INEX 2002 - 2006: Understanding XML Retrieval Evaluation

Mounia Lalmas and Anastasios Tombros

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

Abstract. Evaluating the effectiveness of XML retrieval requires building test collections where the evaluation paradigms are provided according to criteria that take into account structural aspects. The INitiative for the Evaluation of XML retrieval (INEX) was set up in 2002, and aimed to establish an infrastructure and to provide means, in the form of large test collections and appropriate scoring methods, for evaluating the effectiveness of content-oriented XML retrieval. This paper describes the evaluation methodology developed in INEX from 2002 up to 2006. The paper focusses on the ad hoc retrieval track of INEX.

1 Introduction

The continuous growth in XML[1] information repositories has been matched by increasing efforts in the development of XMLretrieval systems (e.g. [2,3]), in large part aiming at supporting content-oriented XML retrieval. These systems exploit the available structural information, as marked up in XML, in documents, in order to implement a more *focussed* retrieval strategy and return document components – the so-called *XML elements* – instead of complete documents in response to a user query. This focussed retrieval approach is of particular benefit for information repositories containing long documents or documents covering a wide variety of topics (e.g. books, user manuals, legal documents), where users' effort to locate relevant content can be reduced by directing them to the most relevant parts of these documents. For example, in response to a user's query on a collection of scientific articles marked-up in XML, an XML retrieval system may return a mixture of paragraph, section, article, etc. elements, that have been estimated as best answers to the user's query. As the number of XML retrieval systems increases, so does the need to evaluate their effectiveness.

The predominant approach to evaluate system retrieval effectiveness is with the use of test collections constructed specifically for that purpose. A test collection usually consists of a set of documents, user requests usually referred to as topics, and relevance assessments which specify the set of "right answers" for the user requests. There have been several large-scale evaluation projects, which resulted in established information retrieval (IR) test collections and evaluation methodologies [21].

Traditional IR test collections and methodology, however, cannot directly be applied to the evaluation of content-oriented XML retrieval as they do not consider structure [7].

[1] http://www.w3.org/XML/

This is because they focus mainly on the evaluation of IR systems that treat documents as independent and well-distinguishable separate units of approximately equal size, or in the same order of magnitude. Since content-oriented XML retrieval allows for document components to be retrieved, multiple elements from the same document can hardly be viewed as independent units. When allowing for the retrieval of arbitrary elements, we must also consider the overlap of elements; e.g. retrieving a complete section consisting of several paragraphs as one element and then a paragraph within the section as a second element. This means that retrieved elements cannot always be regarded as separate units. Finally, the size of the retrieved elements should be considered, especially due to the task definition; e.g. retrieve minimum or maximum units answering the query, retrieve a component from which we can access, or browse to, a maximum number of units answering the query.

In standard document retrieval, given a ranked output list, users look at one document after the other from this list, and then stop at an arbitrary point. Thus, non-linear forms of output are not considered. When multiple elements from the same document are retrieved, a linear ordering of the result items may not be appropriate (i.e. elements from the same document are interspersed with elements of other documents). Single elements typically are not completely independent from their context (i.e. the document they belong to). Thus, frequent context switches would confuse the user in an unnecessary way. It would therefore be more appropriate to cluster together the result elements from the same document.

The evaluation of XML retrieval systems thus makes it necessary to build test collections where the evaluation paradigms are provided according to criteria that take into account the imposed structural aspects. The INitiative for the Evaluation of XML retrieval (INEX)[2], which was set up in 2002, established an infrastructure and provided means, in the form of large test collections and appropriate scoring methods, for evaluating how effective content-oriented XML search systems are. This paper provides a detailed overview of the evaluation methodology developed in INEX from 2002 to 2006. The paper focusses on the main INEX track, the ad-hoc retrieval track.

2 The INEX test-beds

In traditional IR test collections, documents are considered as units of unstructured text, queries are generally treated as bags of terms or phrases, and relevance assessments provide judgments whether a document as a whole is relevant to a query or not. Although a test collection for XML IR consists of the same three parts, each component is rather different from its traditional IR counterpart. XML documents organize their content into smaller, nested structural elements. Each of these elements in the document's hierarchy, along with the document itself (the root of the hierarchy), represent a retrievable unit. In addition, with the use of XML query languages, users of an XML IR system can express their information need as a combination of content and structural conditions, e.g. users can restrict their search to specific structural elements within the collection. Consequently, the relevance assessments for an XML collection must also consider the

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

structural nature of the documents and provide assessments at different levels of the document hierarchy. In addition, all the above are assembled specifically for evaluating particular retrieval tasks.

