Providing Consistent and Exhaustive Relevance Assessments for XML Retrieval Evaluation

Benjamin Piwowarski[*]
LIP6, University Paris 6
8, rue du capitaine Scott
75015 Paris, France
bpiwowar@poleia.lip6.fr

Mounia Lalmas
Department of Computer Science Queen Mary
University of London
England, E1 4NS
mounia@dcs.qmul.ac.uk

ABSTRACT

Comparing retrieval approaches requires test collections, which consist of documents, queries and relevance assessments. Obtaining consistent and exhaustive relevance assessments is crucial for the appropriate comparison of retrieval approaches. Whereas the evaluation methodology for flat text retrieval approaches is well established, the evaluation of XML retrieval approaches is a research issue. This is because XML documents are composed of nested components that cannot be considered independent in terms of relevance. This paper describes the methodology adopted in INEX (the INitiative for the Evaluation of XML Retrieval) to ensure consistent and exhaustive relevance assessments.

Categories and Subject Descriptors

H.3.7 [**Information Systems**]: Information Storage and Retrieval—*Digital Libraries*; H.5.3 [**Information Systems**]: Information Interfaces and Presentation—*Group and Organisation Interfaces*

General Terms

Measurement, Standardisation

Keywords

XML, evaluation, relevance assessment process, INEX

1. INTRODUCTION

The increasing use of the eXtensible Markup Language (XML) in scientific data repositories, digital libraries and on the web, brought about an explosion in the development of XML retrieval systems to store and access XML content [1, 2, 3, 9]. The aim of such retrieval systems is to exploit the explicitl logical structure of documents, and retrieve components instead of whole documents, in response to a user query. Implementing this retrieval paradigm means that an XML retrieval system needs not only to find relevant information in the XML documents, but also to determine the appropriate level of granularity to return to the user.

A consequence of this retrieval paradigm is that the relevance of an element is depends both on content and structural conditions. Evaluating the effectiveness of XML retrieval systems, hence, requires a test collection where the relevance assessments are provided according to a relevance criterion that takes into account the imposed structural aspects. A test collection as such has been developed by INEX[1], the INitiative for the Evaluation of XML Retrieval [5]. The initiative, now in its third year, aims to establish an infrastructure and provide means, in the form of a large XML test collection and appropriate scoring methods, for the evaluation of content-oriented retrieval of XML documents.

In information retrieval (IR) research, when following a system-centred evaluation viewpoint, effectiveness provides a measure of a system's ability to retrieve as many relevant and as few non-relevant documents to a given query as possible. Such an evaluation criterion relies on appropriate measures of relevance. Traditional IR, however, mainly deals with flat text files. An important difference between flat text retrieval and XML retrieval is that, in the latter, the relevance of XML elements cannot be considered independently of each other since XML elements can be nested within each other, as they exhibit a parent-child relationship. When constructing a test collection for evaluating XML retrieval effectiveness, we must consider this dependence in order to obtain exhaustive and consistent relevance assessments.

For example, the fact that a child element is deemed relevant to a query implies that its parent element is also relevant to that same query, eventually to a different extent. We need to assess the relevance of both elements, the child element and its parent element, to ensure that the relevance assessments are exhaustive. When an element has been assessed as non-relevant to a particular query, then none of its children elements can be relevant to that same query. This condition ensures that the relevance assessments are consistent. In this paper, we describe the methodology adopted in INEX to ensure exhaustive and consistent relevance assessments.

The paper is organised as follows. In Section 2, we introduce the bi-dimensional relevance scale used to assess XML elements in INEX. Based on this definition of relevance, we describe, in Section 3, the so-called E-Rules and C-Rules that were proposed and some of which used to ensure as much as possible exhaustive and consistent relevance assessments. In Section 4, we describe the methodology used to obtain the relevance assessments. This in-

[*] Currently at DCC - Universidad de Chile, Santiago, bpiwowar@dcc.uchile.cl

[1] `http://inex.is.informatik.uni-duisburg.de:2004/`

CIKM'04, November 8–13, 2004, Washington, DC, USA.

cludes the selection process of the XML elements to be assessed for relevance, and the development of an online interface, in which the rules were implemented. In Section 5, we present an analysis of the effect of the E-Rules and C-Rules in ensuring exhaustive and consistent assessments.

2. RELEVANCE IN INEX

The INEX document collection is so far made up of the XML marked-up full-texts of 12,107 articles of varying length of the IEEE Computer Society's publications from 12 magazines and 6 transactions, covering the period of 1995-2002. On average an article contains 1,532 XML elements, where the average depth of an element is 6.9.

The XML elements forming a document (an article in the context of INEX) can be nested. Some elements will be large (e.g. sections) and others small (e.g. paragraphs). Since retrieved elements can be at any level of granularity, an element (the larger element) and one of its children (the smaller element) can both be relevant to a given query, but the child is more focussed to the query than its parent. In this case, the child is a better element to retrieve than its parent, because not only it is relevant to the query, but it is also specific to the query.

The above relates to earlier work on hypermedia document retrieval [4], which showed that the relevance of a structured document can be better described by two logical implications. The first one, $d \rightarrow q$ (the document *implies* the query), is the *exhaustivity* of document d for the query q, and models the extent to which the document discusses all the aspects of the query. The second one, $q \rightarrow d$ (the query *implies* the document), is the *specificity* of the document d for the query q, and models to what extent all the aspects of the document concern the query. Therefore a document can be exhaustive but not specific to a query, and vice versa. Put in the context of XML retrieval, some XML elements will be exhaustive but not specific to a given query, as they will be too large; whereas other elements will be specific to a query, but not exhaustive, as they will be too small.

Based on the above, INEX adopted two dimensions to describe the relevance of an XML element:

- Exhaustivity (e-value) measures the extent to which the given element covers or discusses the query.
- Specificity (s-value) measures the extent to which the given element is focussed on the query.

