_i denote the set of elements in rank i. A user browses a ranked list by visiting elements in descending order from the highest to lowest ranks. For elements with tied scores, they are weakly ordered and the user visits these elements in random order until all elements in the rank have been visited. Let Ω denote the set of traversal permutations derived from the user model of browsing R. Let ℓ be the number of permutations of traversal of the ranked list R such that $\ell = |\Omega|$. The number of permutations in Ω can be calculated in terms of the permutations of orderings across all ranks as,

$$\ell = |\Omega| = \prod_{i=1}^{m_R} |R_i|! \quad (3)$$

The isolation of e is $p(e; R)$ for some given ranked list. By conditioning isolation on a permutation $R' \in \Omega$ of R, we get $p(e; R) = \sum_{R' \in \Omega} [p(R') \cdot p(e; R|R')]$.

For the user to choose a particular traversal path $R' \in \Omega$ where every element in R is visited only once, assume a uniform distribution, so $p(R') = 1/|\Omega| = 1/\ell$. Thus, our conditioned expression for $p(e; R)$ becomes

$$p(e; R) = \frac{1}{\ell} \cdot \sum_{R' \in \Omega} p(e; R|R') \quad (4)$$

Now, let us denote $p(e; R|R')$ as simply $p(e; R')$. The difference between R and R' is that R allows weak ordering, but the elements in R' are strictly ordered. The probability of reaching e from elements in R' can be considered a *Bernoulli* process. The process works as follows; every attempt to browse to e fails until e is reached in R', at which point e is reached with perfect certainty. Let $P(e; f)$ be the probability that e is encountered while browsing starting at element f. The trivial cases are $P(e; e) = 1$, and $P(e; f) = 0$ when e is not accessible from f. So, we calculate $p(e; R')$ for a given R' as follows, where $R'[e]$ is the set of elements in descending rank in R' up to e,

$$\begin{aligned} p(e; R') &= \left[\prod_{f \in (R'[e]-e)} 1 - P(e; f) \right] \cdot P(e; e) \\ &= \prod_{f \in (R'[e]-e)} 1 - P(e; f) \end{aligned} \quad (5)$$

EXAMPLE 2.2. *Isolation in a strictly ordered list.* What is the probability of isolating element e in a strictly ordered list R' of 4 elements where the probability of encountering element e from any other element is 0.8? *Ans.* Using equation 5 where $P(e; f) = 0.8$ we get $p(e; R') = (1 - P(e; f))^3 = 0.2^3 = 0.008$.

We now have a complete expression for the isolation of element e in ranked list R by substituting equation 5 into equation 4 and replacing the probability of not visiting element e from element f with $1 - P(e; f)$. So, we get

$$p(e; R) = \frac{1}{\ell} \cdot \sum_{R' \in \Omega} \left[\prod_{f \in (R'[e]-e)} 1 - P(e; f) \right] \quad (6)$$

where $R'[e] - e$ refers to the ranked list $R'[e]$ minus the element e, ℓ is the number of permutations of orderings (equation 3), and $1 - P(e; f)$ is the probability of not reaching e while browsing the element f.

EXAMPLE 2.3. *Isolation in a weakly ordered list.* What is the probability of isolating element e in a weakly ordered list $R = [a \mid b\ e]$ of 3 elements where the probability of encountering element e from a is 0.8 and from b is 0.4 ?

Ans. Referring to equation 6, there are 2 possible routes to e, either $a \rightarrow e$ or $b \rightarrow e$ with probabilities 0.8 and 0.4, respectively. So, $\Omega = \{[a, e, b], [a, b, e]\}$, $\ell = 2$, and we are given that $P(e; a) = 0.8$ and $P(e; b) = 0.4$. Applying equation 6 we get $p(e; R) = \frac{1}{2} \cdot [(1 - 0.8) + (1 - 0.8) \cdot (1 - 0.4)] = 0.16$.

3. APPROXIMATING ISOLATION

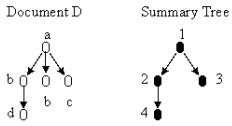
In this section, we introduce the integration of XML summaries with structural relevance to quantitatively model the process of browsing among elements. We show next how isolation in a ranked list can be extended to multiple exclusive sets of overlapped elements, and then using these results we derive an approximation for isolation for calculating SR.

3.1 Incoming XML Summary

Incoming XML summaries are graphs that describe the structure of incoming paths in an XML collection. Summary graphs are formed using XPath queries to generate bisimulations of the elements. The nodes of the summary graph are assigned *labels* that correspond to the tag paths from the root tag to each child tag in a corpus. The *extent* of a node is the set of *elements* in the corpus that match the node's *label*. The size of the *extent* is the number of times that the *label* matches a tag path of an *element* in the corpus. For convenience, each node is assigned a unique structural identifier (SID). There are many types of XML summaries and but we restrict ourselves here to using incoming path summaries. In future work, we intend to extend SR to use other summaries.