2.1 Document Collections

Up to 2004, the collection consisted of the full-text of 12,107 articles, marked-up in XML, from 12 magazines and 6 transactions of the IEEE Computer Society's publications, covering the period of 1995-2002, and totalling 494 MB in size, and 8 million in number of elements. The collection contained scientific articles of varying length. On average, an article contains 1,532 XML nodes, where the average depth of the node is 6.9. In 2005, the collection was extended with further publications from the IEEE Computer Society. A total of 4,712 new articles from the period of 2002-2004 were added, giving a total of 16,819 articles, and totalling 764MB in size and 11 million in number of elements.

The overall structure of a typical article consists of a front matter, a body, and a back matter. The front matter contains the article's metadata, such as title, author, publication information, and abstract. Following it is the article's body, which contains the actual content of the articles. The body is structured into sections, sub-sections, and sub-sub-sections. These logical units start with a title, followed by a number of paragraphs. In addition, the content has markup for references (citations, tables, figures), item lists, and layout (such as emphasised and bold faced text), etc. The back matter contains a bibliography and further information about the article's authors.

INEX 2006 uses a different document collection, made from English documents from Wikipedia[3] [5]. The collection consists of the full-texts, marked-up in XML, of 659,388 articles of the Wikipedia project, and totaling more than 60 GB (4.6 GB without images) and 30 million in number of elements. The collection has a structure similar to the IEEE collection. On average, an article contains 161.35 XML nodes, where the average depth of an element is 6.72.

2.2 Topics

Querying XML documents can be with respect to content and structure. Taking this into account, INEX identified two types of topics:

- Content-only (CO) topics are requests that ignore the document structure and are, in a sense, the traditional topics used in IR test collections. Their resemblance to traditional IR queries is, however, only in their appearance. They pose a challenge to XML retrieval in that the retrieval results to such topics can be elements of various complexity, e.g. at different levels of the XML documents' structure.
- Content-and-structure (CAS) topics are requests that contain conditions referring both to content and structure of the sought elements. These conditions may refer to the content of specific elements (e.g. the elements to be returned must contain a section about a particular topic), or may specify the type of the requested answer elements (e.g. sections should be retrieved).

[3] http://en.wikipedia.org

CO and CAS topics reflect two types of users with varying levels of knowledge about the structure of the searched collection. The first type simulates users who either do not have any knowledge of the document structure or who choose not to use such knowledge. The need for this type of query stems from the fact that users may not care about the structure of the result components or may not be familiar with the exact structure of the XML documents. This profile is likely to fit most users searching XML repositories. The second type of users aims to make use of any insight about the document structure that they may possess. CAS topics simulate users who do have some knowledge of the structure of the searched collection. They may then use this knowledge as a precision enhancing device in trying to make the information need more concrete. This user type is more likely to fit, e.g., librarians.

As in TREC, an INEX topic consists of the standard title, description and narrative fields. For CO topics, the title is a sequence of terms. For CAS topics, the title is expressed using the NEXI query language, which is a variant of XPATH defined for content-oriented XML retrieval evaluation - it is more focussed on querying content than many of the XML query languages [18][4]. An example of a CAS topic is given in Figure 1. Different to the `contain` criteria of the XPath query language, an element can be `about` "intelligent transportation system" without actually containing any of the three words "intelligent", "transportation" and "system".

```
<inex_topic topic_id="76" query_type="CAS">
<title>
  //article[(./fm//yr = ’2000’ OR ./fm//yr = ’1999’) AND about(.,
  ’"intelligent transportation system"’)]//sec[about(.,’automation
  +vehicle’)]
</title>
<description>
  Automated vehicle applications in articles from 1999 or
  2000 about intelligent transportation systems.
</description>
<narrative>
  To be relevant, the target component must be from an
  article on intelligent transportation systems published in 1999 or
  2000 and must include a section which discusses automated vehicle
  applications, proposed or implemented, in an intelligent
  transportation system.
</narrative>
</inex_topic>
```

Fig. 1. A CAS topic from the INEX 2003 test collection

[4] The topic title format used in INEX 2002 was not based on a variant of XPATH [6], and as such led to often ambiguous queries, with respect to constraints imposed on the structure. Using the NEXI title topic format, not only removed ambiguity, but was also more in line with current development of XML query languages [1].