In many IR evaluation frameworks, the relevance value of a document is restricted to 0 (not relevant) or 1 (relevant). Such a scale is not suited for XML retrieval because of the nested nature of the XML elements forming a document. For example, in the case of articles, a section composed of many paragraphs, where only one paragraph is relevant - whether with respect to specificity or exhaustivity - to a query is also relevant to that same query, *but to a lesser extent*. We therefore require relevance values between 0 and 1 to reflect the relative relevance of an element with respect to related elements. INEX therefore adopted a four-graded scale based on the one proposed in [8] for the two dimensions of relevance. With respect to exhaustivity:

- Not exhaustive (0): the XML element does not discuss the query at all.
- Marginally exhaustive (1): the XML element discusses only few aspects of the query.
- Fairly exhaustive (2): the XML element discusses many aspects of the query.
- Highly exhaustive (3): the XML element discusses most or all aspects of the query.

With respect to specificity:

- Not specific (0): the query is not a theme of the XML element.
- Marginally specific (1): the query is a minor theme of the XML element.
- Fairly specific (2): the query is a major theme of the XML element.
- Highly specific (3): the query is the only theme of the XML element.

The combination of the two dimensions is used to identify those relevant XML elements, which are both exhaustive and specific to the topic of request (the query) and hence represent, *according to INEX, the most appropriate unit of information to return as an answer to the query*.

The exhaustivity and specificity values are not independent from each other. A non-exhaustive element (0 e-value) is also not specific (has a 0 s-value), and vice versa, thus restricting to 10 the combination of e-value and s-value. In the remaining of the paper, an assessment will be denoted $EeSs$ where e and s are integers between 0 and 3, and E stands for e-value and S for s-value. For example, $E2S3$ corresponds to "fairly exhaustive and highly specific". An element will be considered *relevant* if $e > 0$ and $s > 0$. An element is *not relevant* if its assessment is $E0S0$. We will denote "?" an unknown value for the exhaustivity or the specificity dimension.

In INEX 2002, another but comparable definition of relevance was used, also based on two dimensions. The first dimension corresponds to the exhaustivity dimension defined in INEX 2003[2]. The second dimension, the coverage, is related to specificity. It has four values: no coverage (not specific), too small, too big (fairly or marginally specific) and exact (highly specific). The value that cannot be related to a value of specificity is "too small", which meaning was "exhaustive but too small to act as a meaningful unit". We refer to the INEX 2002 definition when discussing the analysis of the assessments in Section 5.

3. EXHAUSTIVITY AND CONSISTENCY

To compare the effectiveness of retrieval approaches, we require test collections where the relevance assessments are as accurate as possible. In the context of XML retrieval, this means that we must ensure that the relevance assessments are as exhaustive[3] and consistent as possible. This is a complex and tedious task because the relevance of an XML element cannot be assessed independently of that of other elements.

To ensure that relevance assessments are as exhaustive and consistent as possible, INEX developed a number of rules. These are described in Sections 3.1 and 3.2, respectively. In the remainder of this paper, we denote x a given XML element, w its parent element and $y_1, ..., y_n$ its (n) children elements. The exhaustivity value and specificity value of an XML element z are denoted E_z and S_z, respectively.

[2] In INEX 2002, this dimension was called "relevance", but this term was misleading and was therefore replaced.

[3] The exhaustivity dimension shall not be confused with the exhaustivity of the assessments. The former refers to the assessment scale whereas the latter is related to the exhaustiveness of the relevance assessments with respect to a query.

3.1 Exhaustivity of the assessments

The INEX test collection contains 8.2 million elements, each being a possible result for each of the 126 queries forming the testbed[4]. It is therefore not feasible to assess the relevance of all the elements forming the collection in order to obtain exhaustive assessments. INEX follows the pooling method [11] (Section 4.1) similar to TREC [12]. This method selects among the retrieved results submitted by the participants' retrieval systems, those that will be submitted for assessment.

In the context of XML retrieval, it may (and will often) happen that an element is submitted for assessment but neither its parent element nor any of its children elements (because they have not been retrieved by any of the systems). These (related) elements will therefore be left unassessed. If the retrieved (submitted) element is assessed as $S2$, then at least one of its descendants is more specific ($S3$). If we leave these descendants unassessed, a retrieval system that retrieves the unassessed highly specific ($S3$) element will be ranked below a system that retrieved the assessed $S2$ element. Evaluating XML retrieval approaches using this data cannot lead to reliable and fair comparisons. The relevance of some of the elements related to those forming the pool needs to be assessed.

The approach adopted by INEX is as follows. A pool, which consists of XML elements, is created for each query (see detail in Section 4.1). Each element of the pool must be assessed for its relevance to the query. Depending on the relevance value of the element, additional elements are added to the pool so that they can be assessed. The process ends when all elements in the pool have been assessed.

Not all related elements need to be assessed, so related element should not be automatically added to the pool. Determining which one should be or not is important to keep the assessment task feasible. In INEX 2003, the addition of elements in the pool is based on three rules, discussed next, that select which elements should be added to the pool.

E-RULE 1 (NOT RELEVANT). *When an element has been assessed as not relevant ($E0S0$), no element is added to the pool.*

This rule enables assessors to quickly judge documents that are not relevant. As long as no relevant information is found (in an article), no additional element is added to the pool.

E-RULE 2 (HIGHLY SPECIFIC). *When an element has been assessed as highly specific ($S3$), only its ancestor elements[5] are added to the pool.*

Adding the children elements of a highly specific element is not necessary as we have already reached a highly specific element, and retrieving anything smaller does not make sense. However, adding its ancestors can lead to the discovery of new highly specific elements which can be one of the ancestors (which should also be more exhaustive) or one of its siblings.

E-RULE 3 (OTHER CASES). *When an element has been assessed as marginally or fairly specific ($S1$ or $S2$), its children elements and its ancestor elements are added to the pool.*

The last rule forces assessors to identify more specific elements ($S3$), if any (we recall that the retrieval task in INEX is to return as an answer to a query the most exhaustive and specific elements for that query, with preference to specificity). The rule can however add too many elements. Consider the case of a highly exhaustive element that has many non-relevant siblings. The parent element will then be assessed as marginally or fairly specific as it contains irrelevant material. According to E-Rule 3, all its children - i.e. the siblings - will then be added to the pool although most of them are not relevant.

In INEX 2004, we are planning to modify E-Rule 3 (the part regarding the addition of children elements) so that children elements are added only if there is some relevance "missing":

E-RULE 4 (GENERAL). *When the element is relevant (its relevance value is everything except $E0S0$), add its children elements only if:*

$$\sum_{y_i} E_{y_i} < E_x$$

where the summation is over all assessed children elements.