The formal definition of the XML summary (also known as XML synopsis, see [17, 3]) is shown below.

DEFINITION 3.1. *A graph synopsis for $G = (V_G, E_G)$ is a node-labeled, directed graph $S(G) = (V_S, E_S)$, where each node $v \in V_S$ corresponds to a set extent$(v) \subseteq V_G$ such that: (1) All elements in extent(v) have the same label (denoted by label(v), i.e., the label of the summary node); (2) $\cup_{v \in V_S}$extent$(v) = V_G$ and extent$(u) \cap$ extent$(v) = \oslash$ for each $u, v \in V_S$; (3) $(u, v) \in E_S$ if and only if there exists $u' \in$ extent(u) and $v' \in$ extent(v) such that $(u', v') \in$*



Incoming Summary

SID	Label	Path	Extent S.
1	a	a	1
2	b	a,b	2
3	c	a,c	1
4	d	a,b,d	1

Figure 1: Example summary tree, document and incoming summary

E_G; and, (4) Each node $v \in V_S$ stores only an element $count(v) = |extent(v)|$.

EXAMPLE 3.2. Consider the summary tree S, the collection consisting of a single document and the incoming summary shown in Figure 1. The figure shows how each root-to-child tag path in the document defines a partition with an *extent* in the summary S. The *extent* of *summary nodes* result in frequency histograms that *describe* the occurence of tag paths in the collection.

If we assume that the summary graph edges are bi-directional and equally weighted in both directions, then we can consider the graph as describing a time-reversible discrete Markovian process [19]. So, given a well-formed XML document from a summarized collection, we can describe the Markovian process of browsing between summary partitions in the document based on the size of the extents of the summary. This allows us to estimate the relative time spent browsing in any partition, which is used later in Section 3.3 to calculate the isolation of elements in the summary partition.

The probability of a user being in some summary partition i while browsing the collection shall be denoted as π_i, and we calculate it by using the steady-state probabilities of the time-reversible discrete Markovian process,

$$\pi_i = \frac{\sum_j w_{ij}}{\sum_i \sum_j w_{ij}} \tag{7}$$

where $i, j \in S$ are partitions of the summary, and w_{ij} is the size of the extent of the child node among the partitions i and j. We interpret π_i as the fraction of time that a user who uses a description of the document structure (*i.e.* a summary) to browse will spend π_i of their time in partition i of the document.

EXAMPLE 3.3. Consider the summary shown in Figure 1. Table 1 shows the weighting matrix and the probabilities π_i of the time-reversible Markov chain for the summary in the figure.

	SID j					
SID i	1	2	3	4	TOTAL	π_i
1	0	2	1	1	4	44%
2	2	0	0	1	3	33%
3	1	0	0	0	1	11%
4	0	1	0	0	1	11%
TOTAL	3	3	1	2	9	

Table 1: Probabilities of browsing a given summary partition using equation 7 for Figure 1.

3.2 Multiple Sets of Overlap

In this section, we generalize SR to lists with multiple exclusive sets of overlapped elements in a ranked list. A set of overlapped elements refers to a weakly ordered subset of overlapped elements in a list whose order is based on the overlapped elements' ranks.

EXAMPLE 3.4. For the ranked list $\|e_1\|e_{11}\ e_2\ e_{21}\|e_{22}\|e_{12}\|$, where e_1 overlaps with e_{11} and e_{12}; and e_2 overlaps with e_{21} and e_{22}; there are two sets of overlapped elements: namely, $\|e_1\|e_{11}\|e_{12}\|$; and, $\|e_2\ e_{21}\|e_{22}\|$.

Consider the elements that are higher ranked in R to some element $e \in R$. Let m_R denote the rank of element e. We know that all of the higher ranked elements to rank m_R will be visited prior to e. Referring to equation 4 in section 2.2, each trial in the strictly ordered list R' to reach e from higher ranked elements must fail. So, the contribution to isolation from the higher ranks is a constant factor for all strictly ordered lists $R' \in \Omega$. Now, consider equation 5, the isolation in a strictly ordered list R',

$$\begin{aligned} p(e;R') &= \prod_{f \in R' \lceil e \rceil} 1 - P(e;f) \prod_{f \in (R' \lfloor e \rfloor - e)} 1 - P(e;f) \\ &= p_{hi}(e;R) \cdot \left[\prod_{f \in (R' \lfloor e \rfloor - e)} 1 - P(e;f) \right] \end{aligned}$$

