Advances in XML Retrieval: The INEX Inititative

Norbert Fuhr, University of Duisburg-Essen, Germany
Mounia Lalmas, Queen Mary University of London, UK

February 26, 2007

Abstract

We given a survey over the INEX initiative, which focuses on the evaluation of content based access to XML documents. First, we describe the test setting and the various tracks of INEX. Then we present a new framework for the different views on XML retrieval, where we distinguish between the structural and the content dimension; in this space, current activities are located as well as new areas of research are pointed out. Finally, we discuss the combination of semantic web technologies and XML retrieval, pointing out potential benefits as well as the need for further research in this area.

1 Introduction

The increasing importance of XML as standard document format has led to substantial research efforts in the area of XML information retrieval, aiming at supporting content-oriented XML retrieval. These systems exploit the available structural information in documents, as marked up in XML, 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 way, retrieval is improved in two aspects: On one hand, exploitation of the semantics of the XML markup increases retrieval precision. On the other hand, the more focussed results answers help users to locate the relevant information more quickly. 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 [Voorhees & Harman 02].

The evaluation of XML retrieval systems thus makes it necessary to build test collections that are suitable for this kind of retrieval. The INitiative for the Evaluation of XML retrieval (INEX)[1] , 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 [Fuhr et al. 04, Fuhr et al. 05, Fuhr et al. 06]. In the following, we first give a survey over the test collections and tracks of the INEX initiative. Then we present a framework for the different views on XML retrieval, in order to locate current activities and point out further areas of research.

2 The INEX test-beds

In traditional IR test collections, documents are considered as atomic 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. Below, we describe the INEX counterparts of each of these three components.

2.1 Document Collections

Up to 2004, the collection consisted 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. 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 [Denoyer & Gallinari 06]. 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

[1] http://inex.is.inf.uni-due.de

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).

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 have no 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.

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. An example of a CAS topic is given in Figure 1. Here the about predicate refers to content of the element. 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".

In INEX, different interpretations of the structural constraints were regarded, since each structural constraint could be considered as a strict (must be matched exactly) or vague (does 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.

```
<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>
```

Figure 1: A CAS topic from the INEX 2003 test collection

2.3 Retrieval tasks

The main INEX activity is the ad-hoc retrieval task. 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.

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 a book from the whole book. 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 book.

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 relevance along two dimensions:

- 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).

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. In a similar way, a four-valued scale was defined for describing specificity.

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 be more tolerant of lower specificity and prefer high exhaustivity.

However, obtaining relevance assessments is a very tedious and costly task. An observation made in [Clarke 05] 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. In the second phase, for all elements within highlighted passages (and parent elements of those), assessors were asked to assess their exhaustivity.

An extensive statistical analysis was performed on the INEX 2005 results [Ogilvie & Lalmas 06], which showed that in terms of comparing retrieval performance, not using the exhaustivity dimension led to similar results. As a result, INEX 2006 dropped the exhaustivity dimension, and relevance was defined only along the specificity dimension.

2.5 INEX tracks

In addition to the ad-hoc track described above, there is a number of other tracks in INEX which focus on different types of tasks which are important in the area of content-based access to ML documents. Here we briefly describe some of these tracks.

Interactive track The main motivation for the track is twofold. First, to investigate the behaviour of users when interacting with elements of XML documents, and secondly to investigate and develop approaches for element retrieval which are effective in user-based environments.

For the first goal, the observation data from the user-system interaction gives a good basis for the definition of appropriate retrieval metrics; among other issues, the empirical data shows that in many cases, users prefer to retrieve elements instead of full articles, thus supporting the major hypothesis of the content-only retrieval task. Another major outcome of this track was the need to investigate methods that can be supportive during the search process based on features extracted from the XML formatting. In the first round of this track, users were puzzled by being presented (possibly overlapping) components from the same document at various positions in the result list, mixed with elements from other documents. Thus, in the following year, the XML structure was exploited to achieve a better and more comprehensible presentation of results, e.g., by hierarchical hit lists and highlighting document parts.

Multimedia track This track aims at the evaluation of structured document retrieval approaches which are able to combine the relevance of the different media types into a single (meaningful) ranking that is presented to the user. So far, the focus has been on the combination of (XML) text and image retrieval.

Document Mining The document mining track focuses on the two generic tasks of classification and clustering, by developing methods that are able to exploit the XML markup for this purpose. In addition, there is a structure specific task which deals with structure mapping between different XML DTDs.

XML entity ranking Whereas the other tracks deal with information access to XML elements, the task of this track is to return a list of entities of a specific type (e.g. people, products, artefacts). The user would specify an entity type and a topic, and then the system should return appropriate entities. For example, searching for famous actors with the topic "1930s", it should return Astaire, Chaplin, Gable and Garbo, whereas given a topic "action" should result in Schwarzenegger, Stallone and Van Damme. A variant of this task is list completion which aims at extending a given list of entities with more entities of the same type. For example, a list of SIGIR and ECIR with query context "information retrieval" should be extended with CIKM.

