Sub-document level Information Retrieval: Retrieval and Evaluation

Sukomal Pal

Indian Statistical Institute
February, 2012

Sub-document level Information Retrieval: Retrieval and Evaluation

Sukomal Pal

Thesis submitted to the Indian Statistical Institute
in partial fulfillment of the requirements
for the award of
Doctor of Philosophy.
February, 2012.

Thesis supervisor: Dr. Mandar Mitra

Indian Statistical Institute
203 B T Road, Kolkata - 700108, India.

To

My parents (Maa and Bapi)

the source of my being ...

Abstract

XML is increasingly used to mark up content in present day information repositories. Over the last decade or so, retrieval from XML document collections has emerged as an area of active research. For the Information Retrieval community, XML retrieval poses a two-fold problem:

1. finding effective techniques to retrieve appropriate or the most useful XML elements in response to a user query; and
2. devising an appropriate evaluation methodology to measure the effectivity of such retrieval techniques.

This study examines both these issues. First, we revisited the pivoted length normalization scheme in the Vector Space Model using standard benchmark collections for XML retrieval. We reduced two parameters used in pivoted length normalization to a single combined parameter and experimentally found its optimum value, which works well at both the element and document levels for XML retrieval.

We observed that a substantial number of focused queries used in XML retrieval clearly state, besides the information need, what the user *does not* want. We demonstrated that this negative information, if not handled properly, degrades retrieval performance. We proposed a solution for automatically removing negative information from XML queries. This led to significant improvements in retrieval results.

On the evaluation of XML retrieval, we first studied the sensitivity & robustness of various evaluation metrics and reliability and reusability of the assessment pool that has been used at INEX Ad

Hoc track since 2007. Specifically we investigated the behaviour of the metrics when assessments are incomplete, or when query sets are small. We observed that early precision metrics are more error-prone and less stable under both these conditions. Average metric, however, performs comparatively better in this respect. System rankings remain largely unaffected even when assessment effort is substantially (but systematically) reduced. We also found that the INEX collections remain usable when evaluating non-participating systems.

For a fixed amount of assessment effort, judging shallow pools for many queries is found to be better than judging deep pools for a smaller set of queries. However, when judging only a random sample of a pool, it is better to completely judge fewer topics than to partially judge many topics.

We also proposed a simple and pragmatic approach of creating assessment pool for evaluation of retrieval systems. Instead of using an apriori-fixed pool depth for all topics, the pool is incrementally built and judged interactively on a per query basis. When no new relevant document is found for a reasonably long run of pool-depths, pooling was stopped for the topic. Our proposed approach offers a trade-off between the available effort and required level of performance. Moreover, it is flexible to *deep pooling* for potential topics in order to ensure *better* estimate of recall. We demonstrated the effectivity of the technique by substantially reducing the assessment effort without seriously compromising on the reliability of evaluation. The approach provided good results in the evaluation of XML retrieval as well as traditional document retrieval.

Acknowledgments

I was drawn to the field of ‘Information Retrieval’ by my mentor Dr. Mandar Mitra. It was my sheer luck to work under a supervisor like him who epitomises perfection and meticulousness to me. On top of academic excellence, his openness and down-to-earth personality were exemplary. He gave me immense freedom to do whatever I like and even the freedom not to do anything either. Very often he had to bear with my non-academic priorities in life and with a sage-like calmness he stood beside me. I take this opportunity to bow before him and say, ‘Thank you, Sir’.

While working on a crucial part of this thesis, I received a God-sent help from Dr. Jaap Kamps. He pinpointed the problems in my work and gave his valuable inputs. I owe a lot to him.

On several occasions, I received scholarly help, guidance and suggestions from Dr. Arnab Chakraborty, Prof. S. K. Parui, Dr. Umapada Pal, Dr. Ujjwal Bhattacharya, Dr. Utpal Garain, Dr. Sarbani Palit and many other faculties from ISI. I am really thankful to all of them.

The CVPR department in particular, and ISI in general, provided a wonderful flexi-time environment where I could work any time I liked. Amazingly, even at the most odd hours, I never felt alone in the department. There was at least one more faculty or research staff almost always. I must thank Prof. B. B. Chaudhuri, who besides providing me encouragement, created and actively nurtured this environment.

I would like to thank Dr. Prasenjit Majumder, my senior colleague in the IR lab. His ideas, suggestions, advice and above all, his tremendous sense of humour were always enriching and refreshing. In this connection, I would also like to acknowledge the all-round help and advice from Dr. Pabitra Mitra and Dr. Debrup Chakraborty.

I enjoyed and presently miss the company of my colleagues in the CVPR Unit, especially Bikash Shaw, Dipasree Pal, Jiaul Hoque Paik, Samaresh Maiti, Ayan Bandyopadhyay, Debasis Ganguly, Sauparna PalChowdhury, Kripabandhu Ghosh, Sukanya Mitra, Aparajita Sen, Ankan Bhattacharya, Anandarup Roy, and Srikanto Mondal, to name a few. They helped me on several occasions in my research and/or beyond the research. I remember their cheerful presence in the 'CVPR tea club' where we spent hours discussing myriad issues. Their lively spirit kept me up and got me going. I am really indebted to them.

The lively and homely ambience in the department and in ISI are due to the wonderful people in and around ISI. Specially I would like to mention the names of Mr. Sankar Sen, Mr. Sunil Bhar and Mrs. Bhabani Das from the CVPR office for their support on numerous occasions. There are many people in ISI whom I can not name here but who were helpful to me in many ways and are certainly worth a mention.

In this connection, I would also like to acknowledge the support and wishes of my present colleagues and especially Prof. P. K. Jana, HOD, CSE, at Indian School of Mines, Dhanbad.

On a personal note, the journey during my PhD was a roller-coaster one. I took the plunge for a PhD drawing inspiration from some of my friends and seniors like Dr. Samit Raichaudhuri, Dr. Sankardas Roy, Dr. Prishati RoyChowdhury and Dr. Kumar Bappaditya Salui. However, in between I felt completely hopeless and dejected time and again. It was my family and friends who stood beside me unconditionally; especially Maa, Bapi, Didi, Bonu, brothers-in law Ashis-da and Arju, nephew Tutan and Arka and friends like Manoj, Rana and Tilak. Mere thanks are not enough for the support and encouragement they extended to me in the time of need. Also, I must humbly acknowledge the role of my parents-in-law. For reasons best known to them, they married off their daughter to a research scholar, supported me in all possible way throughout and kept their faith in me.

During my entire academic life, I complained a lot about the inadequacy I had to live with. Now I realize that the inadequacy is part of our reality and often is the driving force to rise above it. My Bapi and Maa went extra miles, often beyond their capacity to plug the gap. They nurtured my independent thought, gave freedom to choose my career and future and never unduly influenced my decision-making process. Even if my decisions were utterly wrong and I faltered, they never intruded. They allowed me to stand up on my own and learn from the mistakes. Now I realize the value of the way they raised me up and I hope I can replicate the same to the next generation . I take it as a privilege to salute their fight against hardships of life and never-say-die attitude and deeply acknowledge their umpteen belief in me.

At last, but probably the most, I must mention two integral parts of my existence – my wife Teena and daughter Srijita. Conceiving the dream of a PhD was mine. But knowingly or unknowingly both of them had to bear the pain of a surrogate-carrier. Their sacrifice of their due from a husband and a father respectively was nothing short of a genuine contribution to this endeavor watermarked over the following pages. The errors, pitfalls and shortcomings therein are completely mine. However, fruitful revelations, if any, are equally credited to them as well.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Rapid advances in electronic and computer technology have ushered in what we call an age of information. The exponential growth of the Internet and the Web has flooded us with huge quantities of data in different formats on a variety of subjects. To manage this colossal amount of data and extract useful information out of it, we need efficient and effective means of organizing and indexing the data. Though these data are available in different forms and the amount of available multimedia data (images, speech, audio, video etc.) is rapidly increasing, textual data continues to be a fundamental and very widely prevalent form of storing information. Textual information can be broadly classified into three categories based on structure : (i) structured data (ii) unstructured data and (iii) semi-structured data.

Structured data. Structured data refers to data that is stored in a precisely defined stringent format; Database records belong to this category of data. The form of each record in a database — including the number of attributes and the type of each attribute — is fixed and unambiguously defined by the database schema. The user can get an exact answer corresponding to her information need, but the set of possible queries is limited by the simple, fixed and parameterized mode of querying. Further, one cannot extract information from a database if one does not know the database schema.

Unstructured data. In contrast, unstructured data has no fixed, pre-defined format, and is typically couched in free-flowing natural language. Much of the textual information that is crawled and indexed by popular search engines is either very loosely structured or unstructured. This information is mostly contained in HTML files, where the HTML tags such as <TITLE>, <P>, <H1> etc. impose some structure on the text. However, the tags in HTML are primarily syntactic in nature, and pertain more to the layout of the text than to its information content or semantics. Moreover, a significant amount of the HTML data available on the Web is not properly structured. For example, files often contain start tags without the corresponding end-tags, as well as improperly nested tags (even though they are rendered quite intelligently by modern browsers).

Semi-structured data. Semi-structured data, as the name suggests, lies in between the above two. It has a more well-defined structure than unstructured data, but this structure is not as rigid as in traditional databases. Semi-structured data is usually represented using XML[1] (eXtensible Mark-up Language). XML is a mark-up language with some apparent similarities to HTML, but XML imposes a more rigorous structure than HTML. More importantly, unlike HTML, XML permits user-defined tags which can specify the meaning of the data contained between them. Each data element in an XML file is mandatorily encapsulated within a start tag and an end tag. The relationship between different data elements is specified via nesting and references.

Table 1.1: **Relational database** *mydata*

authors

name	**address**	**editor**	**born**
Robert Flakes	10 Tenth St, Decapolis	Julia Ellis	1960/07/26
Jimmy Thomas	2 Second Av, Duo-Duo	Julia Ellis	-

editors

name	**telephone**
Julia Ellis	7356

XML provides a very powerful and flexible data representation method that subsumes traditional relational databases on one hand, and HTML on the other. Table 1.1 shows an example relational database comprising two tables which are represented in XML as:

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

```
<!doctype mydata "http://www.w3.org/mydata">
<mydata>
        <authors>
                <author>
                         <name>Robert Flakes</name>
                         <address>10 Tenth St, Decapolis</address>
                         <editor>Julia Ellis</editor>
                        <born>1960/07/26</born>
                 </author>
                 <author>
                         <name>Jimmy Thomas</name>
                        <address>2 Second Av, Duo-Duo</address>
                         <editor>Julia Ellis</editor>
                </author>
        </authors>
        <editors>
                 <editor>
                         <name>Julia Ellis</name>
                         <telephone>7356</telephone>
                 </editor>
         </editors>
</mydata>
```

Similarly, HTML files can also be converted into XML. The following is an example of an HTML file [150]:

```
<UL>
  <LI>
    R. Goldman, J. McHugh, and J. Widom.
    <A href="ftp://db.stanford.edu/pub/papers/xml.ps"> From Semistructured
       Data to XML: Migrating the Lore Data Model and Query Language
    </A>
    Proceedings of the 2nd International Workshop on the Web and Databases
     (WebDB ’99), pages 25-30, Philadelphia, Pennsylvania, June 1999.
  <LI> T. Lahiri, S. Abiteboul, and J. Widom.
    <A href="ftp://db.stanford.edu/pub/papers/ozone.ps"> Ozone:
       Integrating Structured and Semistructured Data
    </A>.
    Technical Report, Stanford Database Group, October 1998.
</UL>
```

After conversion, the corresponding XML file looks like:

```
<Publication URL="ftp://db.stanford.edu/pub/papers/xml.ps"
   Authors=RG JM JW">
   <Title>
        From Semistructured Data to XML: Migrating the Lore Data Model
        and Query Language
   </Title>
  <Published>
       Proceedings of the 2nd International Workshop on the Web and
       Databases (WebDB ’99)
  </Published>
  <Pages>25-30</Pages>
  <Location>
      <City>Philadelphia</City>
      <State>Pennsylvania</State>
  </Location>
  <Date>
      <Month>June</Month>
      <Year>1999</Year>
  </Date>
  </Publication>

  <Publication URL="ftp://db.stanford.edu/pub/papers/ozone.ps"
         Authors="TL SA JW">
      <Title>
         Ozone: Integrating Structured and Semistructured Data
      </Title>
      <Published>Technical Report</Published>
      <Institution>Stanford University Database Group</Institution>
       <Date>
           <Month>October</Month>
          <Year>1998</Year>
       </Date>
  </Publication>

  <Author ID=TL">  T.Lahiri </Author>
  <Author ID="SA">S. Abiteboul</Author>
  <Author ID="RG">R. Goldman</Author>
  <Author ID="JM">J. McHugh</Author>
  <Author ID="JW">J. Widom</Author>
```

Because of this capability, XML plays and will play a major role in the area of data integration from multiple sources [69].

Converting and storing HTML files in XML format has some clear advantages:

1. It eliminates the laborious, error-prone, and unmaintainable "screen-scraping"[2] [150] approach to extracting useful data from HTML Web pages.

2. Users are no longer constrained by a parameterized mode of querying (as in a database setting), but can still pose precise queries like:

    - Find all authors with two or more SIGIR publications in the same year.
    - What work has been done by Computer Science departments at various universities during 2005-2010 on semistructured data?

    and expect precise and specific results if structural information held by meaningful XML tags is leveraged during retrieval. This element-level search is generally not provided by present-day search engines, which usually crawl and index the *surface web* comprising mainly web pages with unstructured data.