This rule is illustrated with an example. If an assessor judges an element as $E2$, this means that (1) at least one of its children must be $E2$ (i.e. it contains all the relevant information of the parent), or (2) at least two of its children must be $E1$ (i.e. they contain all the relevant information of the parent when merged). Until situation (1) or (2) is reached, sibling elements are added. When either of them is reached, unassessed siblings are removed from the list of elements to assess.

The rules presented in this Section aim at providing exhaustive relevance assessments. These rules did not exist in INEX 2002, but proved to be useful in INEX 2003 (see Section 5.2). These rules might evolve as they are a compromise to be reached between two contradictory objectives: (1) finding the most specific and exhaustive elements and (2) what the assessors want (having as few elements to assess as possible).

3.2 Consistency

In classical IR, documents are assumed independent from each other with respect to their relevance to a given query. This is not the case with XML documents. For example, if a section composed of paragraphs only is relevant to a query, then *at least one* of its paragraphs must contain some relevant material to that same query. It is therefore necessary to have consistency rules that restrict relevance values for some elements. In this section, we describe the rules used in INEX, which were based on our definition and (intuitive) interpretation of exhaustivity and specificity. We present the rules in chronological order: INEX 2002, INEX 2003 and INEX 2004 (the latter are currently being debated).

3.2.1 INEX 2002 rules

In INEX 2002, one consistency rule was used. This rule concerns the "amount of relevant information" allowed between an element and its parent element.

C-RULE 1 (EXHAUSTIVITY GROWTH). *An XML element cannot be more exhaustive than its parent element:*

$$\forall i, E_w \geq E_x \geq E_{y_i}$$

This ensures that an element cannot have less relevant information than any of its children elements. Exhaustivity values can only increase from children to parent elements.

A study performed on the INEX 2002 relevance assessments showed that inconsistent assessments were indeed made [7], and some of them could have been avoided by the enforcement of consistency rules. Some of them were implemented in INEX 2003.

[4] This Figure includes the INEX 2002 and INEX 2003 queries. The INEX 2004 queries are currently being selected.

[5] Its parent, and the parent of its parent, etc.

3.2.2 INEX 2003 rules

INEX 2003 used an interface (see Section 4.2) in the relevance assessment task, which checked the consistency of the assessments while the assessments were being made. Two new rules were used, which were discussed and agreed upon at the INEX 2002 workshop. The first rule, which relates to the exhaustivity dimension, states that an element that has no relevant children cannot be relevant.

C-RULE 2 (NON-RELEVANCE OF CHILDREN).

$$\forall i, E_{y_i} = 0 \implies E_x = 0$$

Originally, y_i referred to XML elements, as delimited by XML tags. This interpretation led to problems, for example, in the case of a paragraph element that contains some of its text in italics. In this case, the paragraph was considered to contain one child, the "italic" element[6]. If this element is not relevant, then according to C-Rule 2, the paragraph element is then not exhaustive ($E0$), although its other text, i.e. not in italics, may be relevant. To solve this, text nodes associated with an XML element also constitute children of this element although they cannot be assessed. So for example, a paragraph that has some normal text (text-node) first, then some text in italics, then some normal text again, is considered to have three children elements.

The second rule is concerned with the specificity dimension, and states that the amount of relevant information in an XML element cannot be greater than the amount of relevant information in all its children elements, for a given query.

C-RULE 3 (MAXIMUM SPECIFICITY).

$$S_x \leq \max_i \left(S_{y_i}\right)$$

For example, if a section has two children, which are composed of, respectively, 30% and 50% of relevant information, the section cannot be composed of more than 50% of relevant information. In other words, specificity cannot increase when going from (all) children elements to parent elements.

3.2.3 INEX 2004 rules

In the third round of INEX, we are considering new rules, which were discussed at the INEX 2003 workshop. The first rule is a generalisation of C-Rule 2 and is related to E-Rule 4. With this new rule, we want to express that relevant information cannot be lost between a parent element and its children elements. That is, all the information present in the parent element must be present in some of, eventually distributed among, its children elements. For example, if a parent with two children has been assessed as highly exhaustive ($E3$), it does not seem correct to have the two children marginally relevant ($E1$), or one of them not relevant ($E0$) and the other fairly relevant ($E2$). In both cases, we have lost "one degree" of exhaustivity. This is expressed by the following rule:

C-RULE 4 (EXHAUSTIVITY MASS).

$$\sum_i E_{y_i} \geq E_x$$

The above rule is indeed more general than C-Rule 2 because if all children have 0 e-value, then the parent will also have 0 e-value.

The next rule is symmetric to C-Rule 3. The motivation for this rule is that, for instance, if a parent element has two children elements containing respectively 30% and 50% of relevant information, the parent element cannot contain less than 30% of relevant information.

More generally, the amount of relevant information in an element cannot be less than the amount of relevant information in any of its children.

C-RULE 5 (MINIMUM SPECIFICITY).

$$S_x \geq \min_i \left(S_{y_i}\right)$$

The last rule is still under debate, and is not based on the exhaustivity and specificity values of elements only. This rule is related to the metrics used to evaluate the effectiveness of XML retrieval systems, where returning overlapping elements (e.g. a section and all its paragraphs) is to be discouraged [6].

If two overlapping elements are both highly specific (s-value of 3) and have the same exhaustivity value, they cannot be distinguished with respect to which one is a better answer for a given query. In INEX 2004, we are proposing to disallow this case, which is expressed by the enforcement of the following rule:

C-RULE 6 (HIGHLY SPECIFIC UNIQUENESS).

$$E_x > 1 \wedge S_x = 3 \implies \forall i, E_x > E_{y_i}$$

When an element is highly specific, it must be more exhaustive than any of its children. We added the restriction that the element must also be at least fairly exhaustive ($E2$ or $E3$); otherwise the rule would imply that a marginally exhaustive element ($E1$) will have all its children not relevant ($E0$), which is not allowed. This is due to the restricted number of values (4) in the exhaustivity scale.

As opposed to E-Rules, C-Rules are mainly based on the definition of exhaustivity and specificity. These rules enable us to check the consistency of the assessments and, as described in the next section, to prevent assessors from introducing inconsistent assessments. It should also be noted that most of all the rules do not depend on the exact values of the relevance scale, so are still valid if a different and more refined (e.g. continuous) scale is used.