Heterogeneous collection track The IEEE-CS test collection used in INEX has been based on a single DTD (whereas Wikipedia doesn't have any DTD at all) In practical environments, most XML collections will consist of documents from different sources, and thus with different DTDs or Schemas. If there is a semantic diversity between the collections, not every collection is suitable to satisfy the user's information need. So a heterogeneous collection poses a number of challenges for XML retrieval, including:

- For content-oriented queries, most current approaches use the DTD for defining elements that would form reasonable answers. In heterogeneous collections, DTD-independent methods need to be developed.
- For content and structure queries, there is the additional problem of mapping structural conditions from one DTD or schema onto other (possibly unknown) DTDs and schemas.
- Both content-only and content-and-structure approaches should be able to preselect suitable collections. This way, retrieval costs can be minimized by neglecting collections which would probably not yield valuable answers but are expensive to query w.r.t. communication efforts.

3 Views on XML

In the previous section, we have described the current research activities which are evaluated in INEX. Now we want to take a more general look at information access to XML documents. Figure 2 gives a survey over the possible views on XML documents, which can also be regarded as a design space for XML IR systems. WE ditinguish two dimensions, namely the *stucture* and *content type*. The former deals with diffferent the structural aspects of XML documents, starting from a simple view as a tree (nested) structure up to the database-oriented view as implemented in the XQuery language. The second dimension deals with the types of content we may find in XML documents. In most cases, we assume all content to be text only. However, markup of an element may indicate a specific data type (e.g. a date) or even complex object types. In the following, we describe each of the two design dimensions in more detail.

3.1 XML structure

Nested Structure. Whereas classical retrieval regards documents as atomic units, the XML markup of a document immediately implies a nested, tree-like structure. Following this view, a retrieval method should be able to retrieve subtrees (i.e. complete elements) instead of complete documents only. Typical query languages for this kind of retrieval provide no specific means for specifying structural constraints, in most cases they only allow for the specification of a set of terms. The corresponding retrieval method aims at performing a relevance-oriented selection of answer elements, i.e. the system should return the most

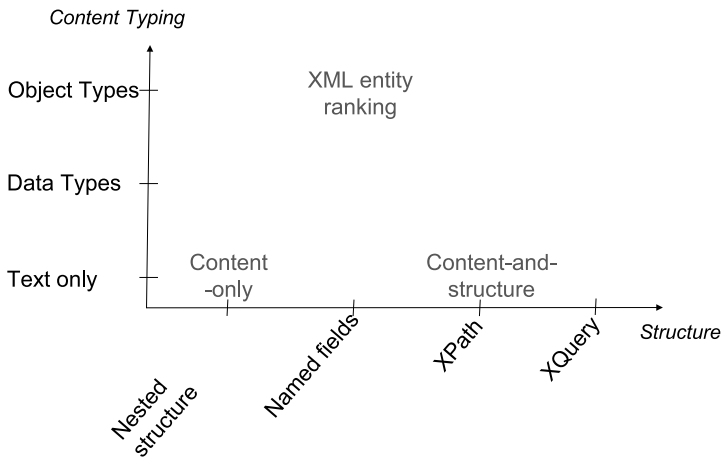

Figure 2: Views on XML

specific relevant elements. So this view corresponds to the content-only queries regarded in INEX.

Named Fields This view is somewhat orthogonal to the nested structure view: Here, we only regard the element names, without considering their structural relationships. Thus, a document can be seen as a set of named fields (sometimes also called a linear data model). The most prominent representative of this view is the Dublin Core name space[2]. Here we can refer to elements through field names only, whereas the context of an element is ignored (e.g., in a document, we may not be able to distinguish between the author of the document and that of a referenced paper). Another problem is that of false coordination: e.g., for a document with two authors from different institutions, our retrieval method may not be able to coordinate author names and their correct affiliations. On the other hand, experimental results from INEX [Kamps et al. 06] indicate that XML retrieval quality does not suffer from restriction to named fields.

XPath XPath provides full expressiveness for navigating through the document tree, by parent/child and ancestor/descendant relationships, whereas horizontal navigation is supported via operators like following/preceding, following-sibling and preceding-sibling; in addition, there are the attribute and the names-

[2]http://dublincore.org/documents/dces/

pace axis. With these operators — in combination with the specification of element names — XPath allows for the selection of arbitrary elements. However this language may be already too complex for users. For this reason, INEX is using a simplified version of XPath (called NEXI) [Trotman & Sigurbjornsson 05] as query language, which restricts navifgatiohn to the descendant axis.