In 2005, in an effort to investigate the usefulness of structural constraints, variants of the CO and CAS topics were developed. CO topics were extended into Content-Only + Structure (CO+S) topics. The aim was to enable the performance comparison of an XML system across two retrieval scenarios on the same topic, one when structural constraints are taken into account (+S) and the other when these are ignored (CO). The CO+S topics included an optional field called CAS title (<castitle>), which was a representation of the same information need contained in the <title> field of a CO topic but including additional knowledge in the form of structural constraint. CAS titles were expressed in the NEXI query language. An example is given in Figure 2.

```
<inex_topic topic_id="231" query_type="CO+S">
 <title>markov chains in graph related algorithms</title>
 <castitle>//article//sec[about(.,"markov chains" algorithm graphs)]
 </castitle>
 <description>Retrieve information about the use of markov chains in
    graph theory and in graphs-related algorithms.
 </description>
 <narrative>I have just finished my Msc. in mathematics, in the field
    of stochastic processes. My research was in a subject related to
    Markov chains. My aim is to find possible implementations of my
    knowledge in current research. I'm mainly interested in
    applications in graph theory, that is, algorithms related to graphs
    that use the theory of markov chains. I'm interested in at
    least a short specification of the nature of implementation (e.g.
    what is the exact theory used, and to which purpose), hence the
    relevant elements should be sections, pagargaphs or even abstracts
    of documents, but in any case, should be part of the content of the
    document (as opposed to, say, vt, or bib).
 </narrative>
</inex_topic>
```

Fig. 2. A CO+S topic from the INEX 2005 test collection

How to interpret the structural constraints (whether as part of CAS or CO+S topics) evolved over the years, since each structural constraint could be considered as a strict (must be matched exactly) or vague (do not need to be matched exactly) criterion. In the latter case, structural constraints were to be viewed as hints as to where to look for relevant information. In 2002, the structural constraints of CAS topics were strictly interpreted. In 2003, both interpretations, strict and vague, were followed, whereas since 2004 only the latter was followed[5].

As of today, INEX has a total of 401 topics (60 (2002), 66 (2003), 63 (2004), 87 (2005) and 125 (2006)).

[5] A dedicated investigation on the interpretations on structural constraints and their impact on retrieval effectiveness was performed in 2005 using CAS topics specifically developed for this purpose. This in not reported here, but see [20] for details.

2.3 Retrieval tasks

The main INEX activity is the ad-hoc retrieval task. In IR literature [21], ad-hoc retrieval is described as a simulation of how a library might be used and involves the searching of a static set of documents using a new set of topics. Here, the collection consists of XML documents, composed of different granularity of nested XML elements, each of which represents a possible unit of retrieval. The user's query may also contain structural constraints or hints in addition to the content conditions. In addition, the output of an XML retrieval system may follow the traditional ranked list presentation, or may extend to non-linear forms, such as grouping of elements per document.

A major departure from traditional IR is that XML retrieval systems need not only score elements with respect to their relevance to a query, but also determine the appropriate level of element granularity to return to users, thus allowing focussed access to XML documents. In INEX, a relevant element is defined to be at the *right level of granularity* if it discusses all the topics requested in the user query – it is *exhaustive* to the query – *and* does not discuss other topics – it is *specific* to that query. With this definition, it is possible to differentiate, for example, between the only relevant section in an encyclopaedia from the whole encyclopaedia. Although both may be relevant to a given user query, the former is likely to trigger higher user satisfaction as it will be more specific to the query than the encyclopaedia.

Up to 2004, ad-hoc retrieval was defined as the *general* task of returning, instead of whole documents, those XML elements that are most specific and exhaustive to the user's query. In other words, systems should return components that contain as much relevant information and as little irrelevant information as possible. Within this general task, several sub-tasks were defined, where the main difference was the treatment of the structural constraints.

The *CO sub-task* makes use of the CO topics, where an effective system is one that retrieves the most specific elements and only those, which are relevant to the topic of request. The *CAS sub-task* makes use of CAS topics, where an effective system is one that retrieves the most specific document components, which are relevant to the topic of request and match, either strictly or vaguely, the structural constraints specified in the query. In 2002, a strict interpretation of the CAS structural constraints was adopted, whereas in 2003, both, a strict and a vague interpretations was followed, leading to the *SCAS sub-task* (strict content-and-structure), defined as for the INEX 2002 CAS sub-task, and the *VCAS sub-task* (vague content-and-structure). In that last sub-task, the goal of an XML retrieval system was to return relevant elements that may not exactly conform to the structural conditions expressed within the user's query, but where the path specifications should be considered hints as to where to look. In 2004, the two sub-tasks investigated were the CO sub-task, and the VCAS sub-task. The SCAS sub-task was felt to be an unrealistic task because specifying an information need is not an easy task, in particular for semi-structured data with a wide variety of tag names.