3. Third is the issue of the *deep web*. The web contains huge quantities of transient pages dynamically generated from databases hidden behind a static query page. These dynamic pages are created on-the-fly using the data retrieved from a database in response to a specific user request. Such pages are generally not indexed by Web search engines. According to Bergman [9], the size of this *deep web* is 500 times larger than that of the *surface web*, but only one-third of this data is indexed by search engines [47].

    In order to provide a Web search service that is sufficiently comprehensive in scope, we need to bring this database-centric *deep web* into search ambit. For this, easy conversion and interoperability between structured and unstructured data formats is essential. Thus, in this scenario too, XML — as a semi-structured format — holds great promise.

[2] http://en.wikipedia.org/wiki/Screen_scraping

In sum, due to its power and flexibility, XML has emerged as a widely-used standard for data organization, representation and exchange on the Web and in digital libraries.

1.1 XML Retrieval

Due to the advantages of XML discussed above, it is used as a mark-up language in many large repositories of documents containing a mixture of text, multimedia, and metadata. Such information repositories increasingly contain both long as well as heterogeneous documents, covering a wide variety of topics, e.g. books, user manuals, legal documents, and customer feedback data. Owing to the mark-up, these documents can be regarded as aggregates of smaller, hierarchically structured entities that are separately indexable and retrievable [22]. This is in contrast to the traditional view of documents as atomic units of information in content-oriented retrieval (See Figure 1.1). How to effectively and efficiently index, store, query and retrieve these entities has been an active area of research for both the Database (DB) and Information Retrieval (IR) communities in recent times.

Researchers from the DB community typically try to solve the problem by applying traditional database techniques to XML data. They formulate SQL-like query languages, address the problems of data-integrity and referential integrity constraints.

Since several of the standard models and techniques of traditional IR have already been successfully applied to the Web to index, store, query and retrieve mostly unstructured documents (HTML pages), the IR community has also started to explore how these can be applied to content-oriented XML retrieval. The phenomenal growth in the number of XML repositories has been matched by increasing efforts in the development of XML IR systems. These systems exploit besides the content, structural information, both syntactic and semantic, provided by the XML mark-up, in order to return document components or XML elements instead of whole documents in response to a user query. The capability to retrieve information at the sub-document level is important from an end-user's point of view, since the effort required to locate relevant content can now be minimised by retrieving the most relevant part(s) of documents, rather than whole documents. This

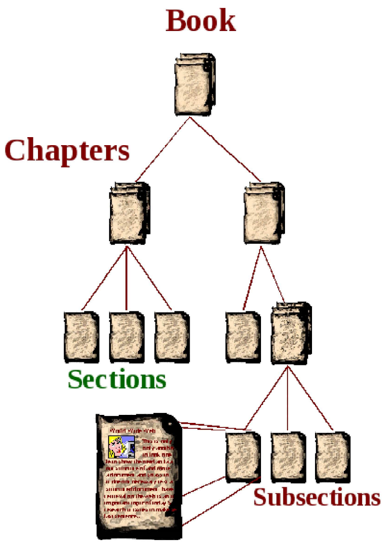


Figure 1.1: Hierarchical Structure of XML

type of focused retrieval is particularly useful when dealing with collections of long documents or documents covering a wide variety of topics as mentioned earlier. But how to locate from a hierarchically structured document, a useful area of content that is potentially relevant to the user and secondly, how to dynamically determine the most suitable granularity that should be retrieved at the sub-document level are two challenges in XML retrieval. Hence, despite some similarities with unstructured data, XML data needs special treatment so far retrieval is concerned.

From the perspective of retrieval evaluation, since a whole document can not be considered entirely *relevant* (or *irrelevant*) what should be the notion of relevance at the sub-document level? Among two competing retrieval techniques, if one retrieves a *small* element while the other a much detailed and *longer* element, both containing useful information from the same document, which one should be given credit? XML therefore needs an evaluation set-up where components like notion of relevance, the yardstick to measure retrieval effectiveness etc. are very much different from the traditional document retrieval settings.

In sum, the above issues demanded a new paradigm in the techniques and evaluation methodology of XML retrieval. This led to the formation of the *Initiative for the Evaluation of XML retrieval* (abbreviated INEX). Among several tracks (see Sec. 2.2 for details) the Ad Hoc track, which has traditionally been the main track at INEX provides the setup for the present thesis. We have been participating in this task since 2006. The work described in the following section centers around this track.

1.2 Present Research

Two important issues related to XML retrieval are:

- finding effective techniques to retrieve appropriate or the most useful XML elements in response to a user query; and
- devising an appropriate evaluation methodology to measure the effectivity of such retrieval techniques.

Retrieval Techniques: Unlike traditional IR where documents are regarded as atomic units, XML retrieval deals with elements at the sub-document level. XML elements are indexed, sought for in the queries and retrieved in response. But which elements are to be indexed? Should all elements be indexed? As XML is hierarchical in structure, such a scheme will introduce a lot of redundancy in indexing and index size will grow unnecessarily. On the other hand, if a few elements are selectively indexed, what should the basis of selection be? Also, in that case, how can the system retrieve elements that are not indexed? What will be an effective ranking strategy for the elements which are indexed and for the elements not directly indexed? There are several retrieval-related issues like these irrespective of the models we use.

Retrieval Evaluation: Similarly, on the evaluation front, what is the notion of *relevance* in the context of semi-structured retrieval? How much *coverage* of the information need is required in an element to make it relevant? If a small element contains some useful information, but its overlapping predecessor contains more useful information as well as some irrelevant information, which is better? Is element-length then a factor in the definition of relevance? Is *specificity* (how focused an element is with respect to the information need) or *exhaustivity* (how comprehensive an element is with respect to the information need) of an element the property that should be given importance in the relevance? What should be the metric(s) to capture these issues in the evaluation of XML retrieval? Do we need to define some new metric or will the traditional metrics based on *precision* and *recall* from document retrieval setting work here? Once suitable metrics are defined how do the metrics behave? Are these metrics sensitive to variations in evaluation parameters like number of queries, pool size and/or pooling methods?

1.3 Research Contributions

This thesis attempts to address some of the issues from both the aspects of retrieval and its evaluation.

1.3.1 Retrieval Techniques

We explore effective indexing and term-weighting strategies using Vector Space Model (VSM) for XML retrieval. We implement XML retrieval techniques within the SMART [3] retrieval system with automatic query expansion [81] based on VSM framework. Specifically we study the following issues:

- **Length Normalization revisited:** *parameter reduction and optimization*

  Length normalization is an important factor in VSM. A normalization scheme that retrieves documents of all lengths with similar chances as their likelihood of relevance will outperform other schemes which retrieve documents with chances very different from their likelihood of relevance. Singhal [130] introduced parameters *pivot* and *slope* within the *normalization* component of term-weighting schemes used in the VSM. The recommended values of

[3] ftp://ftp.cs.cornell.edu/pub/smart/

these parameters were determined using full-length English document collections (specifically those used at TREC[4]).

XML retrieval is a new paradigm, and the Wikipedia document collection is significantly different from the TREC collections. We, therefore, revisit length normalization issue in VSM within the setting provided by INEX. We study first *pivoted document length normalization* and then *element length normalization*. We observe that the number of parameters used in length normalization can be reduced from two (pivot and slope) to one (a single combined *pivot-slope*). We experimentally find an 'optimum' value for the said parameter. This value works surprisingly well for XML documents used in the INEX Adhoc track across a couple of years. The optimum value also yields good performance for element retrieval as well.

- **Query Refinement:** *Using negative information in queries to augment retrieval performance*

  It is easier for a user to formulate a complex information need in natural language rather than using terse keyword queries. In XML, often search queries are verbose. In the narrative (N) section of the INEX topics, a user often states, beside the information need, what (s)he *does not* want. Most of the retrieval systems do not handle this *negative information* (which defines what the user *does not* want) separately. Is this negative information really helpful in augmenting retrieval performance? If yes, can this information be automatically extracted?

  We show that the simple removal of the negative components of queries *significantly* improves retrieval performance. First we manually eliminate negative sentences from the N section of the INEX queries and observe that retrieval performance gets augmented compared to original runs. Next, training with similar queries used in earlier INEXes, we also propose a method of automatic detection and removal of negation in queries. Retrieval with the queries after automatic removal of negation yields *statistically* equivalent performance compared to the manual method.

[4] Text REtrieval conference (http://trec.nist.gov)

1.3.2 Evaluation

On the evaluation of XML retrieval, we study in detail sensitivity & robustness of various evaluation metrics that are presently being used at INEX Ad Hoc track. We also do an analysis on the pooling methodology used in evaluation forums with some proposals. Our contributions in evaluation are summarized below:

- **INEX Evaluation:** *A study on Reliability and Reusability*

  Since 2007, INEX has been using a set of precision-recall based metrics for its adhoc tasks. Our aim is to investigate the reliability and robustness of these retrieval measures, and of the INEX pooling method. We investigate four specific questions. i) How reliable are the metrics when assessments are incomplete, or when query sets are small? ii) What is the minimum pool/query-set size that can be used to reliably evaluate systems? iii) Can the INEX collections be used to fairly evaluate new systems that did not participate in the pooling process? And, iv) for a fixed amount of assessment effort, would this effort be better spent in thoroughly judging a few queries, or in judging many queries relatively superficially?

  Our findings validate properties of precision-recall-based metrics observed in document retrieval settings. Early precision measures are found to be more error-prone and less stable under incomplete judgments and small topic-set sizes. We also find that system rankings remain largely unaffected even when assessment effort is substantially (but systematically) reduced. We confirm that the INEX collections remain usable when evaluating non-participating systems. Finally, we observe that for a fixed amount of effort, judging shallow pools for many queries is better than judging deep pools for a smaller set of queries. However, when judging only a random sample of a pool, it is better to completely judge fewer topics than to partially judge many topics. This result confirms the effectiveness of pooling methods.

- **Cost-effective Reusable Pooling:**

  Evaluation of Information Retrieval (IR) systems at standard forums like TREC, CLEF[5],

[5] Conference and Labs of the Evaluation Forum (http://www.clef-initiative.eu/)

NTCIR[6], INEX or FIRE[7] is based on Cranfield paradigm where a test collection is built with three major components: 1) a set of documents (*corpus*), 2) a set of information need (*topics*) and 3) a set of relevance judgments for each topic (*qrels*). Ideally these qrels should be complete; i.e. each document in the corpus should be judged as relevant or non-relevant with respect to each topic in the topic set. For a large corpus it is infeasible to construct such test collection because of prohibitive amount of time and effort involved therein. A more practical and efficient approach called *pooling* is used where top-k (generally $k = 100$) documents from the ranked output of participating systems are chosen for relevance judgment with respect to each topic. The documents judged relevant within the chosen set are assumed to be the only relevant documents for the query and the vast majority of unjudged documents outside the chosen set are considered non-relevant.

Although this pooling has been proven effective, with the growing size of corpora, it has become immensely resource-intensive in terms of manpower and time. Efforts have, therefore, been directed to make the entire evaluation process less time-consuming. Some of these low-cost evaluation techniques have either compromised on *unbiasedness* or *completeness* of pooling [25, 13, 17] or completely eliminated the pooling process [2]. However, the point to be noted is that pooling is not only used for evaluation of retrieval systems but for the diagnosis and tuning of the systems as well. To troubleshoot why a system is not able to retrieve relevant documents in the early ranks, one needs the knowledge of relevant documents for the topic that a pool provides.

We demonstrate a simple and pragmatic approach for the creation of smaller pools for evaluation of adhoc retrieval systems. Instead of using an apriori-fixed depth, variable pool-depth based pooling is adopted. The pool for each topic is incrementally built and judged interactively. When no new relevant document is found for a reasonably long run of pool-depths, pooling can be stopped for the topic. Based on available effort and required performance level, the proposed approach can be adjusted for optimality. The proposed technique serves two purposes:

[6] NII Test Collection for IR Systems (http://research.nii.ac.jp/ntcir/index-en.html)

[7] Forum for Information Retrieval Evaluation (http://www.isical.ac.in/∼clia/)

  - it reduces the total assessment effort without compromising on the reusability of the pool and
  - it offers the possibility of having a better estimate of *recall* per query.

The majority of the recent low-cost evaluation strategies ([2, 25, 13, 17]) lead to loss of information about the actual set of relevant documents. As discussed above, this information is particularly useful for post-hoc fault analysis that is intended to improve system performance. For our approach, this problem is much less severe. Experiments on TREC-7, TREC-8 and NTCIR-5 data validate its efficacy in a document retrieval setting as well.

1.4 Organization

The rest of the thesis is organized as follows:

- Chapter 2 builds the background and discusses related work. It summarizes some of the seminal work done in XML retrieval and provides a chronological tour of the different evaluation measures used so far for the INEX adhoc task.
  (Publication [90])

- Chapter 3 elaborates our retrieval endeavour as part of our participation in INEX adhoc task. It discusses our experiments on length normalization and parameter tuning and defines a baseline retrieval strategy.
  (Publications [95, 89, 92])

- In chapter 4, we take up the issue of query refinement. This chapter describes how negation removal can provide improvements over the baseline retrieval performance.
  (Publications [96, 36])

- Chapters 5, 6, 7 and 8 together present our work on the evaluation of XML retrieval. In chapter 5, we explore *Pool Sampling* from two angles. Keeping the number of queries unchanged, we observe the behaviour of evaluation metrics when the pool is reduced first *randomly* and

then *systematically* per query.
(Publications [91, 93])

- Chapter 6 (*Query Sampling*) describes experiments that progressively reduce the query-set size while keeping the set of assessments per query unchanged. Once again, the behavior of evaluation metrics is studied using two different approaches.
  (Publications [91, 93])

- Chapter 7 analyzes the issue of reusability of the INEX collections. We simulate the evaluation of a ‘new’ system which does not contribute at the pooling stage.
  (Publication [93])

- Chapter 8 discusses a simple low-cost pool optimization technique.
  (Publication [94])

- Chapter 9 concludes by outlining direction for future work.