where $R' \lfloor e \rfloor$ refers to elements equally ranked to e in R', $R' \lceil e \rceil$ refers to elements higher ranked to e in R', and we introduce the higher ranked isolation factor p_{hi}, where we replace R' with R because the higher-ranked elements to e in R' will be equal for all possible cases of R'. We calculate p_{hi} for e using failed trials from the higher ranks,

$$p_{hi}(e;R) = \prod_{f \in R \lceil e \rceil} 1 - P(e;f)$$

For any element f in R that is not overlapped with e, we know that $P(e;f) = 0$, so these elements can be removed from evaluation. So, in the evaluation of structural relevance, for any given element e, we only need to consider the elements overlapped with the element of interest. For multiple clusters of overlap, for each element of interest, we need only consider its higher or tie ranked elements in the list.

The set of higher-ranked elements to an element $e \in R$ is $R \lceil e \rceil = \{f \mid e, f \in R\ \forall f\ \exists\ G[f] > G[e]\}$ where $G[\cdot]$ is the score of an element. Let function $ov(X, x)$ denote the set of overlapped elements to some element x in some list X. So, we combine overlap with higher ranked elements to get the set of overlapped, higher ranked elements to element e,

$$R \lceil e \rceil_{ov} = ov(R \lceil e \rceil, e) \tag{8}$$

The set of tied elements for element e is $R \lfloor e \rfloor = \{f \mid e, f \in R\ \forall f\ \exists\ G[f] = G[e]\}$. So, the set of overlapped, tie ranked elements of element e, including element e, is

$$R \lfloor e \rfloor_{ov} = ov(R \lfloor e \rfloor, e) \bigcup e \tag{9}$$

For SR for some element e, we only consider the overlapped subset $R_{ov}(e)$ of the ranked list R which is found by taking the union of equations 8 and 9,

$$R_{ov}(e) = R\lceil e\rceil_{ov} \bigcup R\lfloor e\rfloor_{ov} \quad (10)$$

Thus, isolation of an element in a ranked list is strictly dependent on its higher ranked and tie ranked overlapped elements in the list.

3.3 Isolation Revisited

In section 2, we introduced structural relevance, showed how it is calculated using isolation, and then showed in equation 6 how isolation is dependent on the probability $P(e;f)$ of encountering a specific element while browsing a different given element. In section 3.1, we presented the XML summary as a Markovian process of browsing the summarized collection in terms of how the *browser* (*i.e.*, user who is browsing the collection) transitions between the summary partitions. Now, we revisit isolation and using results from section 3.2 we present a complete expression for calculating SR.

Assume that browsing from element f to e requires entering the summary partition of e (*i.e.*, being outside of the partition of e) and, simulateously, browsing along the structural paths in the document instance of e and f. So, $P(e;f)$ is the probability of being outside of element e's partition and the probability that structural paths are followed to reach e. Using equation 7, let us denote the probability of being in the partition of element e as $\pi_{(e)}$. So, being outside of the partition is $1-\pi_{(e)}$. Let X denote the probability of following a set of structural paths from element f to element e. The limiting probability of reaching e from f over an infinite number of trials is 1 if we assume that the number of structural paths are finite and that the browser has a positive probability of taking any structural path whenever possible. So, we get,

$$P(e;f) \quad = \quad (1-\pi_{(e)})\cdot X \approx 1-\pi_{(e)} \quad (11)$$

So, referring to equation 6, substituting equation 11 into the isolation of element e in ranked list R, we get

$$\begin{aligned} p(e;R) &= \frac{1}{\ell}\cdot\sum_{R'\in\Omega}\left[\prod_{f\in(R'[e]-e)} 1-P(e;f)\right] \\ &= \frac{1}{\ell}\cdot\sum_{R'\in\Omega}\left[\prod_{f\in(R'[e]-e)} \pi_{(e)}\right] \end{aligned} \quad (12)$$

Recall equation 3 where ℓ is defined across all the elements in the ranked list. Let $R_{(e)}$ be the elements in the rank of element e. To get all cases in ℓ where e is fixed, we would simply reduce the size of the rank $R_{(e)}$ by one, and we get $\ell(e) = (1/|R_{(e)}|)\cdot(\prod_{i=1..m_R}|R_i|!)$, where the number of ranks in R is m_R.