However, within this general task, the actual relationship between retrieved elements was not considered, and many systems returned overlapping elements (e.g. nested elements). Indeed, the top 10 ranked systems for the CO sub-task in INEX 2004 contained between 70% to 80% overlapping elements. This had very strong implications with respect to measuring effectiveness (Section 3), where approaches that attempted to implement a more focussed approach (e.g. retrieving the best elements, e.g., between two nested relevant elements, returning the one most specific to the query) performed poorly. As a result, the *focussed sub-task* was defined in 2005, intended for approaches concerned with the focussed retrieval of XML elements, i.e. aiming at targeting the appropriate level of granularity of relevant content that should be returned to the user for a given topic. The aim was for systems to find the most exhaustive and specific element on a path within a given document containing relevant information and return to the user only this most appropriate unit of retrieval. Returning overlapping elements was not permitted. The INEX ad-hoc general task, as carried out by most systems up to 2004, was renamed in 2005 as the *thorough sub-task*.

Within all the above sub-tasks, the output of XML retrieval systems was assumed to be a ranked list of XML elements, ordered by their presumed relevance to the query, whether overlapping elements were allowed or not. However, user studies [17] suggested that users were expecting to be returned elements grouped per document, and to have access to the overall context of an element. The *fetch & browse task* was introduced in 2005 for this reason. The aim was to first identify relevant documents (the fetching phase), and then to identify the most exhaustive and specific elements within the fetched documents (the browsing phase). In the fetching phase, documents had to be ranked according to how exhaustive and specific they were. In the browsing phase, ranking had to be done according to how exhaustive and specific the relevant elements in the document were, compared to other elements in the same document.

In 2005, no explicit constraints were given regarding whether returning overlapping elements within a document was allowed. The rationale was that there should be a combination of how many documents to return, and within each document, how many relevant elements to return. In 2006, the same task, renamed the *relevant in context sub-task*, required systems to return for each article an unranked set of non-overlapping elements, covering the relevant material in the document. In addition, a new task was introduced in 2006, the *best in context sub-task*, where the aim was to find the best-entry-point, here a single element, for starting to read articles with relevant information. This sub-task can be viewed as the extreme case of the fetch & browse approach, where only one element is returned per article.

2.4 Relevance

Most dictionaries define relevance as "pertinence to the matter at hand". In terms of IR, it is usually understood as the connection between a retrieved item and the user's query. In XML retrieval, the relationship between a retrieved item and the user's query is further complicated by the need to consider the structure in the documents. Since retrieved elements can be at any level of granularity, an element and one of its child elements can both be relevant to a given query, but the child element may be more focussed on the topic of the query than its parent element, which may contain additional irrelevant content. In this case, the child element is a better element to retrieve than its parent element, because not only it is relevant to the query, but it is also specific to the query. To accommodate the specificity aspect, INEX defined in 2002 relevance along two dimensions:

- **Topical relevance**, which reflects the extent to which the information contained in an element satisfies the information need, i.e. measures the *exhaustivity* of the topic within an element.
- **Component coverage**, which reflects the extent to which an element is focussed on the information need, and not on other, irrelevant topics, i.e. measures the *specificity* of an element with regards to the topic.

A multiple degree relevance scale was necessary to allow the explicit representation of how exhaustively a topic is discussed within an element with respect to its child elements. For example, a section containing two paragraphs may be regarded more relevant than either of its paragraphs by themselves. Binary values of relevance cannot reflect this difference. INEX therefore adopted a four-point relevance scale [12]:

- Irrelevant (0): The element does not contain any information about the topic of request.
- Marginally relevant (1): The element mentions the topic of request, but only in passing.
- Fairly relevant (2): The element discusses the topic of request, but not exhaustively.
- Highly relevant (3): The element discusses the topic of request exhaustively.

As for topical relevance, a multiple degree scale was also necessary for the component coverage dimension. This is to allow to reward retrieval systems that are able to retrieve the appropriate ("exact") sized elements. For example, a retrieval system that is able to locate the only relevant section in a book is more effective than one that returns a whole chapter. A four-point relevance scale for component coverage was adopted:

- No coverage (N): The topic, or an aspect of the topic, is not a theme of the element.
- Too large (L): The topic, or an aspect of the topic, is only a minor theme of the element.
- Too small (S): The topic, or an aspect of the topic, is the main or only theme of the element, but the component is too small to act as a meaningful unit of information.
- Exact coverage (E): The topic, or an aspect of the topic, is the main or only theme of the element, and the element acts as a meaningful unit of information.