Chapter 2

Background

This chapter sets the background for the rest of the thesis. We start with a brief history of the genesis and evolution of the research on XML retrieval and its evaluation. Though both the database community and the IR community explored XML retrieval from two different angles, this chapter concentrates on the endeavours of the IR community. It also describes INEX and provides a chronological tour of the different evaluation measures used so far for the INEX Ad Hoc task.

2.1 Introduction

Though XML retrieval research has gained much importance and momentum of late, its history dates back to the early 90s. Fuhr [33] proposed the idea of introducing vague queries and imprecise information into traditional databases and offering ranked results to a database query. Instead of using the usual two-valued logic to decide about the inclusion of a tuple in the result set, Fuhr described a probabilistic approach to structured data retrieval, and thus brought information retrieval techniques to the database domain. Fuhr's work, however, was confined to structured data (relational databases), and did not consider textual data, which are predominantly unstructured or semi-structured in nature.

Navarro et al. [87, 88] presented a model for querying textual databases by both content and structure, which they claimed was quite expressive and efficiently implementable. The query language was formulated as an operational algebra and was not meant to be accessed by end-users. A query expression in this model can be viewed as a syntax tree, in which each non-leaf node represents an operator, and its subtrees represent its operands. These operators take sets of nodes and return a set of nodes. Leaf nodes correspond to *elementary* queries. Elementary content-based queries look for segments matching words or regular expressions, while the simplest structural queries either look for textual nodes by name, or specify constraints based on inclusion, distance, etc. The answer to a query is a set of nodes (structural elements in the text database) which are obtained by processing the syntax tree in a bottom-up manner. However, the model does not support relevance ranking.

Luk et al. [74] surveyed commercial systems or prototypes for XML retrieval that were existing upto the late 90s. These were categorized into DB-oriented systems, IR-oriented systems and Hybrid (DB + IR) systems. However the survey was sketchy and did not discuss details of any retrieval technique.

A second survey by Luk et al. [75] addressed this issue, and can be considered as a good starting point for studying indexing and searching techniques for XML documents. The authors provided an elaborate discussion about the various indexing and searching techniques and retrieval models proposed or implemented and practised upto the late 90s.

Most of these early studies were from a predominantly database perspective, and focused on adding IR features (e.g. similarity calculation and/or ranked retrieval) to database systems. The first conscious attempt to bring the XML and IR communities together was made at SIGIR 2000: a workshop on XML and IR [15] provided a platform to discuss relevant issues in these two disciplines, to define tasks and propose future directions for research. The workshop opened with a survey of the-then state-of-the-art in XML retrieval, consisted of three technical sessions on query languages, retrieval algorithms and IR systems respectively, and concluded with a panel discussion. The workshop highlighted a genuine need to create an XML retrieval evaluation framework consisting of benchmark XML data and evaluation metrics.

The workshop also culminated in a special issue of the Journal of the American Society for Information Science and Technology [3], which includes extended versions of the SIGIR2000 workshop papers.

A second edition of the “XML and Information Retrieval” workshop was held in conjunction with SIGIR 2002. Papers presented there proposed modifications of the standard Vector Space Model, as well as some new models for XML retrieval [4].

Since the initiation of INEX in 2002, most research efforts in the area of XML retrieval have centred around INEX. Participants have tried several retrieval models like the Vector Space Model (VSM), Probabilistic models, Logistic Regression (LR) model, Language Modelling (LM) techniques, Machine Learning (ML) techniques and also fusion of two or more techniques. There have also been a number of different attempts in the evaluation of XML-IR at INEX.

In the following sections, we elaborate on the experimental framework provided by INEX for XML retrieval. Then we discuss on the XML query languages that are used for sub-document level retrieval. VSM as a retrieval model and its use in XML retrieval is introduced next. This is followed by a discussion on evaluation efforts in XML retrieval.

2.2 INEX : INitiative for the Evaluation of XML Retrieval

The growing activity on XML eventually led to the formation of INEX in 2002 to study the issues specific to XML retrieval. This is a forum in the lines of TREC where participating researchers can discuss and evaluate their retrieval techniques using reasonably large test collections and uniform scoring procedures. Various XML retrieval tasks, each corresponding to a “track”, have been studied at INEX. Table 2.1 presents an overview of the tracks that have been offered over the years since then at INEX. The objectives of the various tracks are briefly described below.

Ad Hoc track. The adhoc task is intended to model arguably the most common information seeking behavior, in which a user submits a query (representing a one-time or casual information need) to a system, which then tries to retrieve information items (at the appropriate granularity) from within a text collection that are relevant to the user’s information need.

Relevance feedback. This track investigates relevance feedback methods for XML retrieval in

Table 2.1: **Summary of INEX tasks, 2002–2011**

<table>
<tr><th>2002</th><th>2003</th><th>2004</th><th>2005</th><th>2006</th><th>2007</th><th>2008</th><th>2009</th><th>2010</th><th>2011</th></tr>
<tr><td colspan="9">Ad Hoc</td><td></td></tr>
<tr><td></td><td></td><td colspan="3">Rel. feedback</td><td></td><td></td><td></td><td colspan="2">Rel. feedback</td></tr>
<tr><td></td><td></td><td colspan="3">Heterogeneous</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="3">NLP</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="8">Interactive</td></tr>
<tr><td></td><td></td><td></td><td colspan="6">Doc. mining</td><td></td></tr>
<tr><td></td><td></td><td></td><td colspan="3">Multimedia</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>Use case</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="4">Entity ranking</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="4">Link the Wiki</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="5">Book search</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">Efficiency</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">QA</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">Data-centric</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">Web Service Discovery</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Snippet Ret.</td></tr>
</table>

general, and in particular, query reformulation approaches that can potentially use both content and structural information.

Heterogeneous track. Real-life collections of XML documents are likely to contain documents that are diverse with respect to syntax (i.e. the documents are based on different DTDs), semantics (i.e. the collection covers a variety of topics), and genre. This track explores how XML retrieval systems handle such heterogeneous collections.

Natural Language Processing (NLP). The aim of this track was to study how natural language formulations of structural conditions can be handled by systems.

Interactive track. The main aim of this track was to study the behavior of searchers when interacting with components of XML documents.

Document mining track. This track looks at techniques for classifying and clustering XML documents.

Multimedia track. The main objective of the Multimedia track was to investigate retrieval from repositories containing not only text, but also other types of media, such as images, speech, and video.

Use case track. The aim of this track was to identify the potential users, scenarios and use-cases of XML retrieval systems.

Entity Ranking. The goal of the Entity Ranking track was to examine techniques for list completion and associative ranking. Given a list of entities, systems would have to either extract other entities of the same kind from a document collection, or construct similar lists, but on a different topic.

Link the Wiki. This track looks at algorithms for performing automated link discovery in XML documents from Te Ara Encyclopedia collection.

Book Search. This track focuses on retrieval from book collections. Its goal includes the investigation of book-specific relevance ranking strategies, user interface issues and user behavior, exploiting special features, such as back-of-the-book indexes provided by authors, and linking to associated metadata like catalogue information from libraries.

Efficiency. This track closely follows the Ad Hoc track setting but evaluates the effectiveness and efficiency of XML ranked retrieval approaches on real data and real queries with high dimensional structural constraints. One task was to study scalability issue of any ranked retrieval engine within a time-budget.

Question Answering. This track investigates methods for accessing structured documents that can be used to address real-world focused information needs formulated as natural language questions.

Data-centric. This track provides an evaluation framework for retrieval from data-oriented XML corpora like the IMDB collection, which are rich in structure and do not contain long text-fields.

Web Service Discovery. Web service constitutes an important building block in many computing paradigms like Pervasive Computing, Service-Oriented Computing, Cloud Computing etc.

This track aims to investigate efficient and effective Web services discovery mechanism based on searching service descriptions provided in Web Services Description Language (WSDL)[1].

Snippet Retrieval. The goal of the snippet retrieval track is to determine how best to generate informative snippets for search results so that these snippets provide sufficient information to allow the user to determine the relevance of each document, without needing to view the document itself [2].

2.3 Query Languages

Though there are quite a few database query languages for data-centric XML documents, when viewed from an IR point of view, the problem of retrieving useful XML elements from XML documents demands a new paradigm for query languages. For INEX 2002, two methods for formulating queries were proposed [56]: Content-Only (CO) queries, and Content-And-Structure (CAS) queries. Both types of queries are marked up in XML, but while CO queries are very similar to TREC queries and do not include any structural information or constraints, CAS queries contain either explicit or implicit structural constraints specified using XPath [144]-like notation. Specifically, the title field of CAS queries has the following format:

```
<!ELEMENT Title ( te?, (cw, ce?)+ )>
```

Thus, a title may specify a target element (*te*) to be retrieved, in addition to a set of search terms (*cw*). It may also provide additional constraints on the context elements (*ce*) in which particular search terms should occur.

Though the CO query language proved to be adequate, the CAS 2002 query language was found to have certain deficiencies [139]. For INEX 2003, the problem was temporarily solved using an extended version of XPath [57], but eventually this extended XPath notation also proved too complex to use correctly. Indeed, 63% of the distributed queries had major semantic or syntactic errors. The INEX query working group outlined a list of changes [127] based on which the *Narrowed Extended*

[1] http://www.inex.otago.ac.nz/tracks/webservices/webservices.asp

[2] https://inex.mmci.uni-saarland.de/tracks/snippet/

XPath I or NEXI [138] query language was formulated. NEXI is essentially a much-simplified version of XPath that includes an added *about* clause. An example of a structural constraint contained in a NEXI CAS query is given below:

```
 //article[(.//fm/yr  = 2000  OR .//fm/yr =  1999) AND
about(.,   "intelligent    transportation   system")]   //sec[about(.,
automation +vehicle)]
```

This constraint requires systems to return `sec` elements about vehicle automation from documents about intelligent transportation systems that were published in 1999 or 2000.

NEXI CAS-queries containing structural requirements can be interpreted in two ways:

- **strictly** — such queries, called SCAS queries, contain a target element specification that *has to be met* for retrieval;
- **vaguely** — such queries, called VCAS queries, contain an optional target path specification, that only serves as a hint and need not necessarily be satisfied for retrieval.

NEXI, which was used in INEX 2004 to simplify the query formation process from a user's information need, reduced the error-rate to 12% from 63% for the adhoc task CAS queries.

Besides NEXI, a new query language named XXL (FleXible XML Search Language) has also been proposed [124]. XXL is based on XQuery [20], but includes a similarity operator ($\sim$) on element names and contents.

Kamps et al. [53] provide a detailed study on the CO and CAS queries posed by users and the elements returned by different systems in response to those queries. They observe that searchers use the additional expressive power of structural constraints primarily as search hints, rather than strict requirements. While three-fourths of the queries use constraints (which could not be expressed by keyword-only CO queries) on the context of the elements to be returned, only one third of the examined queries use the hierarchical structure of the documents. In general, these structured queries act as precision-enhancing devices, having a positive effect on early precision at low recall values but a negative effect on overall recall. This suggests that structured queries can be a powerful tool for the searcher who is interested solely in the precision of the first handful of results. The authors also observe that a query language with low expressivity is safer, as it reduces the chance of making semantic mistakes.

2.4 The Vector Space Model

The vector space model [116] is one of the most commonly used models in the field of IR. It was developed by Salton and his students in the 1960s and the early 1970s. Under this model any given text — an article (or a portion of an article), a query, etc. — is represented as a list of *terms* or *keywords* with associated *weights*. A term is usually a word or a phrase and the weight corresponding to a term is a measure of its importance in representing the information in the given text.

If the size of the vocabulary of a text collection (i.e. the total number of distinct terms in all the documents of the collection) is T, then, in this model, a text D_i can be represented as a vector in a T-dimensional vector space:

$$D_i = (d_{i1}, d_{i2}, \ldots, d_{iT})$$

where d_{ik} is the weight of term t_k in document D_i. A weight of zero is used for terms that are absent from a particular document, and positive weights characterize terms contained in a document. Note that most text vectors will be sparse since any text uses a very small subset of the entire vocabulary.

2.4.1 Retrieval

The relatedness of two pieces of text — a query and a document, for example — can now be estimated by measuring the proximity of the corresponding vectors: when two vectors are “close”, the corresponding texts are expected to be semantically related. In the vector space model, the closeness of two vectors can be measured using the vector inner product. The relatedness of two texts $D = (d_1, d_2, \ldots, d_T)$ and $Q = (q_1, q_2, \ldots, q_T)$, called the *similarity*, can therefore be calculated as follows:

$$\mathit{Sim}(D, Q) = D \cdot Q = \sum_{i=1}^{T} \sum_{j=1}^{T} d_i \times q_j \times (\vec{u_i} \cdot \vec{u_j})$$

where $\vec{u_i}$ and $\vec{u_j}$ are the unit vectors in dimensions i and j.