As we noted in equation 8, for some element e in R we need only consider the overlapped elements in $R'[e]$. So, for every ranked list $R'[e]$ there will be $\ell(e)$ number of possible traversals with e at a given rank. Among the tie-ranked elements, there will be $|R_{ov}\lfloor e\rfloor|$ relative positions in which e may occur. So, we substitute $|R_{ov}\lfloor e\rfloor|$ and $\ell(e)$ into isolation equation 12, and then rearrange, to get

$$\begin{aligned} p(e;R) &= \frac{\ell(e)}{\ell}\sum_{n=1}^{|R_{ov}\lfloor e\rfloor|}\pi_{(e)}^{n+m-1} \\ &= \frac{1}{|R_{(e)}|}\sum_{n=1}^{|R_{ov}\lfloor e\rfloor|}\pi_{(e)}^{n+m-1} \end{aligned} \quad (13)$$

where $R_{(e)}$ are the set of elements in the rank of element e, $R_{ov}\lfloor e\rfloor$ is the set of overlapped elements in the rank of element e, m is the number of higher ranked, overlapped elements to e, and $1-\pi_{(e)}$ is the approximated probability of browsing to element e from any overlapped element f.

Now, referring to the modified precall and precision metrics in section 2.1, consider structural relevance in equation 1 and substitute in the approximated isolation for $p(e;R[u])$ from equation 13 to get our final expression for SR

$$SR[u] \quad = \quad \sum_{e\in R[u]}\frac{rel(e)}{|R[u]_{(e)}|}\sum_{n=1}^{|R[u]_{ov}\lfloor e\rfloor|}\pi_{(e)}^{n+m-1} \quad (14)$$

4. XML RETRIEVAL METRICS

In this section we briefly present and discuss three XML retrieval measures. XCG is presented first (a detailed experimental comparison with SR is deferred to the following section), followed by PRUM and HiXEval.

4.1 Extended Cumulated Gain

Extended cumulated gain (XCG) is a cumulated gain (CG) [8] measure that addresses structural dependencies in the recall-base, such as near-misses and overlap, in content-oriented XML retrieval evaluation [10]. It is a flexible measure that incorporates multi-criteria assessments and modeling of user satisfaction. The relevance assessment of elements is used to determine the *ideal recall-base*, which contains elements with the highest scores without overlapping [9], and the *full recall-base*, which contains all relevant elements. Using two recall-bases, ideal and full, requires dependency normalisation heuristics to ensure that the total score for any element does not exceed the maximum score achievable when the ideal node itself is retrieved [10].

The scores of elements are determined using a relevance value function. The score is indicative of the utility of the element to a user. The relevance value function uses quantizations of relevance assessments to return a score in $[0,1]$. The function takes into account overlap, ranking of elements and provides a weighting factor α to represent the user's intolerance to overlapped elements.

The cumulated gain and ideal gain are calculated by summing the scores of ranked elements up to some prescribed position. Denote the score of the a-th element in a list of k_R length as $xG[a], a\in[1,k_R]$. Furthermore, denote the score of the a-th element in an ideal list of k_I length as $xI[a], a\in[1,k_I]$. So, up to a given position a,we express the cumulated score for a list (xCG) and the cumulated score for an ideal list (xCI) as follows,

$$xCG[a] = \sum_{i=1}^{a} xG[i],\ xCI[a] = \sum_{i=1}^{a} xI[i]$$

There are a number of different ways to compare the cumulated scores for a list and its ideal. In this work, we con-

sidered only the normalized extended CG ($nxCG$), which is simply the ratio of the cumulated scores for the list and its ideal, such that

$$nxCG[a] = \frac{xCG[a]}{xCI[a]} \quad (15)$$

4.2 Precision Recall with User Modeling

Precision recall with user modeling (PRUM) is one of the newest proposals for XML retrieval evaluation metrics. It measures the percentage of ideal elements in a collection that are seen by a user while browsing a ranked list [16, 15]. Like XCG, it relies on the definition of an ideal recall-base which consists of relevant elements that do not overlap. In addition, PRUM incorporates multi-criteria assessments and modeling of user satisfaction based on structural constraints. Unlike XCG, PRUM is a probabilistic measure and is defined as the expected number of ideal elements that are seen by the user, up to a given rank, while she browses the collection from the ranked list [1].

SR and PRUM use similar probabilistic event models and user models. But they differ in how relevance is considered. In SR, there is a distinction between relevance of an element and the relevance of a set of elements. SR does not explicitly include the use of multi-criteria assessments. SR employs summaries to model user satisfaction, whereas PRUM uses browsing habits derived from the assessment process. PRUM is at an early stage of development, and real world results to compare with SR are not available at the present time.