4. OBTAINING THE RELEVANCE ASSESSMENTS

In this section, we describe how the relevance assessments were obtained in INEX. In Section 4.1, we describe how the evaluation pools were constructed for each query with the aim of starting with a good basis for ensuring exhaustive assessments. In Section 4.2, we describe the online interface that was used to perform the relevance assessment task, and in particular the implementation of the consistency rules.

4.1 Pooling submissions

If we merge all the participants' submissions, there is an average of 16,000 unique elements and 5,300 unique documents per query for INEX 2003. The average number of submitted lists per query is 51.4 (participants were allowed 3 submission runs per query, and around 25 participants submitted runs). It is clearly not possible to ask each participant to assess so many different documents and their elements (within 24 hours, an assessor can judge around 500 documents). As stated in Section 3.1, INEX follows the pooling method to select, for each query, the elements to be assessed for

[6] There is of course the discussion whether italic elements are meaningful retrieval units. This is still an issue in INEX.

their relevance. In this section, we discuss how the elements forming the initial pools are selected, and provide some insight on how appropriate this selection was to ensure a good basis for obtaining exhaustive assessments. This is important because, as described in Section 3.1, elements that are added to the pool for assessment are related to those originally in the pool.

In INEX 2002, the first 100 elements from each participant submission were merged for each query. In INEX 2003, the unique retrieved elements of 500 articles from all the participant submissions were combined in a round robin fashion to form the pools. For the i^{th} round (or iteration), the i^{th} element of each participant submission was added to the pool (unless it was already in the pool). The process was repeated until the pool contained at least 500 documents (i.e. articles). A document is said to be "in" the pool when at least one of its element is in the pool. We will call such documents "pool documents".

It is also possible to stop the process when the pool contains a certain number of elements (independently on the number of pool documents). The decision to stop the process based on the number of pool documents came from the hypothesis that assessing elements from the same documents would be less consuming than assessing elements from different documents. As assessors in INEX are asked to read (or skim) the article before assessing any element, this hypothesis seems valid.

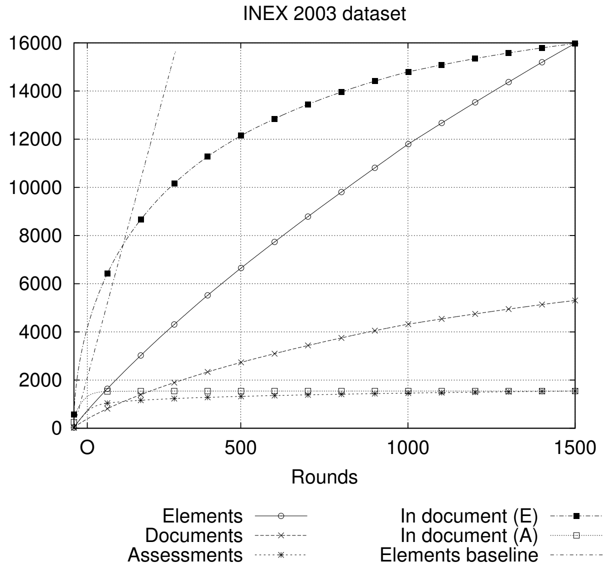


Figure 1: Pooling. The X-axis is the number of iteration, the Y-axis represent the number of elements, documents or assessments in the pools. The values are averaged over the number of pools. The "O" marks the state of the official INEX 2003 pools (500 documents).

To investigate whether stopping at 500 pool documents was appropriate (in terms of providing a good basis for the original pool), we looked at the content of the pools at 1 to 1,500 iterations for the INEX 2003 submissions (since each submission run contained 1,500 elements). That is, we did not stop the iteration - adding elements to the pool - when 500 pool documents were obtained, but when all the 1,500 elements of each submission have been added to the pool. We then computed the following values, which were averaged over all the pools (one pool per query), and plotted them in Figure 1:

Elements Number of XML elements in the pool after the i^{th} iteration;

Documents Number of documents in the pool after the i^{th} iteration;

Assessments Number of XML elements in the pool after the i^{th} iteration that were assessed. Note that this number is the same as the number of elements up to a certain number of iterations because these are elements that are in the official pool (before "O" in Figure 1).

In document (E) Number of XML elements that were in one submission and in one of the pool documents after the i^{th} iteration. At the 1,500th iteration, this number is equal to the number of elements in the pool since all submitted elements are then in the pool.

In document (A) Number of XML elements of "document (E)" which were assessed. The difference (E)-(A) at "O" is the number of elements that were *not assessed*, but that were in at least one submission and in a pool document. In general, the difference (E)-(A) represent the number of elements we have to add to the pool if we want that all elements in submissions that are within a pool document to be assessed.

The number of elements in the pool grows almost linearly with the number of iterations, but there are many common elements between submitted runs - which is to be expected. This can be observed by looking at the "Elements baseline" curve, which gives the average number of elements that should be in the pools if the elements were unique.

The number of documents grows far less quickly. This means that as the number of iterations increases, an increasing numbers of elements are already in one of the pool documents. Another interesting observation comes from looking at the "In document (E)" curve. We can see that 50% of elements in the participants' submissions are in the first 20% documents that are added in the pool. This clearly indicates that the top ranked elements returned by the systems tend to be in the same documents.

This therefore shows that stopping the pooling process based on a desired number of documents is a valid approach for constructing the original pool.

It is possible, although with additional overhead, to assess *all* the submitted elements within the pool documents. This would involve adding an average 3,900 elements to each pool if we consider the difference between the "In document (E)" and "Elements" curves. If we look at the difference between the "In document (E)" and "In document (A)" curves, the actual number of element that would have to be added to the assessed ones is only 3,425. This means that an average of 463 assessed elements in INEX 2003 were not in the pool but in the submitted lists. We could then increase the exhaustivity of the assessments by adding more elements from the submitted lists - those that are part of the pool documents.