For simplicity, we make the (obviously false) term independence assumption — any two terms $t_i, t_j (i \neq j)$ are taken to be unrelated, or equivalently, the corresponding dimensions $\vec{u_i}, \vec{u_j}$ are considered to be orthogonal. Thus, under this assumption, if D contains the word "cigarette" (but not "tobacco"), and Q contains the word "tobacco" (but not "cigarette"), the presence of these words do not contribute to the similarity (or relatedness) of D and Q, even though it is obvious that these two words are related[3].

Using $\vec{u_i} \cdot \vec{u_j} = 0$ whenever $i \neq j$, the formula above reduces to:

$$\mathit{Sim}(D, Q) = \sum_{i=1}^{T} d_i \times q_i$$

This formulation has the following desirable properties:

- The similarity between two articles increases with the number of matching terms (terms contained in both articles). This agrees with the intuition that if two texts have a number of keywords in common, they are likely to be strongly related.

- The similarity increases with the importance of matching terms. Once again, this agrees with the intuition that if the matching terms are important in either text, then the vectors should be considered closer than if the matching terms are not important in the texts. For example, a match on the word "telecommunications" is more indicative of a semantic relation than a match on the word "the".

Given a natural language user-query, we construct its vector representation $Q = (q_1, \ldots, q_T)$ in much the same fashion, and a numeric estimate of the potential usefulness of document D_i is

[3] This simplifying assumption is commonly made in modern IR systems to avoid the complexities that arise from using inter-word correlations [157]. Alternate techniques (like query expansion) are used to extract some of the benefits that we would get from a detailed knowledge of inter-word relationships.

obtained by computing the similarity between the query vector and the vector for D_i

$$Sim(Q, D_i) = \sum_{j=1}^{T} q_j \times d_{ij}$$

Documents in the information base are then ranked in decreasing order of their similarity to the query and the highest ranked documents (these are expected to be the most useful) are presented to the user [120, 117].

2.4.2 Indexing

We now turn to the process of *indexing*, the method by which terms are assigned to a document and weights are computed for the assigned terms.

Term Assignment. The list of terms assigned to a text is typically obtained using the following steps [118]:

1. *Tokenization*: The text is first broken into individual words, punctuation marks, and other tokens.

2. *Stopword removal*: Common function words like *the*, *of*, *an*, etc. (also called stopwords) are removed from the list of words obtained above.

3. *Stemming*: Morphological variants of a word are normalized to the same *stem* [72]. Usually simple rules for suffix removal are used in this process. For example, “believing” is converted to “belief” by first removing the suffix “ing”, and then converting the terminal ‘v’ to ‘f’. Similarly, “believes” is also converted to “belief”.

4. *Phrase recognition*: Multi-word phrases (e.g. “information retrieval”, “computer science”) are recognized using statistical or syntactic techniques and are used in addition to the list of single words to index the text. [32]

Term Weights. With a simple inner product similarity computation formula, the quality of document ranking is crucially dependent upon the assignment of proper weights to terms in the texts (documents and queries). Judicious assignment of term weights can substantially improve the ranking effectiveness of a system [119, 10]. Most modern IR systems automatically assign weights to the terms in a text using the following three factors:

1. **Term Frequency (tf)**, the number of occurrences of a term within a document. Articles that repeatedly use a query term are potentially more useful than articles that rarely use that query term [73]. Therefore, the weight of a term should be an increasing function of its tf.

2. **Inverse Document Frequency (idf)**, an inverse measure of the number of documents in the collection in which a term occurs. Common words, i.e. words that are used in numerous different articles, are less important than words that are used in a few articles [132, 133, 121, 104]. For example, a match on an uncommon word like “angiogram” should contribute more towards a document’s predicted usefulness compared to a match on a more common word like “medical”. Thus, the weight of a term should be an inverse function of its document frequency.

3. **Document Length**. Long documents often tend to repeat terms and thus, in general, have higher term frequencies. Long documents also use numerous different words. Thus the number of matches between a query and a long document tends to be high. For these reasons, long documents can get a preference in retrieval over short documents just because of their length, and not necessarily because they are more informative for the user. Therefore, document term weights should be scaled down by some measure of the document’s length. This is called *document length normalization* [128, 103].

Using these factors, we can define the weight of a term in a document as:

$$\frac{\textit{tf-factor} \times \textit{idf-factor}}{\textit{document length}}$$

2.4.3 VSM in XML-IR

VSM being one of the most popular and effective IR models for unstructured document retrieval, quite a few attempts have been made to use VSM in the field of XML-IR. One of the earliest among them was by Schlieder et al. [125] where they applied VSM and tree matching techniques. Their structured term or *s-term* ranking model considered both the documents and queries as labelled trees. Every document component rooted at a node with the same label as the query root is a potential candidate to be returned as a result. However, retrieval was done only at the whole document level. Weigel et al. [149] extended the *s-term* ranking model with additional data structures and algorithms to support more effective as well as more efficient XML retrieval.

Grabs and Schek [44, 4] observed that since XML has a heterogeneous form, queries might have very different scopes. Term statistics should, therefore, reflect the scope of the query. The authors presented a flexible IR model for single-category, multi-category and nested XML retrieval where the basic index and statistics data are integrated on the fly at retrieval time depending on the scope of the query.

Carmel et al. [16] extended the vector space model to XML retrieval. Within the expression for *content-based similarity* between queries and document, they introduced a factor called context resemblance based on *structural similarity*. They also proposed several heuristic functions, which proved effective on the INEX 2002 test collection.

In the follow-up work, Mass et al. ([78], [79]) stored tf-idf statistics at the XML component level instead of the document level. They created separate indices for full articles, sections, paragraphs etc. Each component was then separately ranked on the basis of a similarity score computed using the appropriate tf-idf statistics. However the component level indices ignore the context provided by the document in which a component occurs. A document pivot factor (*DocPivot*) ([126], [128]) was therefore incorporated in [79] to scale final element scores by the containing article score. The final score of a component C with original score S_c and its containing article score S_a is then given by

$$DocPivot * S_a + (1 - DocPivot) * S_c.$$

Crouch [26] criticised the idea of separately indexing each component-type as this consumes a lot of physical storage [26]. The author chose 18 subvectors to represent a XML document. Among them, 8 subvectors having content-bearing terms from the nodes like *article title, abstract, keywords, body* etc. were used for indexing. Only the leaf-nodes of the XML-tree were indexed and vector representation of a parent node was generated by merging the vectors of its children. The other 10 subvectors which are structure-based serve as filters on the result set returned by CAS queries. The paper showed that this leaf-level indexing gives results comparable to separately indexing all elements but with a much reduced (upto 50% less) index-size.

Hassler et al. [46] elegantly incorporated the vector space model in their user-driven XML retrieval system. From the query they segregated structural constraints and content part. Similarity was measured separately for structure and content and final RSV was calculated as a linear combination of the two similarity-scores.

Hubert [49] came up with an adapted VSM. In this model, the similarity score is determined by a combination of three functions:

$$Score(T, E) = (\sum_{\forall t \in T} f(t, E).g(t, T)).p(T, E)$$

$f(t, E)$ measuring the contribution of term t in element E; $g(t, T)$ measuring the contribution of t in the topic T; function $p(T, E)$ measuring the global presence of topic T in element E. $f(t, E)$ is taken as the ratio of frequency of term t in E to the frequency of all query-terms in E. $g(t, T)$ is a function of frequency of term t in T, term-frequency of all the terms in T, number of elements containing term t and rank of t according to the number of elements containing term t. Factor $p(T, E)$ is an exponential function of number of terms in topic T and number of query-terms appearing in E as well. Though this score is calculated at each element, the score is propagated upwards the hierarchical structure of XML tree. The propagated score is gradually decreased by a tunable reducing factor.

Lehtonen [68] in his EXTIRP system completely ignored information coded in tags for CO queries. Two indices are built, one for words and the other for phrases. The entire document collection is divided into disjoint fragments which are independently treated as full-documents under the traditional tf-idf based VSM.

Huang et al. [48] used VSM to calculate the similarity between queries and elements using both full-text and a compact representation (that retains only important terms and strips off all structural and formatting tags).

Tanioka [135] within his *preferential unification system* of XML elements, used a simplified VSM for article and element retrieval. Similarly, Verbyst and Mulhem [140] used VSM to compute content-similarity between elements.

Besides the adhoc task, VSM has been and is still being widely used in other INEX tasks like the interactive track [31], document mining [67], [145], Question Answering track [71] and so on.

2.5 Evaluation Metrics

The two most frequent and basic measures for information retrieval effectiveness are *precision* and *recall*. These are defined for the simple case where an IR system returns a set of documents for a query [76].

Precision (P) is the fraction of retrieved documents that are relevant.

$$\text{Precision} = \frac{\#\text{(relevant items retrieved)}}{\#\text{(retrieved items)}} \tag{2.1}$$

Recall (R) is the fraction of relevant documents that are retrieved.

$$\text{Recall} = \frac{\#\text{(relevant items retrieved)}}{\#\text{(relevant items)}} \tag{2.2}$$

The above definition of recall and precision are based on the notion that there is a retrieved set of documents and a non-retrieved set of documents. In ranked retrieval, however, there is no such clear demarcation. All articles are simply ranked by their predicted goodness. One has to evaluate the quality of the ranking in the entire collection. If both precision and recall are calculated for each query whenever a relevant article is found and these values are plotted, the curve demonstrates the performance of the system.

Generally, however, it is more useful to represent the performance of a system by a single number. *Average Precision (AP)* is such a metric that is commonly used to measure the goodness of ranking for a system. For a single topic, AP is the mean over all relevant documents of the precision computed at each rank at which a relevant document is retrieved, using 0 as the precision for relevant documents that are not retrieved. This is equivalent to the area under the curve of the non-interpolated recall-precision graph for the topic.

Interpolated Average Precision is another metric that is often used to measure overall retrieval effectiveness. For a query, $P_{\text{int}}[x]$, the *interpolated* precision at some recall level $x \in [0, 1]$ is defined as the highest precision found for any recall level $r \geq x$. Thus,

$$P_{\text{int}}[x] = \max_{r \geq x} \quad (P[r])$$

where $P[r]$ denotes the precision calculated when a system first achieves a recall level of r for the given query. Generally, interpolated precision is calculated for 0%, 10%, . . ., 100% recall levels. The average of these 11 values is termed the 11-point Average Precision. In other words, for a topic t, 11-point (interpolated) *AP* is given by:

$$\text{11-point } AP(t) = \frac{1}{11} \sum_{x \in \{0.0, 0.1, \ldots, 1.0\}} P_{\text{int}}[x](t) \tag{2.3}$$

To get a single number to represent the overall performance of a system over a set of n topics, the scores across all the topics in the given set are usually averaged. Thus, *Mean (non-interpolated) Average Precision (MAP)*, a widely used metric, is defined as the arithmetic mean of (non-interpolated) AP over all topics.

$$MAP = \frac{1}{n} \sum_{t=1}^{n} AP(t). \tag{2.4}$$

Similarly, the average interpolated precision of a system at recall x is given by:

$$\text{Avg. } P_{\text{int}}[x] = \frac{1}{n} \sum_{t=1}^{n} P_{\text{int}}[x](t) \tag{2.5}$$

In the same way we can define mean 11-point AP. It is known that 11-point AP is less sensitive to changes in the top ranks than non-interpolated AP. In the rest of this thesis, we have generally used MAP (Eq. 2.4) — as implemented in the `trec_eval` script — to measure the performance of a document retrieval system.

2.6 Metrics for XML retrieval

The basic goal of XML-IR is, like traditional IR, to form a ranked list of XML elements ordered by their relevance to the query. However, unlike traditional IR, where whole documents are returned, XML-IR aims to retrieve only the relevant portions of documents. The retrieved document-parts or elements can be of varying granularity and may contain overlap among themselves. Traditional document-level metrics such as precision and recall are thus not enough to effectively evaluate XML retrieval. Indeed, formulating a good evaluation framework for XML retrieval has itself been a challenging problem, and the evaluation metrics used at INEX have kept changing over the years. We summarize the evolution of the INEX evaluation methodology below. For a nice summary of the metrics used at INEX till 2005, see [55].

2.6.1 INEX 2002

In the initial years of INEX, evaluation was based on the following task definition for XML retrieval systems [64, 63]: "the general task of an IR engine has been defined in INEX as the task of returning, instead of whole documents, those components (XML elements) that are most *specific* and *exhaustive*". The *exhaustivity* (E) of an element is defined as the extent to which the element covers the information need expressed by the topic or query, while its *specificity* (S) measures how much of the element focuses on the given topic (i.e. whether or not it discusses other irrelevant topics).