Based on the combination of the topical relevance and component coverage, it becomes possible to identify those relevant elements, which are both exhaustive and specific to the topic of request and hence represent the most appropriate unit to return to the user. In the evaluation we can then reward systems that are able to retrieve these elements.

In a study of the collected assessments for 2002, the use of “too small” led to some misinterpretations while assessing the coverage of an element [11]. The problem was that, for CAS topics, the “too small” and “too large” coverage categories were incorrectly interpreted as the relation between the actual size of the result element and the size of the target element instead of the relation between the relevant and irrelevant contents of the result element. To address this issue, INEX 2003 renamed the two dimensions to exhaustivity and specificity:

- **Exhaustivity**, which measures how exhaustively an element discusses the topic of the user's request.
- **Specificity**, which measures the extent to which an element focuses on the topic of request (and not on other, irrelevant topics).

The scale for the exhaustivity dimension, which replaces the topical relevance dimension, was redefined by simply replacing the word relevant to exhaustive. To avoid direct association with element size, the specificity dimension, which replaces the component coverage, adopted an ordinal scale similar to that defined for the exhaustivity dimension:

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

Although there have been arguments against the separation into two relevance dimensions, this was believed to provide a more stable measure of relevance than if assessors were asked to rate elements on a single scale. One reason for this is that assessors are likely to place varying emphasis on these two dimensions when assigning a single relevance value. For example, one assessor might tend to rate highly specific elements as more relevant, while another might to be more tolerant of lower specificity and prefer high exhaustivity.

However, obtaining relevance assessments is a very tedious and costly task [14]. An observation made in [4] was that the assessment process could be simplified if first, relevant passages of text were identified by highlighting, and then the elements within these passages were assessed. As a consequence, at INEX 2005, the assessment method was changed, leading to the redefinition of the scales for specificity. The procedure was a two-phase process. In the first phase, assessors highlighted text fragments containing only relevant information. The specificity dimension was then automatically measured on a continuous scale [0,1], by calculating the ratio of the relevant content of an XML element: a completely highlighted element had a specificity value of 1, whereas a non-highlighted element had a specificity value of 0. For all other elements, the specificity value was defined as the ratio (in characters) of the highlighted text (i.e. relevant information) to the element size. For example, an element with specificity of 0.72 has 72% of its content highlighted.

In the second phase, for all elements within highlighted passages (and parent elements of those), assessors were asked to assess their exhaustivity. Following the outcomes of extensive statistical analysis performed on the INEX 2004 results [13] - which showed that in terms of comparing retrieval effectiveness, the same conclusions could be drawn using a smaller number of grades for the exhaustivity dimension[6] - INEX 2005 adopted the following $3 + 1$ exhaustivity values:

[6] The same observation was reached for the specificity dimension, but as the assessment procedure was changed in INEX 2005, the new highlighting process allowed for a continuous scale of specificity to be calculated automatically.

- Highly exhaustive (2): the element discussed most or all aspects of the query.
- Partly exhaustive (1): the element discussed only few aspects of the query.
- Not exhaustive (0): the element did not discuss the query.
- Too Small (?): the element contains relevant material but is too small to be relevant on it own.

The category of "too small" was introduced to allow assessors to label elements, which although contained relevant information, were too small to sensibly reason about their level of exhaustivity. In 2002 the "too smal" category was with respect to the specificity aspect of relevance, whereas in 2005, it is a degree of exhaustivity, and was deemed necessary to free assessors from the burden of having to assess very small text fragments whose level of exhaustivity could not be sensibly decided.

As the ultimate aim of an evaluation is to be able to state that a system performs consistently better than another system, a continuous discussion in INEX was whether such a sophisticated definition of relevance, and in particular the exhaustivity dimension, was needed. A simpler definition, e.g. using one dimension, would be less costly to obtain, and an analysis of the results may arrive at the same conclusion. In addition, assessors felt that gauging exhaustivity was a cognitively difficult task to perform, and that the extra burden led to less consistent assessments [19] compared to those obtained at TREC. An extensive statistical analysis was performed on the INEX 2005 results [13], which showed that in terms of comparing retrieval performance, not using the exhaustivity dimension led to similar results in terms of comparing retrieval effectiveness. As a result, INEX 2006 dropped the exhaustivity dimension, and relevance was defined only along the specificity dimension.