Furthermore, since assessing elements in the same document is not as time consuming as assessing the same amount of elements from different documents, the above would not add too much further assessment time for the assessors. This is illustrated in Figure 2 where the user effort is plotted. After each pooling round, we measure both the total amount of time needed by the user to assess every element (in hours) and the time he or she spend to assess a single element (in second). This number were computed with respect to two different pooling algorithms:

1. is the above pooling algorithm,

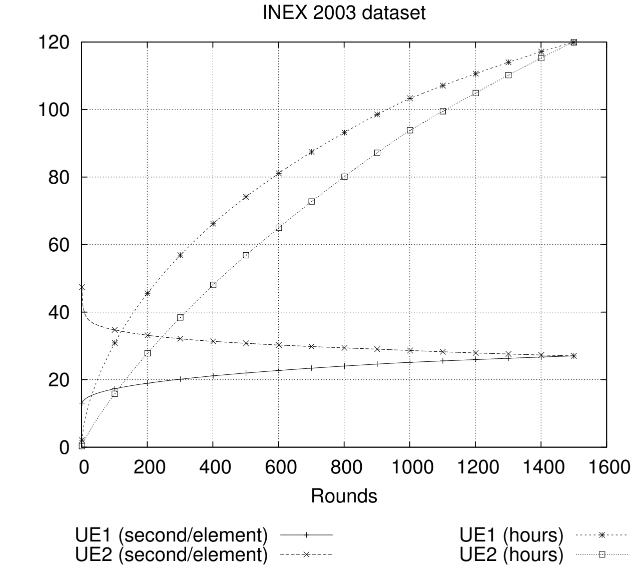


Figure 2: Pooling and user effort. The X-axis is the number of iterations, the Y-axis represents either the number of hours or number of seconds. we denote ”UE” the user effort (measured in seconds). The number after ”UE” denotes the pooling algorithm used: (2) is the classical pooling algorithm, (1) is a pooling algorithm for which, after convergence, every element which is both in the submitted lists and in a document that is already in the pool is added to the pool.

2. is the classical pooling algorithm

Note that in the figure, the number of documents in the pool remains the same for both algorithms. To compute these values, we used the values of 14 seconds per assessed element and of 47 seconds per new document to assess. These values were calculated on the INEX 2003 data using least square error minimisation. At “O” rounds, the user effort is 16 seconds per element for algorithm (1) against 36 seconds per element for algorithm (2). This indicates clearly that more elements from submitted lists could be assessed without increasing substantially the user effort.

Another improvement of the pooling algorithm would be to take into account the structure when we merge submissions. For example, two paragraphs from the same section could be replaced by only one element, the enclosing section. This would reduce the number of assessments while ensuring we have some relevance information (although not complete) on the paragraphs.

4.2 The interface

In this section, we describe the interface that was developed for INEX 2003 to perform the relevance assessment task. The aim of the interface was to ensure consistent and exhaustive relevance assessments, but also to ease the assessment process. This is crucial because in INEX, the assessors are the INEX participants, who are not paid to perform the task, but had to do it in addition to their daily activity. We first describe how assessors can interact with the interface (Section 4.2.1), then we discuss how the consistency rules are implemented in the interface (Section 4.2.2).

4.2.1 Interaction

The main interface part is the article view (Figure 3). The article view has two parts. The upper and main part provides a view of the document where the XML tags (in light grey) and the assessments associated with their corresponding elements are shown. When an assessor clicks on an XML tag, a panel pops up. This panel provides various information regarding the element to assess (1) a (unique) reference to that element using XPath notation[7]; (2) the current relevance judgement of that element in words; and (3) a number of possible relevance values - shown in squares - that can be assigned to that element. There is a maximum of 10 relevance values, plus the unknown value “?”. The latter means that the element remains to be assessed, and can also be used to erase a given relevance value (e.g. the assessor wants to re-assess the element at a later stage). The relevance values not permitted by the consistency rules appear in grey. When the assessor clicks to one of the allowed values (its corresponding square), the panel closes. The assessment is stored, and exhaustivity and consistency rules are then applied to eventually add new elements to assess, and to restrict relevance values of related elements, respectively.

It is possible to add assessments *one by one* as described above. In INEX 2003, new judging modes were added as they were requested by assessors: it was possible to judge a *group* of user-selected elements and to assess an element and all its *sibling*s.

At the bottom of the interface (the second part of the interface), a bar shows information regarding the actual status of the article being assessed in terms of number of elements with a given relevance value, etc, also helping the assessor to view how much more work is needed to complete the assessment of the article and the pool itself. Additional information includes means to help assessors navigating through the elements to assess, articles and pool (more details can be found in [10]).

4.2.2 Implementing consistency rules

To apply the rules described in Section 3.2, we need to know precisely the e- and s-values given by the assessor. However, until an article has been fully assessed, the relevance values of many of its elements will be unknown. These unknown values have to be considered when preventing inconsistent assessments. The aim here is to use the available evidence in an intelligent way to help assessors while performing their assessment task. We already discussed the consistency rules in Section 3.2. In this section, we describe how these rules are implemented with the above purpose.

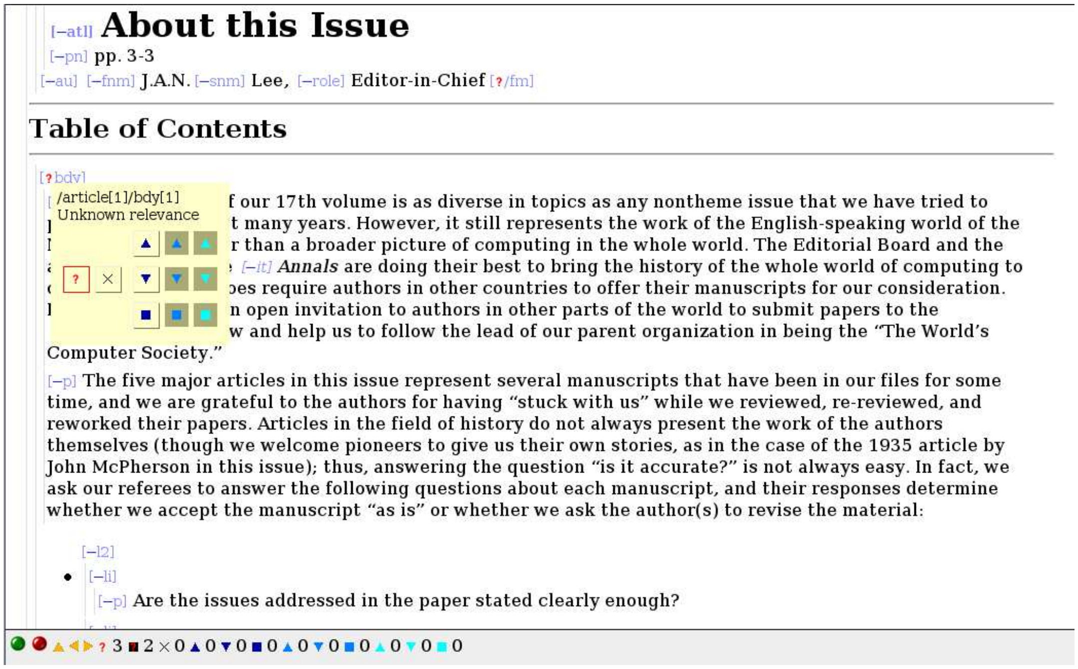
At the start of the assessment process, each element x within the article has a maximum s-value $s_x^{\max}$ (respectively e-value $e_x^{\max}$) of 3 and a minimum s-value $s_x^{\min}$ (respectively e-value $e_x^{\min}$) of 0.