E and S for an XML-element were both quantified using the following 4-point scale: high (3), fair (2), marginal (1) and not exhaustive/specific (0) [41]. Out of the 16 possible combinations of (E, S) scores that can be assigned to an element, 10 combinations are meaningful. These are given by:

$$\{(0,0),(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3)\}$$

To compute various single-valued metrics that may be used to compare systems, pairs of (E, S)

values were mapped to a single measure of relevance by a quantization function

$$f_{quant} : E \times S \ \rightarrow \ [0,1]$$

Two variants of $f(e,s)$ were used:

$$f_{strict}(e,s) = \begin{cases} 1 & \text{if } (e,s) = (3,3) \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

$$f_{SOG}(e,s) = \begin{cases} 1.00 & \text{if } (e,s) = (3,3) \\ 0.90 & \text{if } (e,s) = (2,3) \\ 0.75 & \text{if } (e,s) \in \{(1,3),(3,2)\} \\ 0.50 & \text{if } (e,s) = (2,2) \\ 0.25 & \text{if } (e,s) \in \{(1,2,(3,1)\} \\ 0.10 & \text{if } (e,s) \in \{(2,1),(1,1)\} \\ 0.00 & \text{if } (e,s) = (0,0). \end{cases} \tag{2.7}$$

Since each document component in a ranked list is assigned a single relevance value, and overlap among components is ignored, methods for calculating recall / precision curves for document retrieval can be easily applied to the values assigned by the quantization functions using an approach suggested by Raghavan et al. [102]. In this approach, the precision at an arbitrary recall value $x \in [0,1]$ is interpreted as the probability $P(rel \mid retr)$ that a retrieved document component is relevant:

$$P(rel|retr)(x) = \frac{x.n}{x.n + esl_{x.n}} = \frac{x.n}{x.n + j + \frac{s.i}{r+1}} \tag{2.8}$$

Here, n is the total number of relevant document-components in the collection with regard to the user request; $esl_{x.n}$ denotes the *expected search length* [24], i.e. the estimated number of non-relevant elements retrieved until the recall point x is reached i.e., the $x.n$-th relevant element is retrieved. The paper [102] also considered weak ordering of retrieved items where two or more items may be retrieved at the same rank. Let l denote the rank at which the *x.n*-th relevant component is retrieved. Then, in the above equation, j is the number of non-relevant document components retrieved before rank l, s is the number of relevant components to be taken from rank l, and r and i are the numbers of relevant and non-relevant components at rank l, respectively.

2.6.2 INEX 2003

The INEX 2002 measures did not take into account the overlap between retrieved elements and considered descendant and ancestor elements independently of each other. The INEX 2003 metrics (computed by the `inex_eval_ng` program) address the issue by incorporating component size and overlap within the definition of recall and precision. These metrics are based on the notion of an ideal concept space (proposed by Wong et al. [151]) and are formulated as follows (for derivation, see [42]):

$$recall = \frac{\sum_{i=1}^{k} e(c_i).\frac{|c'_i|}{|c_i|}}{\sum_{i=1}^{N} e(c_i)} \tag{2.9}$$

$$precision = \frac{\sum_{i=1}^{k} s(c_i).|c'_i|}{\sum_{i=1}^{k} |c'_i|} \tag{2.10}$$

Here, $c_1, \ldots, c_k$ represent document components forming a ranked result list, N is the total number of components in the collection, $|c_i|$ denotes the size of component c_i, and $|c'_i|$ is the portion of the component that has not been seen by the user previously. If each c_i is represented as a set of (term, position) pairs, $|c'_i|$ can be calculated as:

$$\begin{aligned} |c'_i| &= |c_i - \bigcup_{c \in C[1,n-1]} (c)| && (2.11) \\ &= |c_i| - \sum_{c \in C[1,n-1]} |c| && (2.12) \end{aligned}$$

where n is the rank of c_i in the output list, and $C[1, n-1]$ is the set of elements retrieved up to (and not including) rank n. Finally, $e(c_i)$ and $s(c_i)$ denote respectively the quantized version of the exhaustivity (e_{raw}) and specificity (s_{raw}) values assigned to c_i by an assessor. The quantization operation was defined by Kazai et al. [58] as:

STRICT case:

$$e(c_i) = \begin{cases} 1 & \text{if } e_{raw} = 3 \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

$$s(c_i) = \begin{cases} 1 & \text{if } s_{raw} = 3 \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

GENERALIZED case:

$$e(c_i) = e_{raw}/3 \tag{2.15}$$

$$s(c_i) = s_{raw}/3 \tag{2.16}$$

2.6.3 INEX 2004

While it is acceptable to consider the two dimensions separately in the ideal concept space interpretation, the inex-2003 metric was criticised for separating exhaustivity and specificity since their combination is required to identify the most desirable retrieval components [58]. At INEX 2004, therefore, evaluation methodology went back to that of INEX 2002 with some more variations of quantization function than original strict and specificity-oriented-generalized (SOG) versions. Also, the overlap issue was addressed by introducing a set-based overlap indicator that characterises a run by the percentage of overlapping items in the submission result list R [143]:

$$\text{overlap-indicator} = \frac{|\{p \in R | \exists q \in R \wedge p \neq q \wedge \text{overlap}(p, q)\}|}{|R|} \tag{2.17}$$

Here, p and q are structure nodes, so they are either disjoint or in an ancestor-descendant relation. Accordingly, overlap(p, q) is either 0 or 1.

2.6.4 INEX 2005

A number of metrics were discussed at INEX 2003 by the working group on evaluation metrics [59].

XCG: XCG metrics are an extension of Kekäläinen et al.'s Cumulated Gain(CG)-based metrics [62] proposed by Kazai et al. for the XML domain [58]. A *gain vector G* is formed from a ranked list by replacing each document-id by its graded relevance value (integer score in {0, 1, 2, 3} where 0 denotes no relevance and 3 maximum relevance). Thus, $G(i)$ gives the relevance grade of the i-th ranking document. The cumulated gain (CG) vector is then defined as:

$$CG(i) = \begin{cases} G(i) & \text{for i = 1} \\ CG(i-1) + G(i) & \text{for i} \geq 2 \end{cases} \tag{2.18}$$

For each query, an ideal gain vector G' can be constructed by arranging all items in decreasing order of the relevance grades, and an ideal CG vector can be formed from G'. The normalized CG vector, nCG, is obtained when CG is divided by CG' co-ordinate wise, i.e. $nCG(V, I) = \langle v_1/i_1, v_2/i_2, \ldots, v_k/i_k \rangle$, where CG vector $V = \langle v_1, v_2, \ldots, v_k \rangle$, and ideal CG vector ($CG'$) is given by $I = \langle i_1, i_2, \ldots, i_k \rangle$.

Obviously, for any rank i, $nCG(i) = 1$ corresponds to ideal performance and, in general, the area between $nCG(V, I)$ and $nCG(I, I)$ represents the quality of an IR technique: the smaller the area the better.

The assessed generalized f_{SOG}-value is used as the relevance value in XCG. To mitigate the problem of overlap, Kazai et al. [58] built an ideal-recall base by recursively choosing the element with the highest f_{SOG}-value and removing the rest of the elements from each path. If two or more elements have the same f-value in a path then the one with greater height is chosen. While I is derived from the ideal recall-base, V is taken from the original recall-base containing overlap. For any partly seen component c_i, relevance-value $rv(c_i)$ is:

$$rv_{final}(c_i) = \alpha.\frac{\sum_{j=1}^{k} rv_{final}(c_j).|c_j|}{\sum_{j=1}^{k} |c_j|} + (1-\alpha) rv_{previous}(c_i) \tag{2.19}$$

where k is the number of c_i's child nodes denoted by c_j, $|.|$ denotes component size, α is the user's intolerance to previously seen component ($0 \leq \alpha \leq 1$, where 1 corresponds to no tolerance to previously viewed component), $rv_{previous}$ denotes the score independent of result list before considering the overlap factor.

Though there exists a general belief that system ranks would be significantly affected by how overlap is handled, Woodley et al. [153] experimented on the role of overlap with XCG metric using INEX 2004 data. They showed that high scoring systems with high overlap perform well regardless of how overlap is handled.

T2I: Since XML does not have a fixed, pre-defined retrieval unit, de Vries et al. [142] came up with evaluation metrics derived from a user-effort-oriented view of IR. User-effort is expressed in terms of the time wasted in inspecting irrelevant elements. The basic assumption here is that a user needs an entry-point into a document rather than a focused element. A retrieval system is thus supposed to produce a ranked list of entry-points. The user starts reading a retrieved article from an entry-point and continues reading until his/her *tolerance to irrelevance* (T2I) is reached, at which time the user moves to the next entry-point. T2I expressed by parameter τ_{NR} represents the maximum time a user spends reading irrelevant text. The authors proposed three T2I-based metrics:

1. Average precision after a fixed amount of user effort in accordance with Hull's proposal [50]. This is given by :

$$\text{Average Precision } = \frac{\tau_{NR}}{T}. \sum_{t=1}^{T/\tau_{NR}} \text{Precision after } t.\tau_{NR} \text{ seconds wasted user effort}$$

   with suitable choice of τ_{NR} and T (some multiple of τ_{NR}).

2. Precision after *expected search length* (ESL), a parameter defined by Cooper [24]. ESL is expressed here in terms of *expected search duration* (ESD) [30] as the sum/total of user-effort wasted inspecting irrelevant elements from the result list, and the effort to find the remaining relevant items by randomly searching through the collection.

3. Probability of relevance $P(rel|retr)$ for R relevant fragments computed in terms of ESL_R:

$$P_R(rel|retr) = \frac{R}{(R + ESL_R)}$$

For XML evaluation, the metrics could be defined by expressing the user effort in terms words or characters.

PRUM: Piwowarski et al. [99] proposed another metric PRUM (Precision Recall with User Modeling) where they extended the idea of Raghavan et al.'s probabilistic precision-recall to include users' browsing behavior [102]. In a simple traditional user model, a user treats each retrieved element independently and its relevance is considered in isolation. But PRUM considers each retrieved element as an entry point to the collection and allows the user to consult the ancestor, descendants and siblings of a node. Each of these context elements is seen by the user stochastically, $i.e.$ with some pre-assigned probability. Precision in PRUM is defined for a given query q as

$$Prec(i) = Prob(Lur|Retr, L = l, Q = q)$$

where

$$\begin{aligned} i &= \text{recall level lying between 0 and 1} \\ Retr &= \text{event that an element in the list is consulted} \\ l &= \text{percentage of relevant elements user wants to see} \\ q &= \text{the query topic} \\ Lur &= \text{event that the element consulted leads to an unseen relevant unit} \end{aligned}$$

The authors claimed that PRUM is designed for new IR domains like XML, Web, or video. It is a generalization of commonly accepted metrics as it reduces to standard precision-recall when browsing between elements is not allowed, and to one of T2I metrics following Raghavan et al.'s formula with proper setting of navigational probabilities.

Piwowarski [98] further extended PRUM, known as EPRUM, with an alternative definition of precision. He considered precision as the ratio of the minimum number of ranks that a user has to view in the list returned by an ideal system and the number of ranks user actually visits from the list returned by the system to be evaluated to cover the same number of relevant items. EPRUM is computed using three sets of parameters:

i. probability that a user consults an element in the corpus

ii. probability that a user browses from a considered element to any neighbouring element and

iii. probability that a user finds an ideal element.

The paper [98] presented how it was used to precisely evaluate submissions in INEX 2005 considering more realistic user navigation across elements.

Metrics at INEX 2005:

Among the five types of metrics discussed above, three XCG-based metrics were taken as the official INEX 2005 metrics used to measure retrieval effectiveness of submitted runs [61]:

Normalized xCG (or nXCG): This is exactly the same as $nxCG$ discussed above. Systems are compared at different cut-off values, $e.g.$ $nxCG[1]$ and $nxCG[100]$. We may take the average of $nxCG[i]$ scores upto a given rank as

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

ep/gr: It stands for effort-precision/gain-recall. The metric measures effort as the number of visited ranks that a user spends compared to the effort needed in an ideal system. $ep(r)$ is measured for cumulated gain value r as [See Fig 2.1] :

$$ep(r) = \frac{i_{ideal}}{i_{run}}$$

This ep is calculated at arbitrary gain-recall points where gain-recall is defined as the cumulated gain divided by total achievable cumulated gain:

$$gr[i] = \frac{xCG[i]}{xCI[n]},\ n \text{ being the total number of relevant elements}$$

x

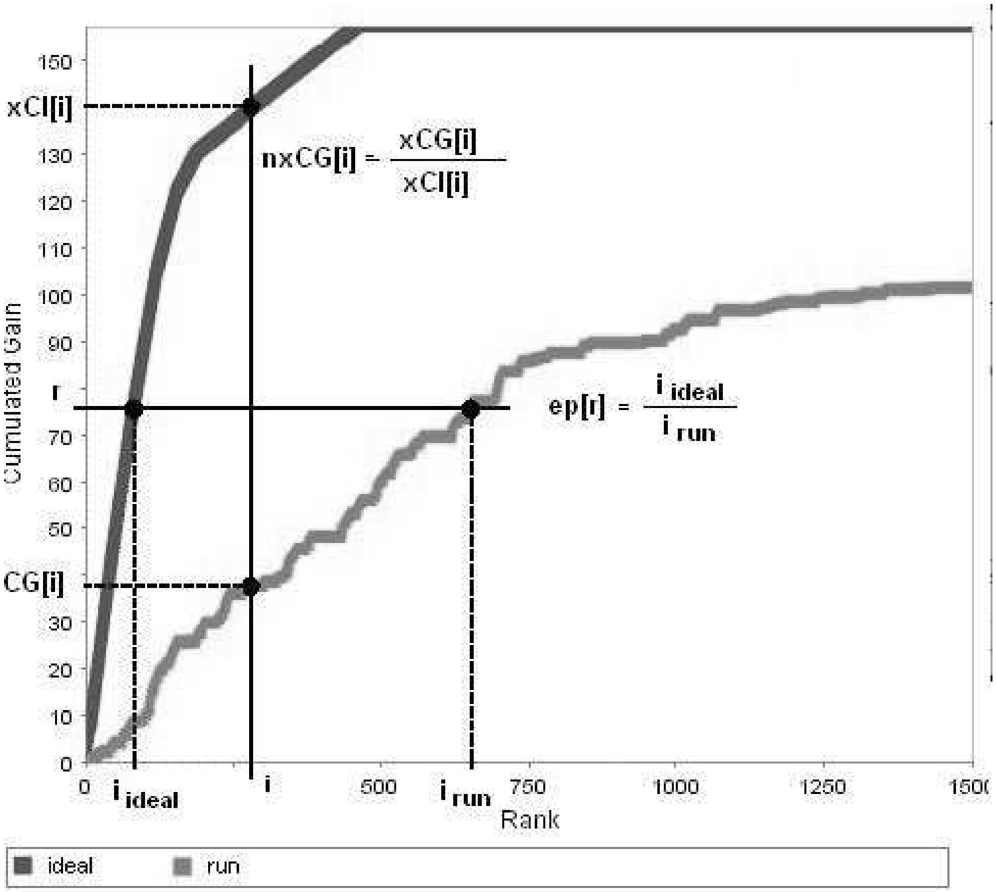


Figure 2.1: Cumulated Gain vs. Rank.