3 Metrics

Since its launch in 2002, INEX has been challenged by the issue of how to measure an XML retrieval system's effectiveness. The main complication comes from the necessity to consider the dependency between elements when evaluating effectiveness. Unlike traditional IR, users in XML retrieval have access to other, structurally related elements from returned result elements. They may hence locate additional relevant information by browsing or scrolling. This motivates the need to consider so-called *near-misses*, which are elements from where users can access relevant content, within the evaluation. The alternative, to ignore near-misses, would lead to a strict evaluation scenario, especially when dealing with fine-grained XML documents. In this section, we restrict ourselves to the metrics used to evaluate the thorough and focussed sub-tasks, as the evaluation of the other sub-tasks is still an on-going research issue.

The effectiveness of most ad-hoc retrieval tasks is measured by the established and widely used precision and recall metrics, or their variants. From 2002 to 2004, INEX used the *inex_eval metric* [6], which applies the measure of *precall* [15] to XML elements. As for precision and recall, inex_eval is based on a counting mechanism, i.e. based on number of retrieved and relevant elements.

As a consequence, if we consider near-misses when evaluating retrieval effectiveness, then systems that return *overlapping* elements (e.g. both a paragraph and its enclosing section) will be evaluated as more effective than those that do not return overlapping elements (e.g. either the paragraph or its enclosing section). If both the paragraph and its enclosing section are relevant, then this family of effectiveness measures will count both these nested elements as separate relevant components that increase the count of relevant and retrieved elements. Therefore, despite not retrieving entirely new relevant information, systems that favour the retrieval of overlapping components would receive higher effectiveness scores. [10] showed that unless this issue was explicitly addressed, unfair advantage could be gained by systems that deliberately return overlaps over XML retrieval systems that put effort into reducing redundant information being retrieved.

The first step to address this problem - as discussed in Section 2.3 - was to define the two sub-tasks, thorough and focussed, to distinguish between systems that were interested in estimating the relevance of elements given a topic of request, and those that aimed at providing focussed access to XML content. Using the inex_eval measure to evaluate the thorough sub-task is then appropriate.

With respect to the focussed sub-task, as we still want to appropriately reward the retrieval of near-misses, we need to differentiate between those elements that should be retrieved, i.e. the desired elements, and those elements that are structurally related to the desired elements, i.e. the near-misses. It was therefore necessary to separate between these two sets by marking a subset of the relevant elements in the recall-base as ideal answers, i.e. the so-called desired elements. We refer to this set as the *ideal recall-base*. However, using inex_eval on the ideal-recall base to evaluate the focussed sub-task would mean that near-misses cannot be considered when evaluating retrieval performance. As a result, INEX adopted in 2005 a new metric, called *XCG* for both sub-tasks.

The XCG measures are an extension of the Cumulated Gain (CG) based measures [8], and include the user-oriented measures of normalized extended cumulated gain and the system-oriented effort-precision/gain-recall measures. These measures are not based on a counting mechanisms, but on cumulated gains associated with return results, which were are appropriate to evaluate the focussed sub-task where near-misses are considered. For sake of consistency, the same family of measures were also adopted to evaluate the thorough sub-task.

For each returned element, a gain value, $xG[.]$, is calculated, which is a value in the interval $[0, 1]$. A value of 0 reflects no gain, 1 is the highest gain value, and values between 0 and 1 represent various gain levels. The gain value depends on the element's exhaustivity and specificity. Given that INEX employs two relevance dimensions, the gain value is calculated as a combination of these dimensions, thus reflecting the worth of a retrieved element. INEX uses *quantisation functions* to provide a relative ordering of the various combinations of exhaustivity and specificity values and a mapping of these to a single relevance scale in $[0, 1]$. Various quantisation functions have been used over the years as a means to model assumptions regarding the worth of retrieved elements to users or scenarios. INEX 2003 used the quantisations defined below, where e and s stand, respectively, for exhaustivity and specificity.

$$quant_{strict}(e, s) := \begin{cases} 1 & \text{if } e = 3 \text{ and } s = 3, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The strict function is used to evaluate XML retrieval methods with respect to their capability of retrieving highly exhaustive and highly specific components. This function models the scenario where only highly specific and highly exhaustive components are considered worthy.