We use rules to update these maximal and minimum values, which we call the boundary values to continually check the consistency of the relevance value of an element. If the s-value and e-value of the element are outside the boundary, an inconsistency is detected. If the boundary values are equal to the same value (i.e. maximum value = minimal value), then there is only one relevance value allowed for the element, which is then inferred as the relevance value of that element. In all other cases, the element is considered as *consistent* until further evidence.

To sum up, for any element x, the maximum and minimum s-values ($S_x^{\max}$ and $S_x^{\min}$) are either the s-value given by the assessor (S_x) or the boundary values ($s_x^{\max}$ and $s_x^{\min}$) if no assessment is known for x. The same applies for the exhaustivity boundary values.

To apply rules that were originally developed for precise e- and s-values, we have to transform them so that they can be applied with interval values (the boundary values). Let us consider C-Rule 1 as

[7] An XPath expression allows to uniquely identify all the XML elements of the documents forming the collection. See `http://www.w3.org/TR/xpath`.



[–atl] About this Issue

[–pn] pp. 3-3

[–au] [–fnm] J.A.N. [–snm] Lee, [–role] Editor-in-Chief [?/fm]

Table of Contents

[?bdy]

/article[1]/bdy[1]
Unknown relevance

f our 17th volume is as diverse in topics as any nontheme issue that we have tried to
t many years. However, it still represents the work of the English-speaking world of the
r than a broader picture of computing in the whole world. The Editorial Board and the
[–it] *Annals* are doing their best to bring the history of the whole world of computing to
oes require authors in other countries to offer their manuscripts for our consideration.
n open invitation to authors in other parts of the world to submit papers to the
w and help us to follow the lead of our parent organization in being the "The World's
Computer Society."

[–p] The five major articles in this issue represent several manuscripts that have been in our files for some time, and we are grateful to the authors for having "stuck with us" while we reviewed, re-reviewed, and reworked their papers. Articles in the field of history do not always present the work of the authors themselves (though we welcome pioneers to give us their own stories, as in the case of the 1935 article by John McPherson in this issue); thus, answering the question "is it accurate?" is not always easy. In fact, we ask our referees to answer the following questions about each manuscript, and their responses determine whether we accept the manuscript "as is" or whether we ask the author(s) to revise the material:

[–l2]

- [–li]
  [–p] Are the issues addressed in the paper stated clearly enough?

Figure 3: Main assessment window for INEX 2003: the user is assessing the body of a document (`/article[1]/bdy[1]`). The assessment panel (below "Table of contents") has three components: the path (first line), the current assessment (second line), and the set of 11 icons (reflecting all possible assessments). Forbidden assessments (e.g. assessing a parent element as not relevant where one of its child elements is relevant) are displayed in a grey box. The current assessment is unknown (?), and only $E0S0$ and $E1S?$ are allowed.

an example. If we know that for a element x and its parent w, which are both unassessed,

$$\left(e_w^{\min}, e_w^{\max}\right) = (0,2)$$

and

$$\left(e_x^{\min}, e_x^{\max}\right) = (1,3)$$

then we can still use the rule which states that $E_w \geq E_x$. That is, we can update $e_w^{\min}$ to 1 as E_w cannot be less than E_x. We can also update $e_x^{\max}$ to 2 as E_x cannot be greater than E_w.

C-Rule 1 can then be easily transformed: $e_x^{\min}$ must be greater or equal than each $E_{y_i}^{\min}$ and $e_x^{\max}$ must be less or equal then $E_w^{\max}$. If this is not the case, $e_x^{\min}$ or $e_x^{\max}$ is updated.

C-Rules 3, 5 and 4 must be transformed with care. For instance, let us consider C-Rule 3. It is easy to show that $s_x^{\max}$ must be less or equal than $max_i(S_{y_i}^{\max})$. It is however possible to update the boundary values of one of the children using another consequence of the same rule, which states that if for all children but one (y_j), the inequality $S_x > S_{y_i}$ holds, then $S_x \leq S_{y_i}$. To take into account the boundary values we can state that if for all $i \neq j$, $S_x^{\min} > S_{y_i}^{\max}$, then $s_{y_i}^{\min}$ must be greater or equal than $S_x^{\min}$.

The other consistency rules can be transformed in a similar fashion. Computing the boundary values is then an iterative process, which ends when values cannot be refined anymore, for all the elements of the article.

5. ANALYSIS

In this section, we provide an analysis of the effect of the interface, of E-Rules and C-Rules. Firstly, we analyse the assessor behaviour in general. Then, we provide some insight on exhaustivity and consistency of assessments; data from INEX 2002 and 2003 are compared in order to fully understand what the E-Rules and C-Rules changed. Finally, we present some data on agreement to conclude this section.

5.1 Sessions

In this section, we describe how the interface was used by the assessors. More specifically, we were interested in the mean duration of a session, in the use of the different modes of assessment (single, group, sibling) and in the behaviour of the assessors within an article (e.g. do they follow a particular path or do they assess "at random"?).

The number of analysed assessments was 203,384[8], which were performed between the 10 September and 25 November 2003. All accesses to the online interface were recorded into a log file. We recorded the exact time when each article and which element in the article was accessed, and its assessment value.