Q & R: $nxCG$ has the disadvantages of not averaging well across the topics. As a remedy, Sakai [108] suggests the Q and R measures:

$$Q = \frac{1}{Num_R}.\sum_{j=1}^{i} isrel(d_j).\frac{cbg(j)}{cig(j)+j} \tag{2.20}$$

$$\text{and} \quad R = \frac{cbg(Num_R)}{cbg(Num_R)+Num_R} \tag{2.21}$$

where

$$\begin{aligned}
Num_R &= \text{ total number of relevant elements} \\
isrel(.) &= \text{boolean function, returning } 1 \text{ for relevant element and } 0 \text{ otherwise} \\
cbg(i) &= \text{ cumulated bonus gain function} \\
&= \; bg(i) + cbg(i-1) \\
bg(i) &= xG(i) + 1 \text{ if } g(i) > 0 \\
&= 0, \text{ otherwise} \\
cig(.) &= \text{ideal cumulated bonus gain vector}
\end{aligned}$$

Pehcevski et al. [97] proposed an alternative XML retrieval evaluation metric solely based on the highlighted text which are generally ignored during evaluation for being ‘too small’ elements. Their metric ‘HiXEval’ credits systems that retrieve elements containing more highlighted textual information without containing non-relevant information. They defined two functions for an element e at a given rank r as:

$$\begin{aligned}
pre_r(e) &= \frac{rsize(e)}{size(e)}, \quad \text{if e is NOT-YET-SEEN} \\
&= \frac{(1-\alpha).rsize(e)}{size(e)}, \quad \text{if } e \text{ is FULLY-SEEN} \\
&= \frac{\alpha(rsize(e) - rsize(e'))}{size(e)} + (1-\alpha).\frac{rsize(e)}{size(e)}, \quad \text{if } e \text{ is PARTIALLY-SEEN} \\
rec_r(e) &= rsize(e), \quad \text{if } e \text{ is NOT-YET-SEEN} \\
&= (1-\alpha).rsize(e), \quad \text{if } e \text{ is FULLY-SEEN} \\
&= \alpha(rsize(e) - rsize(e')) + (1-\alpha).rsize(e), \quad \text{if } e \text{ is PARTIALLY-SEEN}
\end{aligned}$$

where $rsize$ denotes the size of the relevant part of an element e, e' represents a descendent of e that is already retrieved before r in the ranked list, and α is a weighting factor lying between 0 and 1.

Precision and Recall is defined in terms of the above defined functions as :

$$Precision@r = \frac{\sum_{i=1}^{r} pre_i(e)}{r}$$
$$Recall@r = \frac{\sum_{i=1}^{r} rec_i(e)}{Trel}$$

where $Trel$ is the total amount of relevant information for an INEX topic.

To arrive at a single composite value they used the F-measure defined as

$$F@r = \frac{2}{\frac{1}{Precision@r} + \frac{1}{Recall@r}}$$

Pehcevski and Thom [97] also proposed four different ways of measuring overlap to circumvent the problem associated with set-based overlap which does not differentiate among different types of overlap:

a. Overall overlap: identical to the set-based overlap

b. Ascendant overlap: measures percentage of elements that contain at least one other element in the set

c. Descendants overlap: measures percentage of elements that are contained by at least one other element in the set

d. Probabilistic overlap: measures the probability that two randomly chosen elements from a set of retrieved elements overlap with each other.

2.6.5 INEX 2006

The INEX 2006 Ad Hoc track had four retrieval tasks, namely focused task, thorough task, relevant in context task, and best in context task. XCG measures introduced in INEX 2005 were used for focused and thorough tasks. Relevant in context was evaluated using HiXEval measures, while an adapted version of EPRUM was used for best in context task [66].

Kazai et al. [60] provided a well-studied comparison among some of the above metrics. Among the Cumulated Gain-based metrics, they opine that ep/gr graphs, $MAep$, and Q-measure are the most informative and discriminative measures. $MAnxCG$ and $nxCG$ at high cutoff values are the most robust measures, while $nxCG$ at lower cutoffs and $iMAep$ are the most sensitive. Q, R, $MAep$, and $nxCG$ at middle-range cutoffs provide stable but still discriminative measures. They also observed that Q and $MAep$ are highly correlated.

2.6.6 INEX 2007 Onwards

Since INEX 2007, effectiveness has been measured using metrics based on the notions of recall and precision, suitably adapted to fit the XML context:

$$\begin{aligned} \text{precision} &= \frac{\text{amount of relevant text retrieved}}{\text{total amount of } \textit{retrieved} \text{ text}} \\ &= \frac{\text{length of relevant text retrieved (in characters)}}{\text{total length of } \textit{retrieved} \text{ text (in characters)}} \\ \text{recall} &= \frac{\text{length of relevant text retrieved (in characters)}}{\text{total length of } \textit{relevant} \text{ text (in characters)}} \end{aligned}$$

Kamps et al. [54] provide more formal definitions as follows. Let p_r be the document part at rank r in the ranked list L_q returned by a retrieval system for a topic q. Let $size(p_r)$ be the total number of characters contained by p_r and $rsize(p_r)$ be the length (in characters) of relevant text contained in p_r (as highlighted by the assessor during the relevance judgment process). If there is no highlighted text, $rsize(p_r) = 0$. Further, let $Trel(q)$ be the total amount of relevant text for topic q (this is the sum of the lengths of relevant texts across all documents). Then,

$$\text{precision at rank } r,\ P[r] = \frac{\sum_{i=1}^{r} rsize(p_i)}{\sum_{i=1}^{r} size(p_i)} \tag{2.22}$$

and

$$\text{recall at rank } r,\ R[r] = \frac{\sum_{i=1}^{r} rsize(p_i)}{Trel(q)} \tag{2.23}$$

Since retrieval granularity can vary, a comparison of precision values at a given rank across systems may not be meaningful. Instead, precision at various recall levels may be used. Thus, interpolated precision at various recall levels is used for comparing systems, where interpolated precision at recall level x is defined as follows:

$$iP[x] = \begin{cases} \max\limits_{\substack{1 \leq r \leq |L_q| \\ R[r] \geq x}} (P[r]) & \text{if } x \leq R[|L_q|] \\ 0 & \text{if } x > R[|L_q|] \end{cases}$$

For the INEX adhoc tasks, $|L_q| \leq 1500$.

For example, $iP[0.00]$ gives the interpolated precision when the first relevant unit is retrieved, and $iP[0.01]$ is the interpolated precision at 1% recall level for a given topic.

Analogously, for a particular topic t, average interpolated precision AiP is defined as the average of interpolated precision values at 101 standard recall levels $[0.00, 0.01, \ldots, 1.00]$:

$$AiP(t) = \frac{1}{101} \sum_{x=\{0.00,0.01,\ldots,1.00\}} iP[x](t)$$

Overall performance measure: The overall performance of a system was measured by averaging its scores across all the topics in the set. If there are n topics, the performance of a system at recall level x is given by:

$$iP[x]_{overall} = \frac{1}{n} \sum_{t=1}^{n} iP[x](t)$$

Similarly, mean average interpolated precision ($MAiP$) over n topics is expressed as

$$MAiP = \frac{1}{n} \sum_{t=1}^{n} AiP(t).$$

Since INEX 2007, for the focused adhoc task, mean interpolated precision at four selected recall levels, $iP[x]$, $x \in \{0.00, 0.01, 0.05, 0.10\}$ and $MAiP$ were reported, and $iP[0.01]$ was selected as the “official” metric that was used to rank systems.

2.7 SMART system

The SMART system is a sophisticated text processing system based on the vector space model, originally developed at Cornell University during the late 60s and 70s [130]. The SMART system generates vectors for any given text automatically by indexing the text. Automatic indexing of a text usually involves the following steps:

- **Tokenization:** The text is first tokenized into individual words and other tokens.
- **Stop word removal:** Common function words (like *the, of, an, ...*), also called stop words, are removed from this list of tokens. The SMART system uses a predefined list of 571 stop words.
- **Stemming:** Various morhological variants of a word are normalized to the same stem. A variant of the Lovins stemmer is used in this process.
- **Phrase formulation:** Optionally, phrases in a text are recognized and are used in addition to the list of single words to index the text.
- **Weighting:** The term (words and phrases) vector, thus created for a text, is weighted using tf, idf, and length normalization considerations.

In the SMART system, term weighting schemes are denoted by triples of letters. The first letter in a triple is shorthand for the term frequency factor being used in the term weights, the second letter corresponds to the inverse document frequency function, and the third letter corresponds to the normalization factor applied to the term weights.

For all our retrieval related experiments we use the $Lnu.ltn$ term-weighting scheme [130] in which document and query-term weights are computed as follows:

$$\text{Lnu:} \quad \frac{(1+\log(tf))}{\textit{no. of unique terms}} \tag{2.24}$$

and

$$\text{ltn:} \quad (1+\log(tf)) \times \log(\frac{N}{df}) \tag{2.25}$$

2.8 Test Collections

For most of our experiments, we used the INEX adhoc collections. Different components of the collections are described below.

2.8.1 Corpus

The INEX 2006 adhoc collection consists of an XML-ified version of the English Wikipedia. This corpus was used for the INEX adhoc tasks during INEX 2006-08. Some statistics for these corpora [27] are given in Table 2.2 and Table 2.3.

Table 2.2: **INEX 2006 adhoc corpus statistics**

Size	4.6 GB
Number of documents	659, 388
Number of elements	52 million (apprx.)
Avg. #nodes/article	161.35
Avg. depth of an element	6.72
Vocabulary-size	2 million (apprx.)
Number of tags	1200 (apprx.)

Since INEX 2009, a larger Wikipedia collection is being used in the INEX adhoc tasks. It has been created from the October 8, 2008 dump of English Wikipedia articles and incorporates semantic annotations of YAGO[4] [37].

Table 2.3: **INEX 2009 adhoc corpus statistics**

Size	50.7 GB
Number of documents	2, 666, 190
No. of elements (length ≥ 50 chars)	101, 917, 424

[4] stands for *Yet Another Great Ontology*

2.8.2 Topics

INEX topics are developed by the participants. Each topic contains a set of terse keywords (*title* or T), a short *description* (D) and a detailed *narrative* (N) along with some other fields in XML format. An example from the INEX 2009 topic set specifying both structural and content-based requirements is given below:

```
<topic  id="2009111" ct_no="307">
   <title> europe solar power facility
   </title>
   <castitle> //article[about(., solar power)]//facility[about(., europe)]
   </castitle>
   <phrasetitle> "solar power"  "facility in  Europe"
   </phrasetitle>
   <description>I  am looking for solar  power facilities  in Europe
   </description>
   <narrative>I am writing an article  about solar power facilities in
              Europe and I need information  about  these, like  exact
              location,  history and  future plans.
    </narrative>
</topic>
```

Query-sets used in different INEX adhoc tracks are as follows (Table 2.4):

Table 2.4: **INEX adhoc topics**

Year	Number	Query-id
2006	125	289–413
2007	130	414–543
2008	135	543–678
2009	115	2009001–2009115
2010	107	2010001–2010107

2.8.3 Relevance Judgments

Evaluation of Information Retrieval systems at standard forums like TREC, CLEF, NTCIR, INEX or FIRE is based on the Cranfield paradigm where along with a corpus and topic set, one needs a set of relevance judgments for the topic set i.e. ground-truth about which entities (documents / elements / passages) are relevant and which are not for each topic. This labelled data is called *qrels* and the process of creating qrels is called relevance assessment or judgment.

To check the relevance of all the documents in a large corpus for each topic is prohibitively expensive. Therefore relevance assessment is done through *pooling* [134] – a technique where set-based union of top-k (k is suitably chosen depending on available resources, typically taken as 100) retrieved items from all the submitted runs are exhaustively judged for relevance for each topic. Information items outside the top-k pool are assumed non-relevant.

Since INEX deals with XML retrieval, the notion of relevance is defined at the sub-document level. For adhoc tasks, a contiguous text passage, either within an element or ranging across elements, is considered relevant for each document in the pool. An assessor marks a text as relevant using a highlighting tool [65]. At INEX, the participants are themselves the assessors, with a topic being typically judged by its creator.