$$quant_{gen}(e,s) := \begin{cases} 1 & \text{if } (e,s) = (3,3), \\ 0.75 & \text{if } (e,s) \in \{(2,3),(3,\{2,1\})\}, \\ 0.5 & \text{if } (e,s) \in \{(1,3),(2,\{2,1\})\}, \\ 0.25 & \text{if } (e,s) \in \{(1,2),(1,1)\}, \\ 0 & \text{if } (e,s) = (0,0). \end{cases} \quad (2)$$

The generalised function rewards retrieved elements according to their degree of relevance, thus also allowing to reward fairly and marginally relevant elements, which is equivalent to considering near misses when calculating effectiveness performance. Other quantisations were introduced in subsequent years of INEX, emphasising specificity or exhaustivity[7]. A statistical analysis of the INEX 2004 results [13], however, shows that, although quantisation functions express different user preferences, many of them behave similarly when ranking systems. As a consequence, one form of strict and one form of general quantizations have been used since 2005, and were modified to adapt to the new scale used in INEX 2005:

$$quant_{strict5}(e,s) := \begin{cases} 1 & \text{if} \quad e = 2 \quad \text{and} \quad s = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

$$quant_{gen5}(e,s) := e \cdot s \quad (4)$$

$quant_{gen5}$ ignores elements assessed as "too small". To consider too small elements within the evaluation, $quant_{genLifted}$ was introduced, which adds $+1$ to lift all values of exhaustivity. The effect of this is that it allows the scoring of too small elements as near-misses.

$$quant_{genLifted}(e,s) := (e+1) \cdot s \quad (5)$$

A statistical analysis of the INEX 2005 results [13] shows reasonably high agreement between which system pairs are identified as significantly different when using $quant_{gen5}$ and $quant_{genLifted}$. Thus, whether or not the “too small” elements are considered relevant did not make a large difference in the rankings of systems .

In INEX 2006, as the exhaustivity dimension was dropped, the quantization function simply maps an element to its specificity value.

Given a ranked list of elements e_j, each with their calculated gain value $xC[e_j] = quant(e_j)$ where $quant$ is a chosen quantisation function, the cumulative gain at rank i, denoted as $xCG[i]$, is computed as the sum of the relevance scores up to that rank:

$$xCG[i] := \sum_{j=1}^{i} xC(e_j)) \quad (6)$$

[7] More details can be found at http://homepages.cwi.nl/ arjen/INEX/.

For each query, an ideal gain vector, xCI, is derived by filling the rank positions with $xG(c'_j))$ in decreasing order for all assessed elements c'_j. A retrieval run's xCG vector is compared to this ideal ranking by plotting both the actual and ideal cumulative gain functions against the rank position. Normalised xCG ($nxCG$) is:

$$nxCG[i] := \frac{xCG[i]}{xCI[i]} \tag{7}$$

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

xCG and $nxCG$ correspond to user-oriented measures. The system-oriented evaluation measure is effort-precision/gain-recall measures ($MAep$). The effort-precision ep at a given gain-recall value gr is defined as the number of visited ranks required to reach a given level of gain relative to the total gain that can be obtained, and is defined as:

$$ep[r] := \frac{i_{ideal}}{i_{run}} \tag{8}$$

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

$$gr[i] := \frac{xCG[i]}{xCI[n]} = \frac{\sum_{j=1}^{i} xG[j]}{\sum_{j=1}^{n} xI[j]} \tag{9}$$

where n is the number of elements c where $xC[c] > 0$. This method follows the same viewpoint as standard precision/recall, where recall is the control variable and precision the dependent variable. Here, the gain-recall is the control variable and effort-precision is the dependent variable. As with precision/recall, interpolation techniques are used to estimate effort-precision values at non-natural gain-recall points, e.g. when calculating effort-precision at standard recall points of $[0.1, 1]$, denoted as e.g. ep@0.1. For this purpose, a simple linear interpolation method is used. Also, the *non-interpolated mean average effort-precision*, denoted as $MAep$, is calculated by averaging the effort-precision values obtained for each rank where a relevant document is returned.

After extensive analysis on the correlation between the different XCG measures [9], INEX 2006 has adopted $MAep$ for the thorough sub-task, and $nxCG$ (with cut-off values of 5, 10, 25 and 50) for the focussed sub-task, reflecting the viewpoint that the former is a system-oriented task while the latter is a user-oriented task.

In the case of the thorough sub-task (when overlap is not issue), the full recall-base is used to derive both the ideal gain vector xCI and the gain vectors, xCG. [16] showed that using $MAep$ leads to the same results in terms of retrieval systems effectiveness than when using the inex_eval measure, which is what would be expected.