We first determined a session as the set of actions defined as the pair (view of an article, assessment of an element) without interruption. Consecutive actions separated by a maximal time of T_{max} seconds were considered to belong to the same session. We set T_{max} to 18 minutes, which corresponds to the mean time between accessing an article (to view the article) and the assessment of its first element - increased by its standard deviation. Using this value of T_{max}, a session lasted in average 52 minutes, where an average of 111 elements and 20 articles were assessed. Using least square regression technique, we computed that an assessor spent in aver-

[8] We count grouped or sibling assessments as one assessment.

age 47 seconds to assess an article *and then* 11 seconds per element within the document (article).

Regarding how the elements were assessed, in the vast majority of cases, the elements were assessed individually (80.2 %). In 17 % of the cases, the assessments were performed in group, and only in 2.8 %, sibling elements were assessed together. This shows that methods allowing several elements to be assessed together were used. There was no noticeable difference in the relevance values obtained using the different modes of assessments.

The time spent assessing an article strongly depends on the presence of relevant elements in the article. Assessors spent on average 8 minutes (judging on average 28.2 elements) with articles containing at least one relevant element, compared to an average of 1 minute (judging 1.3 elements) with articles containing no relevant elements. This difference in times is also due to the addition of elements to assess to ensure exhaustivity.

After assessing an element, assessors went to assess in 38 % of the cases its sibling elements, in 10% of the cases its parent element, and in 12% of the cases, one of its children elements. On average, in 90% of the cases, the next elements to be assessed were those close to the ones just assessed (average distance of 3 elements). The average behaviour of assessors strongly indicates that elements are judged "near-by-near", and this behaviour is reinforced by the interface that adds new elements to assess when an element has just been assessed.

5.2 Assessments

In this section, we analyse the effect of the 4 E-Rules and 6 C-Rules in ensuring consistent and exhaustive assessments. Our analysis used the INEX 2002 and INEX 2003 data set. Different combinations of rules were used (reflecting their chronological order). These are shown in Table 1. Note that we have two rules set for 2004, namely 2004a and 2004b. The latter includes the C-Rule 6 which is most debated. We had to map the INEX 2002 coverage scale to the INEX 2003 specificity scale: "exact" coverage was mapped onto S3; "too big" coverage was mapped onto S2; "too small" was mapped onto E1S3.

INEX Rules set	2002	2003	2004a	2004b
E-Rules	-	1, 2, 3	1, 2, 4	1, 2, 4
C-Rules	1	1, 2, 3	1, 4, 3, 5	1, 4, 3, 5, 6

Table 1: Summary of rules: we define four different rule sets (named 2002, 2003, 2004a and 2004b). In this table, we show the E/C-Rules used in each set.

5.2.1 Exhaustivity

In this section, we analyse the exhaustivity of the relevance assessments following the use of the E-Rules. Note that no exhaustivity rule was used in INEX 2002. Although assessors were asked to assess as many related elements as possible in an article, the additional assessment was not forced upon them. Firstly, an average of 2,969 elements per pool were assessed in INEX 2003; this number was 1,665 in INEX 2002. In addition, in INEX 2002, 66% of assessed elements were in the original pool; this number was 26% in INEX 2003. This clearly shows that to obtain exhaustive assessments, we must "force" assessors to assess related elements.

Furthermore, 61% of *S*3 (highly specific) elements were not in the original INEX 2003 pools; this number is down to 39% for INEX 2002. The enforcement of E-Rules led to the identification of many more highly specific elements that would not have been found otherwise. Further evidence can be found by looking at Figure 2. With the INEX 2003 E-Rules, 1,050 elements would have been added in the INEX 2002 pools. The new E-Rule 4 reduces this number (an average of 350 elements per pool for the 2004a and 2004b rules).

Inferred relevance values can speed the assessment process. Around 6% of assessments are inferred (see Figure 2). The more C-Rules are used, the higher the number of inferences. By increasing the number of *correct* inferred values, we can ask assessors for further assessments so that to increase the exhaustivity of the relevance assessments.

5.2.2 Consistency

INEX 2002 dataset

Rule set State	2002	2003	2004a	2004b
Consistent	1565	1562	1557	1527
Inferred	101	104	104	104
Inconsistant	22	22	26	57
To assess		1054	348	348

INEX 2003 dataset

Rule set State	2002	2003	2004a	2004b
Consistent	2807	2799	2738	2701
Inferred	162	169	169	169
Inconsistant		1	62	99
To assess		301	271	271

Table 2: Assessed pool states: in these tables, the number of elements are averaged over the assessed pools. In each table, the number of elements that are consistent, inferred, inconsistent or to assess are shown for each E/C-Rules set.

Table 2 shows the number of inconsistent relevance values obtained with the different sets of C-Rules (see Table 1). This number is the greatest for the INEX 2003 data set when using the 2004a and 2004b C-Rules: the average number being 21, 22, 26 and 56 for INEX 2002 and 0, 0, 62 and 99 for INEX 2003 for the 2002, 2003, 2004a and 2004b C-Rules, respectively. This is to be expected as a higher number of elements were assessed on the INEX 2003 data set.

With respect to C-Rules planned for INEX2004, most of the inconsistencies in the INEX 2003 assessments are induced by the C-Rule 6 of INEX2004b (although this new rule does not introduce as much inconsistencies as expected from what was discussed at the INEX 2003 workshop). As this rule ensures that two nested elements cannot be as relevant (highly specific with the same level of exhaustivity), this new rule proved to be useful and applicable.

More generally, both for the INEX 2002 and 2003 data, as the number of rules increases so does the number of inconsistent elements. An interface is therefore crucial for both forbidding and preventing non-permitted relevance values during the relevance task. The former is important to ensure high quality assessments, while the latter is important to facilitate the work of the assessor.

C-Rules	1	2	3	4	5	6
INEX 2002	100	-	-	-	-	-
INEX 2003	98.38	0.38	0.34	-	0.89	-
INEX 2004a	95.68	-	2.75	0.31	1.26	-
INEX 2004b	93.57	-	2.84	0.31	1.11	2.15

Table 3: Rules usage (in %)

We were also interested in the usage of the C-Rules. In Table 3, we calculated the number of times a rule was used to update the e-

or s-boundaries (see Section 4.2.2). E-Rule 1 was by far the most used rule. This is to be expected since this is the only rule that applies to exactly two elements (a parent and its child). All the other rules relate a parent to *all* of its children, and as such may not need to be called so often. The other rules remain nevertheless important.