Although relevance judgment is done at the sub-document level, the INEX pool is created at the document level. Even if a small passage is retrieved from an article, the complete article is included in the pool. This unique feature adds to the robustness and diversity of the INEX pool as the pool, and thus qrels, potentially contain some relevant passages/elements not retrieved by any of the contributing systems.

Some basic statistics about the INEX qrels used in our study are given in Table 2.5:

Table 2.5: **INEX adhoc qrels**

Year	#topics fully judged	#docs pooled
2006	114	58819
2007	107	65503
2008	70	42272
2009	68	50725
2010	52	39031

Our retrieval strategies are evaluated based on the above qrels provided by INEX. Our experiments related to evaluation are also primarily on these relevance data.

Apart from INEX, we also use TREC and NTCIR submission files and their pools for some of our experiments (which will be discussed later) where we generalize our observations on evaluation to the document retrieval setting.

Chapter 3

Baseline Retrieval

This chapter describes our retrieval experiments based on the Vector Space Model (VSM). A substantial portion of this work has been done as part of our participation in the INEX adhoc tasks since 2006. Specifically, effective indexing and term-weighting strategies for XML retrieval using VSM have been explored within the SMART retrieval system. Among the two issues related to XML retrieval that we have attempted to address, this chapter deals with the first one viz. the issue of *length normalization*, an important factor in VSM. Two parameters related to this factor, namely, *pivot* and *slope* have been combined into a single composite parameter. We experimentally find an 'optimum' value for the said parameter which yields good performance on different XML collections used in the INEX adhoc tasks across the years. The optimum value is shown to give good performance at both levels of retrieval granularity: document as well as element.

3.1 Introduction

Traditional Information Retrieval systems return whole documents in response to queries, but the challenge in XML retrieval is to return the most relevant parts of XML documents which meet the given information need. The INEX adhoc task has been classified into a number of sub-categories:

THOROUGH task. The aim of this taks is to study how systems estimate the relevance of retrievable information items. Systems are required to return XML elements ranked by their relevance to the topic of interest.

FOCUSED task. For this task, systems need to return a ranked list of *non-overlapping* XML components to the user.

RELEVANT in CONTEXT task. For this task, systems return XML elements grouped by article.

BEST in CONTEXT task. Systems return articles along with one best entry point to the user.

Upto 2006, only *elements* enclosed within a pair of related tags were considered as valid XML components or information-items. However, since 2007, a passage, either contained within an element or spanning across more than one element is also considered a valid retrievable unit.

Each of the above subtasks can be based on two different query variants: Content-Only (CO) and Content-And-Structure (CAS) queries. (for details see Section 2.3).

In 2006, our participation was in the THOROUGH task. In 2007 and 2008, the task was discontinued. From 2007 onwards, we participated in the FOCUSED task instead. Our retrieval efforts were primarily based on VSM as implemented in the SMART retrieval system. We used the CO variant of the INEX queries. The SMART system which is typically used for unstructured text retrieval or document retrieval was incrementally modified to enable it for sub-document level indexing and retrieval. For both document-level and element-level retrieval, we used pivoted length normalization scheme proposed by Singhal et al. [129]. The default values recommended by Singhal [130] for the parameters involved, namely *pivot* and *slope* were determined in a document retrieval setting through experiments with the TREC collections. The parameters therefore need to be tuned to suit XML retrieval. In this context, we investigate the following questions: will a single set of parameter values work for both document and element retrieval? can we tune the parameters in such a way that the same set of values are applicable across different XML corpora?

In the following section, we describe our retrieval approach which has gradually taken shape over the course of our participation in the INEX adhoc tasks. Next, results are reported and analyzed. Our conclusions are summarized in Section 3.6.

3.2 Approach

Since the INEX adhoc corpus of Wikipedia documents does not have a DTD, we did not have any prior knowledge about the text-containing nodes in the XML collection. Inspecting a few documents, we shortlisted about thirty tags that contain useful information: <p>, <ip1>, <it>, <st>, <fnm>, <snm>, <atl>, <ti>, <p1>, <h2a>, <h>, <wikipedialink>, <section>, <outsidelink>, <td>, <body>, etc.

We started by parsing documents using the LIBXML2[1] parser, and indexed only the textual portions included within the selected tags.

Later, we collected the statistics for the INEX 2006 adhoc corpus. Out of a total of 1259 tags, 74 tags that occur at least twice in the corpus, contain at least 10 characters of text, and are listed in the INEX 2007 qrels. Subsequently, only the content of these tags were indexed. Some examples of such tags are: <article>, <body>, <caption>, <center>, <collectionlink>, <definitionitem>, <defintionlist>, <div>, <em>, <figure>, <gallery>, <item>, <outsidelink>, <p>, <section>, <wikipedialink>, etc.

For the topics, we considered only the <title> and <description> fields for indexing, and discarded all other tags like <inex-topic>, <castitle> and <ontopic-keywords>. No structural information from either the queries or the documents was used.

The extracted portions of the documents and queries were indexed using Salton's blueprint for automatic indexing [115]. Stopwords were removed in two stages. First, we removed frequently occurring common words (like *know, find, information, want, articles, looking, searching, return, documents, relevant, section, retrieve, related, concerning,* etc.) from the INEX topic-sets. Next, words listed in the standard stopword list included within SMART were removed from both documents and queries. Words were stemmed using a variation of the Lovin's stemmer [72] implemented within SMART.

[1] www.xmlsoft.org

3.2.1 Phrases

Documents and queries were indexed using single terms and a controlled vocabulary (or predefined set) of statistical phrases. Initially, we used the default set of phrases used by SMART to index documents/queries. This phrase-list consists of all pairs of words that occur side-by-side 25 times or more in disk 1 of the TIPSTER collection. Later, we constructed a phrase-list from the Wikipedia corpus using N-gram Statistics Package (NSP)[2] and selecting the most frequent 100,000 word bi-grams.

Documents and queries were weighted using the *Lnu.ltn* [11] term-weighting formula (see Equation 2.24 and Equation 2.25).

Since SMART does not natively support the construction of inverted indices at the element level, we adopted a 2-pass strategy for XML retrieval. In the first pass, a set of candidate documents for each query was retrieved. Next, the most suitable elements were retrieved from these retrieved documents. Note that document retrieval can also be treated as element retrieval since <`article`> is a valid element.

3.2.2 Document Retrieval

At the document-level, the first task is a simple, inner-product similarity based retrieval (*VSM-doc-initial*). To boost the performance of document retrieval, we use automatic query expansion. Our automatic query expansion method is based on blind feedback and is given below [82]:

1. For each query, collect statistics about the co-occurrence of query terms within the set $\mathcal{S}$ of $\mathcal{N}$ top-ranked documents retrieved for the query by the initial run. We use a value of 1500 for $\mathcal{N}$. Let $df_{\mathcal{S}}(t)$ be the number of documents in $\mathcal{S}$ that contain term t.

2. Consider the 50 top-ranked documents retrieved by the baseline run. Break each document into overlapping 100-word windows as suggested in [82].

[2] http://www.d.umn.edu/ tpederse/nsp.html

3. Let $\{t_l, \ldots, t_m\}$ be the set of query terms (ordered by increasing $df_{\mathcal{S}}(t_i)$) present in a particular window. Calculate a similarity score *Sim* for the window using the following formula:

$$Sim = idf(t_{\mathit{1}}) + \sum_{i=\mathit{2}}^{m} idf(t_i) \times \min_{j=\mathit{1}}^{i-\mathit{1}}(\mathit{1} - P(t_i|t_j))$$

   where $P(t_i|t_j)$ is estimated based on the statistics collected in Step 1 and is given by

$$\frac{\textit{\# documents in } \mathcal{S} \textit{ containing words } t_i \textit{ and } t_j}{\textit{\# documents in } \mathcal{S} \textit{ containing word } t_j}$$

   This formula is intended to reward windows that contain multiple matching query words. Also, while the first or “most rare” matching term contributes its full idf (inverse document frequency) to *Sim*, the contribution of any subsequent match is deprecated depending on how strongly this match was predicted by a previous match — if a matching term is highly correlated to a previous match, then the contribution of the new match is correspondingly down-weighted.

4. Calculate the maximum *Sim* value over all windows generated from a document. Assign to the document a new similarity equal to this maximum.

5. Rerank the top 50 documents based on the new similarity values.

6. Assuming the new set of top 20 documents to be relevant and all other documents to be non-relevant, use Rocchio relevance feedback [106] to expand the query. The expansion parameters are given below:

$$\begin{aligned}
\text{number of words} &= 20 \\
\text{number of phrases} &= 5 \\
\text{Rocchio } \alpha &= 4 \\
\text{Rocchio } \beta &= 4 \\
\text{Rocchio } \gamma &= 2.
\end{aligned}$$

Finally, for each topic, 1500 documents were retrieved using the expanded query (*VSM-doc-feedback*).

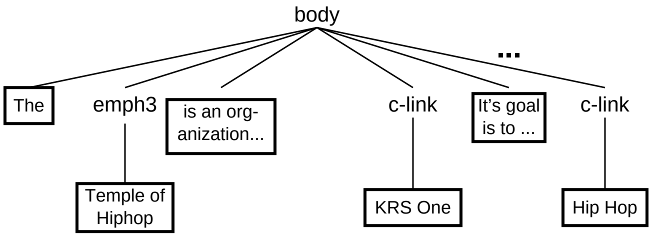


Figure 3.1: Parse tree for a fragment of a wikipedia document

3.2.3 Element Retrieval

Element retrieval is done on the results of our document retrieval step. The set of documents retrieved are parsed using the LIBXML2 parser, and initially only leaf nodes having textual content are identified. Figure 3.1 shows a fragment of a file from the wikipedia collection. The leaf nodes that have textual content are enclosed in rectangles in the figure. The total set of such leaf-level textual elements obtained from the 1500 top-ranked documents were then indexed and compared to the expanded query as before to obtain the final list of 1500 retrieved elements.

Since we considered only the leaf nodes as retrievable elements for this baseline element retrieval (*VSM-initial-Element*), the retrieved elements are automatically non-overlapping. However, as is clear from Figure 3.1, permitting only leaf-level textual elements to be retrieved has an obvious disadvantage: nodes such as <`p`> or <`body`> are typically not considered retrievable elements, because of the occurrence of nodes like <`emph3`> and <`collectionlink`> under the <`p`> or <`body`> node. Thus, very often, the retrieved leaf-nodes contain very small amounts of text that are non-informative in isolation; at the same time they preclude the retrieval of their ancestor nodes which are potentially more relevant. This strategy (*VSM-initial-Element*), therefore, performs poorly.

Next, we consider intermediate non-leaf nodes during retrieval. All intermediate nodes are regarded as retrievable if they have one or more leaf-level textual node(s) as their descendant(s). The content of an intermediate node spans the entire spectrum of text from its left-most descendant to the right-most descendant. The query is compared to all elements that contain text, instead of only the leaf-level textual nodes. In order to avoid any overlap in the final list of retrieved elements, the nodes for a document are sorted in decreasing order of similarity, and all nodes that have an overlap with a higher-ranked node are eliminated. Such retrieval runs are labelled as *VSM-fdbk-elt-slope0.x*.

3.3 Baseline Results

In 2006, our participation at INEX was as a new entrant. Our system was not ready for XML retrieval. We implemented a few minimalistic modifications to make it run on XML data and submitted the retrieval results in order to gain access to the test collection for understanding the issues in building a system for XML retrieval. We are not reporting the retrieval performance of our INEX 2006 participation.

Table 3.1 shows the performance of our XML document retrieval strategies over different INEX test collections. In INEX 2007 our document level performance (*VSM-doc-initial*) was satisfactory. The run ranked 4th among 79 runs in INEX 2007 according to $MAiP$. When feedback was used (*VSM-doc-feedback*), overall performance improved by at least 5%.

Table 3.1: **Document runs on INEX corpus**

Run Id	INEX data	Score				
		iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
VSM-doc-initial	2007	0.4680	0.4524	0.3963	0.3797	0.1991
VSM-doc-feedback	2007	0.4839	0.4682	0.4236	0.3957	0.2116
VSM-doc-initial	2008	0.6506	0.6339	0.5300	0.4791	0.2551
VSM-doc-feedback	2008	0.6415	0.6299	0.5526	0.5160	0.2755
VSM-doc-initial	2009	0.5140	0.4607	0.3838	0.3375	0.1701
VSM-doc-feedback	2009	0.4974	0.4531	0.3963	0.3433	0.1787

Similar observations were made with the INEX 2008 collection. The INEX 2009 corpus was more than 10 times larger than the INEX 2007 collection. The markup was also syntactically different, and contained some extra tags. Though performance dropped, feedback resulted in improvements in overall performance.

The top 1500 documents retrieved at the document retrieval are used as the basis for element retrieval. Table 3.2 provides details about the performance of our initial element retrieval strategy during INEX 2007-08.

Table 3.2: **Element-level runs on INEX corpus**

Run Id	INEX data	Score				
		iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
VSM-initial-Element	2007	0.2406	0.1820	0.0990	0.0548	0.0159
VSM-fdbk-elt-slope0.2	2007	0.4725	0.4172	0.3144	0.2497	0.0790
VSM-fdbk-elt-slope0.3	2007	0.4873	0.4318	0.3358	0.2620	0.0803
VSM-fdbk-elt-slope0.4	2007	0.5032	0.4558	0.3379	0.2374	0.0742
VSMfbElts0.4	2008	0.7152	0.6348	0.4805	0.4259	0.1538

In 2007, the first element retrieval run (*VSM-initial-Element*) used the retrieval results of *VSM-doc-initial* as a starting point. There are two reasons for its poor performance. One, this was the leaf-only run i.e. nodes at intermediate (non-leaf) levels were not considered. Leaf nodes are very often too small to contain any meaningful information; further, it is usually difficult to reliably rank such small pieces of text. Two, to have good element level retrieval, one needs to have good initial document level retrieval. *VSM-doc-initial* was not that good.