4.3 Highlighting XML Retrieval Evaluation

Highlighting XML retrieval evaluation (HiXEval) [14] is another recently proposed approach to measure the effectiveness of XML retrieval systems. HiXEval was motivated by the need to simplify XML evaluation and make it conform to well-established evaluation measures such as precision and recall. HiXEval was proposed as an extension of the traditional definitions of precision and recall to include the knowledge obtained from the highlighting assessment procedure adopted at INEX 2005. The biggest difference between HiXEval and other measures is its contention that the purpose of the XML retrieval task is to find elements that contain as much relevant information as possible, without also containing a significant amount of non-relevant information. With the way relevance has been assessed since 2005, this translates to the aim of returning elements that contain as much highlighted (relevant) content as possible, and as little non-highlighted (non-relevant) content as possible.

In calculating precision and recall, the explicit structure of the documents is ignored because these measures are based upon the amount of highlighted text in and across elements and documents. We leave a comparison of SR with HiXEval for future work.

5. SR AND XCG APPLIED TO INEX

The following investigations were conducted on a single run from 3 different systems for top-10 results for the thorough task in 114 INEX Wikipedia topics [13]. At this stage of our work, the focussed task was not considered because it does not allow overlapping elements, and thus SR modified measures (such as precision or precall) would have the same value as unmodified measures [1]. The results obtained are illustrative of the differences between SR and XCG. The incoming summary for calculating isolation was generated from the Wikipedia collection using the methodology presented in section 3.1. For XCG, we used the normalized extended cumulated gain ($nxCG$) with the configuration genLifted, overlap on, and $\alpha = 1$. The experimental measures were structural relevance with precall (SRPL) and precision (SRP) as described in section 2.1.

Table 2: System outputs for topic 295

Notation for Systems

x/-	:	relevant/irrelevant element
‖	:	rank boundary
o[p/c/s]	:	overlap with ancestor/descendant/sibling

IBMHAIFA : ‖**xocp**‖**xoc**‖**xop**‖**xoc**‖**xop**‖**xops**‖**xoc**‖**-ops**‖**xop**‖**xoc**‖

172477.xml	**1136198.xml**	**14724.xml**
8: /article[1]/body[1]	0: /article[1]/body[1]	2: /article[1]/body[1]
9: /article[1]	1: /article[1]	3: /article[1]
76266.xml	5: /article[1]/body[1]/section[2]	
4: /article[1]/body[1]	7: /article[1]/body[1]/section[5]	
6: /article[1]		

$k = 10, r = 9, ranks = 10, relevant\ docs = 4$

LIP6 : ‖**-op**‖**-oc**‖**-oc**‖**-op**‖**xoc**‖**xop**‖**-oc**‖**-op**‖**-op**‖**-oc**|

3130820.xml	**196073.xml**	**1331267.xml**
7: /article[1]/body[1]	0: /article[1]/body[1]	6: /article[1]
9: /article[1]	1: /article[1]	8: /article[1]/body[1]
14724.xml	**1773624.xml**	
4: /article[1]	2: /article[1]	
5: /article[1]/body[1]	3: /article[1]/body[1]	

$k = 10, r = 2, ranks = 10, relevant\ docs = 1$

MAXPLANCK : ‖**-op -**‖**x**‖**-oc**‖**x**‖**-**‖**-**‖**x**‖**-**‖**-**‖

1773624.xml	**1331267.xml**	**172477.xml**
3: /article[1]	6: /article[1]	7: /article[1]
0: /article[1]/body[1]/section[1]	**2251312.xml**	**1711143.xml**
23273.xml	9: /article[1]	8: /article[1]
5: /article[1]/body[1]/section[9]		**14724.xml**
63285.xml	**419136.xml**	4: /article[1]
2:/article[1]/body[1]/section[4]	1: /article[1]/body[1]/section[6]	

$k = 10, r = 3, ranks = 9, relevant\ docs = 3$

Table 3 shows the graphical results for topics 295, 307, and 335 on the three columns on the left (discussed later on). The right-most column shows the system rankings for each measure in terms of the number of topics that resulted in a given system rank order. The ranking was done for each measure and for each topic by ordering the systems in descending order based on the area of each system's performance curve. The histograms are labeled according to the ranking of systems from left (best) to right (worst): (M)AXPLANCK, (I)BMHAIFA, and (L)IP6. For instance, the first column of the top-most histogram shows that for SRP there were 60 topics where the systems were ranked MIL, or in other words, were ranked in descending order of performance: MAXPLANCK, IBMHAIFA, LIP6. The histograms for SRP and SRPL are sub-divided to show the number of topics for which the measure in question did not obtain the same ranking relative to rankings in XCG. Returning to the SRP histogram, for example, the first column in the histogram shows that SRP ranked the systems as MLI

[1]We will investigate in future work the use of SR as a meanss to measure relevance in the recall-base itself.

for 60 topics and 50 of those topics were also ranked as MLI by XCG.