This can be shown if we restrict these figures to elements that are in the state “inferred” or “inconsistent” after an inference. The usage of the C-Rule 1 then drops down to an average of 88% (inference) and 15% (inconsistent). Other important rules are then C-Rule 4 (inference) and C-Rule 3 and 6 (inconsistency).

5.2.3 Agreement

Another way to look at the exhaustivity and specificity of relevance assessments is by looking at the agreement level between assessors. The INEX 2002 data set contains two pools that were assessed by two assessors, and one pool that was assessed by three assessors. Therefore, we use those pools to compute statistics which are presented below.

Exhaustivity

	0	1	2	3
0	46	71	36	1
1		328	312	18
2			261	142
3				20
%	30	45	35	11

Coverage

	N	S	L	E
N	46	14	90	4
S		27	68	61
L			730	95
E				100
%	30	16	74	38

Table 4: Agreement on exhaustivity (27%) and coverage (37%). The number of assessments for which exhaustivity (left) or the coverage (right) is the same are shown in tables. For each possible exhaustivity or coverage value, the last line give the percentage of agreement.

In XML retrieval, the level of agreement between assessors should be smaller than with flat text retrieval, which is known to be between 40 and 50%. In INEX, as the relevance assessments are not binary but can take up to 10 different values, the agreement level drops down to 22%. When looking at partial agreements (Table 4), that is when judges agree either on exhaustivity or on coverage, this level increases to 27% for exhaustivity and 37% for coverage. If we consider “near matches” for exhaustivity (we consider that judges agree if the difference between exhaustivity is inferior or equal to 1), the agreement is 90%. However, with respect to coverage, agreement remains low, especially when looking at the “exact” coverage (highly specific) elements with 38% of agreement.

The above figures were calculated on a small number of assessments, and should be considered carefully. Finer statistics are needed for example to investigate the level of agreements between assessors when E-Rules and C-Rules are used, as we would expect the C-Rules to have a positive effect on agreement since they restrict the set of relevance values an element can have. A large-scale investigation on these issues are planned for INEX 2004.

6. CONCLUSION

In this paper, we described the methodology adopted in INEX in order to provide consistent and exhaustive relevance assessments for a collection of XML documents. We described the bidimensional scale used for assessment, the rules defined upon this scale, the pooling process and the online interface used by the assessors. We provided some insight into the effect of the rules and the interface in obtaining consistent and exhaustive assessments. However, further investigations and discussions are needed in order to prove the correctness of the rules and to measure the improvement, both in term of exhaustivity and consistency, of the assessments quality. It would also be interesting to measure the differences the rules make with respect to the different metrics used in INEX. As discussed at the INEX 2003 workshop, the interface was found useful by the INEX participants as it eased the assessment process. Many more elements were assessed in INEX 2003 without a significant increase in the assessor effort.

Throughout this article, we also discussed improvements to be performed in the next editions of INEX. First, we introduced a new E-Rule to add less elements to assess. We proposed three new C-Rules. Two of them are simply a consequence of our better understanding of the notion of relevance in the context of XML documents. Our analysis has shown that for each new rule that was introduced after the assessments were done, inconsistent assessments were detected. It is thus important to enforce consistent rules, and as many as possible. Second, we described the pooling process and justified the approach taken in INEX 2003, which was based on the desired number of documents (and not elements). We showed that this process could be refined to increase the exhaustivity of the assessments, without adding too much effort for the assessors.

Acknowledgment

We would like to thank the reviewers for their excellent and helpful comments, and also G. Kazai and G. Dupret for their feedback. The INEX initiative is funded by DELOS, a network of excellence for digital libraries. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication on ly reflects the authors’ views.

7. REFERENCES

[1] R. Baeza-Yates, N. Fuhr, and Y. S. Maarek, editors. *ACM SIGIR 2002 Workshop on XML*, Aug. 2002.

[2] H. M. Blanken, T. Grabs, H.-J. S. andRalf Schenkel, and G. Weikum, editors. *Intelligent Search on XML Data, Applications, Languages, Models, Implementations, and Benchmarks*, volume 2818 of *Lecture Notes in Computer Science*. Springer, 2003.

[3] D. Carmel, Y. Maarek, and A. Soffer, editors. *ACM SIGIR 2000 Workshop on XML*, July 2000.

[4] Y. Chiaramella. Browsing and Querying: two complementary approaches for Multimedia Information Retrieval. In *HIM’97 International Conference*, Dortmund, Germany, 1997.

[5] N. Fuhr, S. Malik, and M. Lalmas. Overview of the initiative for the evaluation of xml retrieval (inex) 2003. In *Proceedings of the Second INEX Workshop*, March 2004.

[6] 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.

[7] G. Kazai, S. Masood, and M. Lalmas. A study of the assessment of relevance for the INEX’02 test collection. In S. McDonald and J. Tait, editors, *Advances in Information Retrieval, 26th European Conference on IR Research, ECIR 2004*, volume 2997 of *Lecture Notes in Computer Science*, Sunderland, UK, Apr. 2004. Springer.

[8] J. Kekäläinen and K. Järvelin. Using graded relevance assessments in IR evaluation. *Journal of the American*

Society for Information Science (JASIS), 53(13):1120–1129, 2002.

[9] R. Luk, H. Leong, T. Dillon, A. Chan, W. B. Croft, and J. Allan. A Survey in Indexing and Searching XML Documents. *JASIS*, 6(53):415–437, Mar. 2002.

[10] B. Piwowarski and M. Lalmas. Interface pour l'évaluation de systèmes de recherche sur des documents XML. In *Premiere COnference en Recherche d'Information et Applications (CORIA'04)*, Toulouse, France, Mar. 2004. Hermès.

[11] K. Sparck Jones and C. J. Van Rijsbergen. Report on the need for and provision of an ideal information retrieval test collection. Technical Report 5266, Computer Laboratory, University of Cambridge, Cambridge, England, 1975.

[12] E. M. Voorhees and D. K. Harman, editors. *The Tenth Text Retrieval Conference (TREC 2001)*, Gaithersburg, MD, USA, 2002. NIST.