XQuery XQuery offers an even higher expressiveness than XPath, due to the fact that it was developed especially for database-like applications. Thus, in addition to XPath, it supports typical database operators like joins, aggregations and constructors for restructuring results

As a simple example, assume a list of book titles with prices and publisher names stored in a file named 'bib.xml' ; then the following query would produce a list of publishers, each along with the average price of its books:

```
FOR $p IN distinct(document("bib.xml")//publisher)
LET $a := avg(document("bib.xml")//book[publisher = $p]/price)
RETURN
   <publisher>
      <name> {$p/text()} </name>
      <avgprice> {$$a} </avgprice>
   </publisher>
```

Here the `FOR` construct loops over all publishers, whereas the following `LET` retrieves all corresponding book prices and then computes their average. In the `RETURN` clause, the XML structure of the result is specified.

From an IR point of view, such a complex query language may seem to be of little use. However, restructuring of results is performed in any IR system when the result list is created — and XQuery could bring in some flexibility here. Joins may be required when relevant information is distributed over different documents, which are linked either explicitly or implicitly. Finally, aggregations also may be of interest for some specific IR queries - e.g. retrieve papers on XML retrieval, and for each author, count the number of publications in this area. (of course, if we assume non-Boolean retrieval, then we would have to sum up retrieval weights instead of counting only).

3.2 XML Content Typing

Now we regard the content dimension of XML retrieval.

Text Most of today's XML retrieval systems assume that an XML document contains only text. In some sense, they still follow the traditional view of a document as a text block, which is now structured via XML tags.

Data Types Different XML elements may contain different types of text, and this information could be exploited in retrieval. Thus, advanced IR system should support the notion of data types, where each such type is accompanied

by a set of (vague) predicates. For example, for an element containing person names, similarity search for proper names should be offered; in technical documents, elements containing measurement values should be searchable by means of the comparison predicates $>$ and $<$ operating on floating point numbers. Also, multi-language documents could be supported via the notion of data types, where each language corresponds to a separate data type, for which language-specific stemming methods can be seen as data type-specific search predicates.

Although data types play a central role in XML Schema (XMLS) [Fallside 01], this approach is of little help for retrieval purposes: XMLS supports type checking at the syntactic level only — and how could such a type checker tell the difference between e.g. German and English text. Furthermore, predicates cannot be specified as part of XMLS types. So IR applications would require a modified version of the XMLS standard.

Object Types One can even go one step further and regard objects occurring in XML documents, like for example persons, locations or companies. Objects may have several attributes (of different data types, and queries may refer to any of these attributes. As an example, regard the following text excerpt from Wikipedia:

> Pablo Picasso (October 25, 1881 - April 8, 1973) was a Spanish painter and sculptor..... In Paris, Picasso entertained a distinguished coterie of friends in the Montmartre and Montparnasse quarters, including André Breton, Guillaume Apollinaire, and writer Gertrude Stein.

If this text were marked up appropriately, a retrieval system should be able to answer queries like e.g. "To which other artists did Picasso have close relationships?"or "Where did he meet Gertrude Stein?". There is substantial work on named entity recognition methods, which allow for automatic markup of object types.

Overall, with data and object types, precision of XML retrieval can be increased.

3.3 INEX Views

Given the two-dimensional design space for XML retrieval as described above, we can now locate the current INEX activities within this space (see also Figure 2): The two major tracks (content-only and content-and-structure queries) are restricted to text as content type, where the former regards only the nested structure, whereas the structural view of the latter is very similar to XPath. In contrast, the XML entity ranking track regards objects as content types, but its structural view is restricted to named fields.

Obviously, the design space is covered very sparsely by the INEX activities. However, this picture reflects the major XML IR research activities. So far,

there is very little research in the other areas, and thus, there also is no strong desire for corresponding evaluation initiatives