Overall, SRP agreed with XCG for 78 out of 114 topics or 68%. SRPL agreed with XCG for 38 out of 114 topics or 33%. Since XCG, SRP and SRPL produce different results, we turn our attention to a small representative subset of topics to provide some insight into the differences.

5.1 Individual Topic Comparison

The detailed results for topic 295, including ranked list, overlap of elements, label paths, and documents, are shown in table 2. The notation used for representing overlapped elements is at the top of table 2. The first column of table 3 shows the evaluations for SRP, SRPL, and XCG for topic 295. Referring to table 2, a user who either randomly explores a list or systematically explores from the head of a list to the end would find the results of IBMHAIFA the best for topic 295. It contains more relevant documents and relevant elements occurring at earlier ranks than either LIP6 or MAXPLANCK. This observation is reflected in both SRP (table 3, column 1, row 1) and SRPL (table 3, column 1, row 2). XCG (table 3, column 1, row 3) differs in that it concludes that MAXPLANCK and IBMHAIFA are the best and nearly equal.

Topic 295 is a good example of how overlap affects the relevance of a list. This depends on the composition of the ranked list in terms of elements, ranking, and overlap. For instance, SRP for topic 295 shows a steep decline in the output of IBMHAIFA at 20% recall because of overlapped elements. We see SRP fluctuate across recall levels, upward with novel elements and downward with overlapped elements. At recall, it settles around 40% precision. But, we can see that the SRP is significantly above 40% in early ranks, and as the overlap becomes more pronounced at late recall levels the SR precision eventually achieves the precision based on 4 documents out of 10 elements. We see this behavior because documents are independent sets of elements and this is reflected in SR-based precision. SRPL (table 3, column 1, row 2), on the other hand, does not exhibit this behavior because precall is based on the expected search length which is strongly determined by the number of irrelevant elements. SR does not account for irrelevant elements, and we can see that overlap degrades SRPL for IBMHAIFA only slightly (*i.e.*, 9 out 10 elements in Table 2 are relevant with SRPL at about 80% for recall 1).

We recognize two general cases that we believe explain the differences between SRP and XCG. The first case involves over-penalization where results contain parent elements consistently ranked higher than children elements; XCG seems to over-penalize these configurations for overlap. The second case involves early recall, where results containing relevant elements at early ranks and results containing relevant elements at late ranks will perform overall equally in XCG. In contrast, results containing relevant elements at early recall score higher in SRP.

Case 1: Overlap Penalization. Topic 307 is a good example of overlap penalization. Over-penalization occurs because of the dependency normalisation heuristic that differentiates between the order of parent and child elements in a ranked list. The heuristic is that if a parent element is *seen*, then its child elements are considered *fully seen*, whereas if a child element is *seen* then its parent is only *partially seen* [10]. This heuristic results in over-penalization in all configurations of XCG except where $\alpha = 0$. Referring to Table 3, IBMHAIFA contains the best results because it returns both more relevant documents, and more relevant elements. But because parent elements are being ranked higher than child elements, XCG ranks IBMHAIFA last in performance. This example also demonstrates the inverse of this phenomena, where LIP6 is rewarded for overlap in its first two ranks.

Table 4: System outputs for topic 307

$IBMHAIFA$: ‖**-op**‖**-oc**‖**xop**‖**xop**‖**xoc**‖**xoc**‖**xop**‖**x**‖**xoc**‖**x**‖
$k = 10, r = 8, ranks = 10, docs = 4, relevant\ docs = 3$
$LIP6$: ‖**xoc**‖**xop**‖**-oc**‖**-op**‖**-oc**‖**-op**‖**xoc**‖**xop**‖**-oc**‖**-op**‖
$k = 10, r = 4, ranks = 10, docs = 5, relevant\ docs = 2$
$MAXPLANCK$: ‖**x -**‖**-oc**‖**x**‖**-**‖**-**‖**-**‖**x**‖**-op**‖**-**‖
$k = 10, r = 3, ranks = 10, docs = 9, relevant\ docs = 3$

Table 5: System outputs for topic 335

$IBMHAIFA$: ‖**xop**‖**xoc**‖**-ops**‖**xocps**‖**-ocps**‖**xops**‖**-ops**‖**-ops**‖**xops**‖**xocps**‖
$k = 10, r = 6, ranks = 10, docs = 1, relevant\ docs = 1$
$LIP6$: ‖**-ocs** ‖**-ops** ‖**-ocs** ‖**-ops** ‖**xocs** ‖**xops** ‖**xocs** ‖**xops** ‖**-ops** ‖**-ocs** ‖
$k = 10, r = 4, ranks = 10, docs = 5, relevant\ docs = 2$
$MAXPLANCK$: ‖**- -** ‖**-** ‖**- x** ‖**-** ‖**-** ‖**-** ‖**xocs** ‖**-ops** ‖
$k = 10, r = 2, ranks = 8, docs = 9, relevant\ docs = 2$