For the focussed retrieval task, the elements in the ideal recall-base represent the desired target elements that should be retrieved, while all other elements in the full recall-base may be awarded partial scores. In this case, the ideal gain vector xCI is derived from the ideal recall-base, whereas the gain vectors, xCG, for the retrieval approaches under evaluation is based on the full recall-base to enable the scoring of near-miss elements. As all relevant elements of the full recall-base not included in the ideal recall-base are considered as near-misses, this strategy allows to support the evaluation viewpoint whereby elements in the ideal recall-base *should* be retrieved, whereas the retrieval of near-misses *could* be rewarded as partial success.

The construction of the ideal recall-base requires a preference function among exhaustivity and specificty value pairs, and a methodology for traversing an XML document (its tree structure) and selecting ideal elements based on their relative preference relations to their structurally related elements. A number of preference relations can be used, based on the quantisation functions used in INEX as these reflect the worth of retrieved elements. Given a chosen quantisation function, it is possible to quantify the value or worth of an element and identify the "best" components within an XML document as those elements with the highest quantised score. Similarly to the quantisation functions, different methodologies for deriving an ideal recall-base may be applied reflecting different strategies. That adopted in INEX is to traverse the XML tree of a document bottom up and selecting the element with the highest highest quantised score. From two elements with equal score the ascendant one is selected.

References

1. S. Amer-Yahia and M. Lalmas. XML Search: Languages, INEX and scoring. *SIGMOD Record*. To Appear.
2. R.A. Baeza-Yates, N. Fuhr, and Y.S. Maarek. “XML and information retrieval" SIGIR workshop *SIGIR Forum*, 36(2), 2002.
3. H.M. Blanken, T. Grabs, H.-J. Schek, R. Schenkel, and G. Weikum, editors. *Intelligent Search on XML Data, Applications, Languages, Models, Implementations, and Benchmarks*, 2003.
4. C. Clarke. Range results in XML retrieval. In *INEX 2005 Workshop on Element Retrieval Methodology*, 2005.
5. L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 40(1), 2006.
6. N. Gövert and G. Kazai. Overview of the INitiative for the Evaluation of XML retrieval (INEX) 2002. *INEX 2002 Workshop Proceedings*, 2003.
7. N. Gövert, N. Fuhr, M. Lalmas, and G. Kazai. Evaluating the effectiveness of content-oriented XML retrieval methods. JIR, 9(6), 2006.
8. K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *TOIS*, 20(4), 2002.
9. G. Kazai and M. Lalmas. INEX 2005 evaluation metrics. *INEX 2005 Workshop proceedings*, 2005.
10. G. Kazai, M. Lalmas, and A.P. de Vries. The overlap problem in content-oriented XML retrieval evaluation. *SIGIR* 2004.
11. G. Kazai, S. Masood, and M. Lalmas. A study of the assessment of relevance for the INEX’02 test collection. *ECIR*, 2004.
12. J. Kekäläinen and K. Järvelin. Using graded relevance assessments in IR evaluation. *JASIST*, 53(13), 2002.
13. P. Ogilvie and M. Lalmas. Investigating the exhaustivity dimension in content-oriented XML element retrieval evaluation. *CIKM* 2006.

14. B. Piwowarski, and M. Lalmas. Providing consistent and exhaustive relevance assessments for XML retrieval evaluation. *CIKM*, 2004.
15. V. V. Raghavan, P. Bollmann, and G. S. Jung. A critical investigation of recall and precision as measures of retrieval system performance. *TOIS*, 7(3), 1989.
16. G. Kazai, and M. Lalmas. eXtended Cumulated Gain Measures for the Evaluation of Content-oriented XML Retrieval. TOIS, 24(4), 2006.
17. A. Tombros, S. Malik, and B. Larsen. Report on the INEX 2004 interactive track. *ACM SIGIR Forum*, 39(1), 2005.
18. A. Trotman and B. Sigurbjornsson. Narrowed extended XPATH I (NEXI). *INEX 2004 Workshop Proceedings*,2004.
19. Andrew Trotman. Wanted: Element retrieval users. *INEX 2005 Workshop on Element Retrieval Methodology*, 2005.
20. A. Trotman and M. Lalmas. Strict and vague interpretation of XML-retrieval queries. *ACM SIGIR*, 2006.
21. E. M. Voorhees and D. K. Harman, editors. *The Tenth Text REtrieval Conference (TREC 2001)*, 2001.