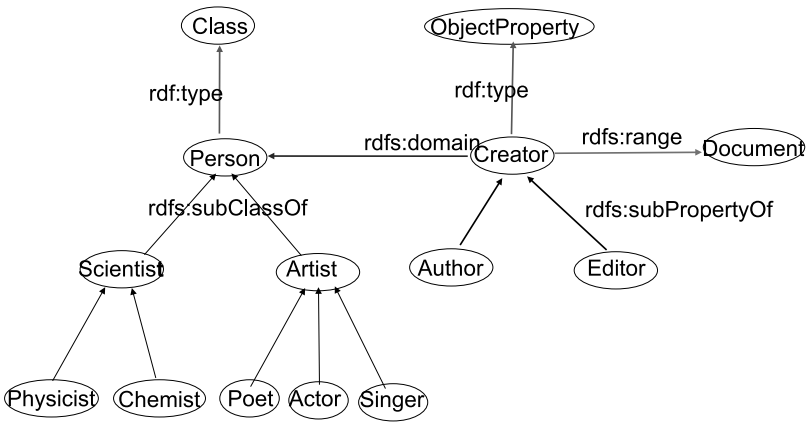

Figure 3: OWL modeling

3.4 Towards semantic retrieval of XML documents

In the discussion from above, we have not regarded the semantics of XML element names. In fact, some XML applications use rather cryptic element names (like e.g. the IEEE-CS corpus used in INEX, with element). However, for new XML applications, the effort for using meaningful element names would be only marginal. Based on this information, the semantics of tag names could be exploited. In Figure 3, we have used OWL [McGuinness & Harmelen 04] for an example modelling of descriptions of artists and scientists (e.g. in Wikipedia). A first benefit would be the possibility to search for generalizations or specializations of concepts. (e.g. searching for artists would retrieve poets, actors and singers. In a similar way, also hierarchies on properties would be supported, and the domain and range of such properties can be considered. Besides enhancing retrieval on a uniform collection, such a modelling would be extremely useful for federated collections where documents are based on different DTDs (like in the heterogeneous track of INEX); in this case, the OWL modelling could be used for a kind of schema mapping, such that queries could be formulated with regard to a single DTD or XML Schema, and the OWL model would be the basis for mapping across different schemas.

However, although substantial research has been devoted to semantic web technologies recently, the proposed kind of XML retrieval is not supported yet, due to a number of shortcomings:

- OWL currently supports XML for literals only, whereas we would like to have the full expressiveness of our XML IR language NEXI combined with OWL.
- OWL does not provide an appropriate query language for this combination.
- OWL does not support uncertain inference, which is essential for XML retrieval (in [Nottelmann & Fuhr 06], we have proposed such an extension).

So there is a need for additional research in order to exploit OWL for 'semantic' retrieval of XML documents. However, we think that such an approach is very promising, since the corresponding definitions and mappings have to be specified at the collection level only — in contrast to the standard semantic web approach which assumes that OWL descriptions are given at the document level.

4 Conclusion and outlook

In this paper, we have given a survey over the INEX initiative, which focuses on the evaluation of content based access to XML documents. We briefly described the test setting and the various tracks of INEX. Then we presented a new framework for the different views on XML retrieval, in order to locate current activities and point out further areas of research. We have shown that there is still a broad area of open research issues which hasn't been addressed yet. Especially, the combination of XML retrieval with semantic web technologies seems to be very promising.

References

Clarke, C. (2005). Range results in XML retrieval. In: *INEX 2005 Workshop on Element Retrieval Methodology.*

Denoyer, L.; Gallinari, P. (2006). The Wikipedia XML Corpus. *SIGIR Forum 40(1).*

Fallside, D. C. (2001). *XML Schema Part 0: Primer.* W3C recommendation, World Wide Web Consortium. `http://www.w3.org/TR/xmlschema-0/`.

Fuhr, N.; Lalmas, M.; Malik, S. (eds.) (2004). *INitiative for the Evaluation of XML Retrieval (INEX). Proceedings of the Second INEX Workshop. Dagstuhl, Germany, December 15–17, 2003.* `http://inex.is.informatik.uni-duisburg.de:2003/proceedings.pdf`.

Fuhr, N.; Lalmas, M.; Malik, S.; Szlavik, Z. (eds.) (2005). *Advances in XML Information Retrieval: Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004, Dagstuhl Castle, Germany, December 6-8, 2004, Revised Selected Papers*, volume 3493. Springer-Verlag GmbH. http://www.springeronline.com/3-540-26166-4.

Fuhr, N.; Lalmas, M.; Malik, S.; Kazai, G. (eds.) (2006). *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005), Dagstuhl 28-30 November 2005, Lecture Notes in Computer Science*, volume 3977. Springer-Verlag GmbH.

Kamps, J.; Marx, M.; de Rijke, M.; Sigurbjörrnsson, B. (2006). Articulating information needs in XML query languages. *ACM Transactions on Information Systems 24(4)*, pages 407–436.

McGuinness, D. L.; van Harmelen, F. (2004). *OWL Web Ontology Language: Overview.* Technical report, World Wide Web Consortium. `http://www.w3.org/TR/owl-features/`.

Nottelmann, H.; Fuhr, N. (2006). Adding Probabilities and Rules to OWL Lite Subsets based on Probabilistic Datalog. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 14(1)*, pages 17–41.

Ogilvie, P.; Lalmas, M. (2006). Investigating the exhaustivity dimension in content-oriented XML element retrieval evaluation. In: *Proceedings of ACM CIKM.* ACM, New York.

Trotman, A.; Sigurbjornsson, B. (2005). Narrowed Extended XPath I (NEXI). In [Fuhr et al. 05]. http://www.springeronline.com/3-540-26166-4.

Voorhees, E. M.; Harman, D. K. (eds.) (2002). *The Tenth Text REtrieval Conference (TREC 2001)*, Gaithersburg, MD, USA. NIST.