Case 2: Early Recall. Topic 335 is a good example of how XCG is not sensitive to early recall in a ranked list. XCG evaluates MAXPLANCK as the best search engine for topic 335 (see table 3, column 2, row 3). Referring to table 5, this makes some sense because MAXPLANCK returns the most number of relevant documents with the least overlap. But, MAXPLANCK is not the best list because the relevant elements in MAXPLANCK occur at the end of the list. In this regard, we would posit that IBMHAIFA has a better result. This can be see in topic 335 for IBMHAIFA that SRP is highest in early recall, although performance degrades significantly in later recall.

6. CONCLUSIONS AND FUTURE WORK

We have presented a general model of structural relevance and shown how it can be used to modify precall and precision for measuring effectiveness in XML retrieval. The SR approach uses XML summaries to represent how users perceive overlap in XML retrieval. The experimental results presented suggest that SR handles situations such as over-penalization of overlap due to heuristics and sensitivity to results with early recall more effectively than XCG. More significantly, we show that SR does not require an ideal recall-base or dependency normalization, as is the case for existing measures.

Future work includes obtaining results on the performance of SR for a larger number of systems, carrying out additional comparisons of SR (with XCG, PRUM, and HiXEval), undertaking reliability tests for the SR metric, and further developing the application of summary-based techniques to SR measures.

7. REFERENCES

[1] D. Carmel, Y. S. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer. Searching xml documents via xml fragments. In *SIGIR '03: Proc. of the 26th Annual Intl. ACM SIGIR Conf. on Res. and Dev. in IR*, pages 151–158, New York, NY, USA, 2003. ACM Press.

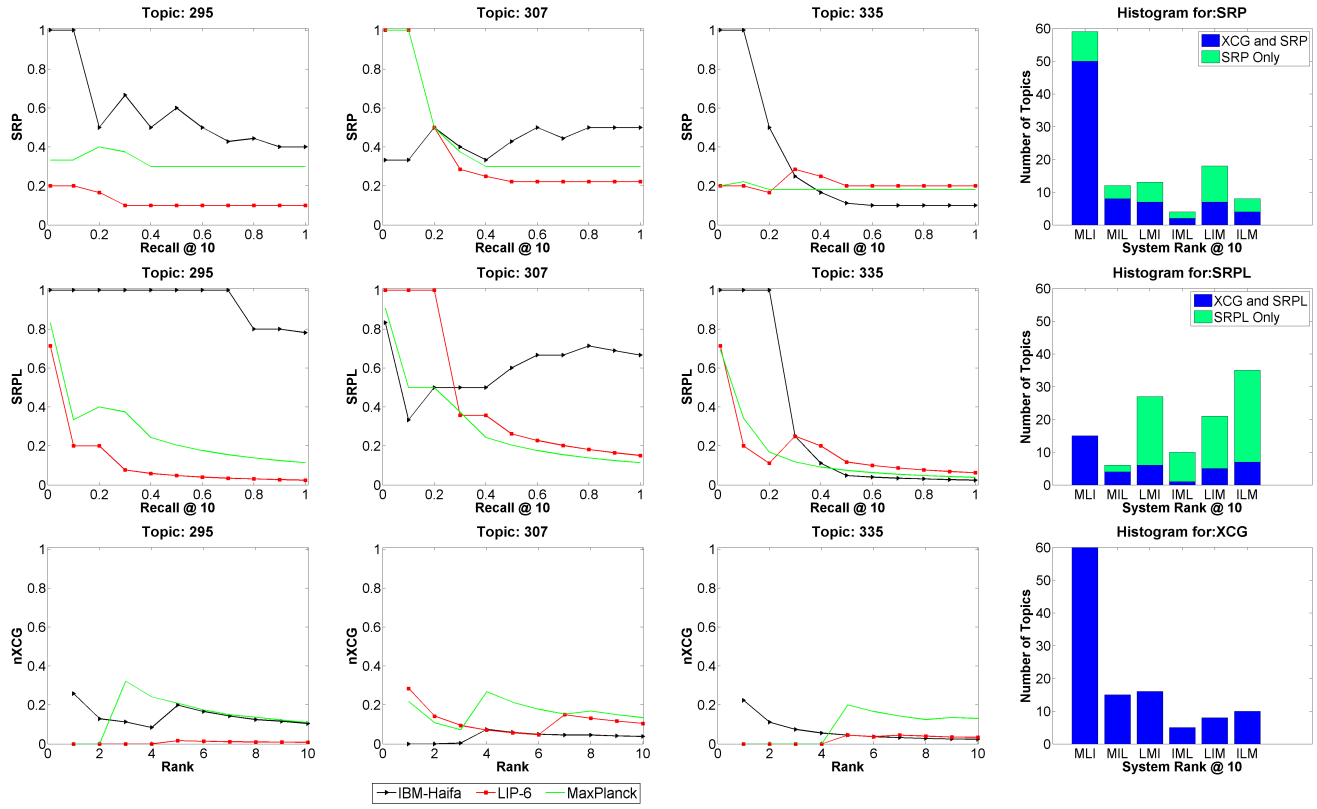


Table 3: System evaluations (SRP, SRPL, and XCG) and histograms across all topics

[2] C. Clarke. Controlling overlap in content-oriented xml retrieval. In *SIGIR '05: Proc. of the 28th Ann. Intl. ACM SIGIR Conf. on Res. and Dev. in IR*, pages 314–321, New York, NY, USA, 2005. ACM Press.

[3] Mariano P. Consens, Flavio Rizzolo, and Alejandro A. Vaisman. Exploring the (semi-)structure of XML web collections. Technical report, UofT - DCS, 2007. http://www.cs.toronto.edu/∼ consens/describex/.

[4] W. S. Cooper. Expected search length: A single measure of retrieval effectiveness based on weak ordering action of retrieval systems. *J of the Amer. Soc. for Info. Science.*, 19:30–41, 1968.

[5] A. de Vries, G. Kazai, and M. Lalmas. Tolerance to irrelevance: A user-effort oriented evaluation of retrieval systems without predefined retrieval unit. In *RIAO 2004*, Vaucluse, France, April 2004.

[6] N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors. *Adv. in XML IR, 3rd Intl. Workshop of INEX 2004*, volume 3493 of *LNCS*. Springer, 2005.

[7] N. Gövert and G. Kazai. Overview of the initiative for the evaluation of xml retrieval (inex) 2002. *Proceedings of the 1st Workshop of the INitiative for the Evaluation of XML Retrieval*, pages 1–17, 2003.

[8] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[9] G. Kazai. Choosing an ideal recall-base for the evaluation of the focussed task: Sensitivity analysis of the xcg evaluation measures. In *Compar. Eval. of XML IR Sys., 5th Intl. Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2006)*. Springer, 2007.

[10] G. Kazai and M. Lalmas. Extended cumulated gain measures for the evaluation of content-oriented xml retrieval. *ACM Trans. Inf. Syst.*, 24(4):503–542, 2006.

[11] G. Kazai, M. Lalmas, and A. de Vries. The overlap problem in content-oriented xml retrieval evaluation. *Proc. of the 27th Ann Intl ACM SIGIR Conf on Res and Dev in IR*, 2004.

[12] J. Kekäläinen, M. Junkkari, P. Arvola, and T. Aalto. TRIX 2004 - Struggling with the Overlap. In Fuhr et al. [6], pages 127–139.

[13] M. Lalmas and G. Kazai. Report on the ad-hoc track of the INEX 2005 workshop. *SIGIR Forum*, 40(1):49–57, 2006.

[14] J. Pehcevski and J. A. Thom. HiXEval: Highlighting XML retrieval evaluation. In *Adv. in XML Info. Retrieval: 4th Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005)*. Springer-Verlag, November 2005.

[15] B. Piwowarski and G. Dupret. Expected precision-recall with user modelling (eprum). In *SIGIR '06: Proc. 29th Ann. Intl ACM SIGIR Conf on R&D in Info Retr*, pages 260–267, NY, NY, USA, 2006. ACM Press.

[16] B. Piwowarski, P. Gallinari, and G. Dupret. Precision recall with user modeling (prum): Application to structured information retrieval. *ACM Trans. Inf. Syst.*, 25(1):1, 2007.

[17] N. Polyzotis and M. N. Garofalakis. Statistical synopses for graph-structured XML databases. In *SIGMOD Conf.*, 2002.

[18] V. Raghavan, P. Bollmann, and G. Jung. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans. Inf. Syst.*, 7(3):205–229, 1989.

[19] S. M. Ross. *Introduction to Probability Models*. Academic Press, New York, 8th edition, 2003.

[20] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. Mixture Models, Overlap, and Structural Hints in XML Element Retrieval. In Fuhr et al. [6], pages 196–210.

[21] A. Tombros, S. Malik, and B. Larsen. Report on the INEX 2004 interactive track. *ACM SIGIR Forum*, 39(1):43–49, 2005.