The first conjecture that retrieving non-leaf nodes improves performance was proved by the fact that each of the runs where intermediate nodes were considered for retrieval (*VSM-fdbk-elt-slope0.x*) performed substantially better than *VSM-initial-Element* (see Fig. 3.2 as well).

The second conjecture that good document retrieval is a pre-requisite for good element retrieval is substantiated when we compare query by query between the document run *VSM-doc-feedback* and the element run *VSMfbElts0.4* on INEX 2008 data (Table 3.3).

For a topic, if the document level run is good (score is above average), it is likely that the corresponding score for the element run will be good. This is true irrespective of the metric chosen

Table 3.3: **Per-query comparison between our INEX 2008 document and element runs**

Metrics	*VSM-doc-feedback* & *VSMfbElts0.4* (among 70 queries)				
	Both above avg	Both below avg	One over & other below avg	Pearson corr (ρ)	95% conf. int.[1]of ρ
iP[0.00]	31	21	18	0.5898	[0.412, 0.724]
iP[0.01]	29	22	19	0.5508	[0.363, 0.695]
AiP	25	29	16	0.6569	[0.499, 0.772]

[1] Based on Fischer's ρ-to-z conversion

($iP[0.00]$, $iP[0.01]$ or AiP). More quantitatively, the Pearson correlation coefficient (ρ) between document-level and element-level scores across queries is found to be significant at 5% (even at as small as 0.0001%) level of significance for all three metrics considered.

We initially considered the $slope = 0.2$ as recommended in [129] for document retrieval using $Lnu.ltn$ strategy. Although this parameter-setting has been seen to perform well in document retrieval across a number of document collections, how effective it is for the element retrieval is not tested. Later in INEX 2007, we, therefore, tried to tune the slope value during element retrieval. Keeping the same pivot value as was in document retrieval, slope value was changed to 0.2, 0.3 and 0.4. $slope = 0.4$ gave best results among the three which was seen to replicate with INEX 2008 data.

3.3.1 Analysis

On a closer look, our element-level runs were far from satisfactory, although our document-level runs were quite good (we achieved 4th rank in terms of $MAiP$ out of 79 runs in INEX 2007). Even the best among element-level runs with INEX 2007 or 2008 data achieves a $MAiP$ score of only 0.1538, which was found to be significantly lower than the scores of corresponding document-level runs on the basis of a t-test. Note that, here document run is actually considered as an element run since the whole XML article is also an element, the *root* element. Also, both kinds of runs are compared and evaluated with the same set of measures. More details about the relative performance of the two document-level runs and the element-level run are shown in Table 3.4.

The figures in the table show the number of queries for which a run performs better or worse than

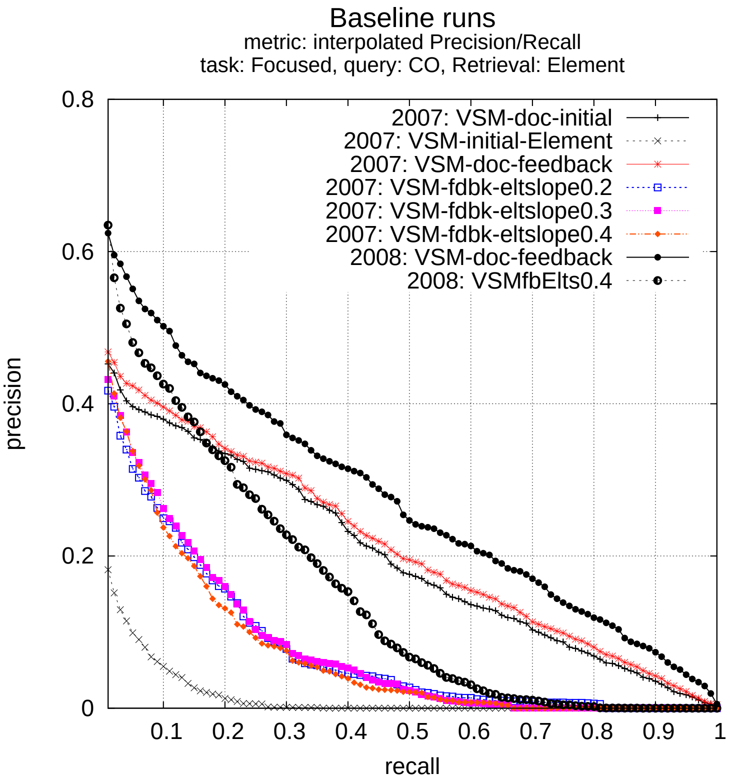


Figure 3.2: Interpolated P-R graph with INEX 2007-08 test collections

Table 3.4: **Comparison between document-level and element-level runs (AiP)**

Run Id	Better than VSMfbElts0.4	Worse than VSMfbElts0.4
VSM-doc-feedback	60(58)	10(10)

another run, as measured by AiP. The figures in parentheses correspond to the number of queries for which the relative performance difference is at least 5%.

Table 3.5: **Comparison between document-level and element-level runs (iP[0.00] and iP[0.01])**

Run Id	iP[0.00]		iP[0.01]	
	Better than VSMfbElts0.4	Worse than VSMfbElts0.4	Better than VSMfbElts0.4	Worse than VSMfbElts0.4
VSM-doc-feedback	28(20)	42(27)	36(29)	34(23)

A similar comparison on the basis of $iP[0.00]$ and $iP[0.01]$ is shown in Table 3.5. Since the elements retrieved by *VSMfbElts0.4* are typically considerably shorter than full documents, the first relevant element retrieved by this run is likely to be more focused than the first relevant element (a full document, actually) retrieved by the other runs. Also, the irrelevant elements preceding the first relevant element are likely to be shorter. Thus, the element level run performs better in terms of $iP[0.00]$. At subsequent recall points, however, the performance of the element-level run drops.

Looking from another aspect, our element runs in general performed poorly, possibly because the system is retrieving larger, less focused pieces of text than it should. Increasing the degree of document length normalization could be one way to address this problem. The term-weighting scheme that we use – pivoted document length normalization [129] – gives us an easy way to test this hypothesis. Under this term-weighting scheme, the document length normalization factor is given by

$$normalization = (1 - slope) \times pivot + slope \times length \tag{3.1}$$

Following Singhal's recommendation [130], we had initially set the slope and pivot parameters to 0.20 and 110 respectively for document runs and 0.2 and 80 for element runs. Increasing the slope value is expected to promote elements that are shorter than the pivot length, while pushing longer elements to lower ranks. Accordingly, we experimented with two more *slope* values, viz. 0.3 and 0.4. Table 3.2 suggests that our intuition is correct: increasing the degree of normalization for long documents seems to improve early precision. However, the $iP[0.05]$ and $iP[0.10]$ figures reach a point of diminishing returns as the slope is increased. Also, the $MAiP$ figures for these runs are rather dismal. These facts set the premise for further exploration of the issue of length normalization in order to understand the effect of normalization on precision at various recall points.

3.4 Length Normalization

We revisited the issue of pivoted length normalization in our term-weighting scheme in INEX 2009. Earlier we had blindly set the slope and pivot parameters to 0.20 and 80 respectively for our runs following [130]. On an adhoc basis we also used two more slope values, viz. 0.3 and 0.4, for the same pivot value. But these parameters were actually tuned for the TREC adhoc document collection. The INEX Wikipedia XML collection is both syntactically and semantically different (not from the news genre). More importantly, the average length of a Wikipedia-page is much smaller than a general English text document from the TREC collection. As Singhal [130] suggests, the average Wikipedia document length may be used as the approximate pivot value. Although this approach may work well for document retrieval, it will not solve the problem of element retrieval. The problem is further compounded by the fact that there is no single granularity of the elements. Hence, using a single average element length across elements of varying granularities does not seem convincing. On the other hand, using different average lengths for different granularities will make the process too complicated and difficult to handle. It would be desirable to set the parameter to a value that is independent of element granularity.

3.4.1 Implementation

In the *Lnu.ltn* scheme, the pivoted document length normalization factor [129] can be rewritten as

$$normalization = 1 + \frac{slope}{(1 - slope) \times pivot} \times (\textit{\# unique terms}) \tag{3.2}$$

where document-length is given by *# unique terms* in the document. We also observe that we can consider the factor $\frac{slope}{(1-slope)*pivot} = c$, a constant which reduces Eqn.(3.2) to

$$normalization = \mathit{1} + c \times (\#\ \textit{unique terms})$$

If the pivot value in an optimal constant c is changed from $pivot$ to $pivot'$, the slope value can be suitably modified from $slope$ to $slope'$ to achieve the same c. In other words, instead of tuning the pivot and slope separately for each collection, we can tune a single parameter, i.e. the constant c.

With $0 \leq slope < \infty$ and $pivot \geq 1$, the factor $c \in [0, \infty)$. Although $pivot$ is supposed to be taken as the average length of an element in terms of number of words and $pivot \geq 1$, we can take any positive value for pivot and slope will be suitably adjusted. For the sake of simplicity, we chose $pivot = 1$ so that the factor c (we call it *pivot-slope* factor) reduces to $\frac{slope}{(1-slope)}$ and varied the factor starting from 0 towards higher values.

Within SMART system, we made the necessary modifications and at first element retrieval was studied. On the returned list of document retrieval with blind feedback, *pivot-slope* factor is gradually increased from 0 to 1 in steps of 0.1 (slope is actually adjusted to effect the variation). Based on the observed change in overall retrieval scores (in terms of *MAiP*) we narrowed down the range of interest. We found that *pivot-slope* $= 0.1$ also seemed to cause over-normalization. Recursively searching in the $[0.0,\ 0.1]$ range, we got a magic-value 0.00073 for the factor (the same value for $slope$ as well) that gave best *MAiP* score.

Later, the experiments were repeated in document retrieval setting also. Surprisingly, the same value of the *pivot-slope* factor worked well for document retrieval as well.

We carried out the experiments with the INEX 2007, 2008 and 2009 collections. The observations are detailed in the following section.

3.4.2 Results

Element Retrieval

Table 3.6 gives a brief summary of the variation of scores for different metrics in element retrieval using INEX 2009 data. Though we planned to vary slope from 0 to higher values, we found that the optimum slope should be very close to 0. Actual range is recursively narrowed down by successively considering $[0,\ 0.1]$, followed by $[0,\ 0.05]$, followed by $[0, 0.005]$ and so on.

Table 3.6: **Element-retrieval over result of *2009:VSM-doc-feedback*, pivot = 1**

slope	0	0.01	0.05	0.10	0.20	0.30	0.40	0.50	1.0	0.00073	0.005
iP[0.00]	0.4146	**0.4908**	0.3221	0.1934	0.0698	0.0460	0.0265	0.0094	0.0044	0.4432	0.4881
iP[0.01]	0.4139	0.4495	0.2149	0.1049	0.0244	0.0170	0.0036	0.0029	0.0000	0.4425	**0.4703**
iP[0.05]	0.3815	0.3519	0.0873	0.0222	0.0122	0.0124	0.0000	0.0000	0.0000	**0.4030**	0.3836
iP[0.10]	0.3294	0.2634	0.0604	0.0115	0.0000	0.0000	0.0000	0.0000	0.0000	**0.3468**	0.3061
MAiP	0.1542	0.0992	0.0168	0.0054	0.0015	0.0011	0.0003	0.0001	0.0000	**0.1596**	0.1231

Though there are few slope values (0.01 or 0.005) where early precision values gives best results but the value 0.00073 shows consistent good results for a wide range of recall points across the entire spectrum, so far element retrieval performances are concerned (Fig. 3.3). Also, note that $MAiP$ is more stable and robust as a metric compared to early precision metrics like $iP[0.00]$ or $iP[0.01]$. (see Chapters 5, 6 and 7 for more details.)

The experiment was repeated for INEX 2007 and 2008 data, where we observed a similar pattern (See Table 3.7 and Table 3.8). With INEX 2008 data, we actually varied pivot-slope factor from 0 to 0.5 in steps of 0.05; therefore *slope* was adjusted to the values as shown in Table 3.8. Once again proper normalization yields marked improvements in the retrieval score (more than 180% for INEX 2007 and more than 89% for INEX 2008) compared to our baseline results (cf. table 3.2).

Table 3.7: **Element-retrieval over result of *2007:VSM-doc-feedback*, pivot = 1**

slope	0	0.00073	0.01	0.1	0.2	0.3	0.4	0.5	1
iP[0.00]	0.3766	0.4750	**0.5351**	0.4559	0.4192	0.3980	0.3831	0.3578	0.2915
iP[0.01]	0.3758	0.4690	**0.4852**	0.2834	0.2111	0.1884	0.1646	0.1454	0.1052
iP[0.05]	0.3703	**0.4607**	0.3785	0.1392	0.0645	0.0479	0.0347	0.0239	0.0104
iP[0.10]	0.3544	**0.4214**	0.3132	0.0641	0.0307	0.0241	0.0189	0.0146	0.0042
MAiP	0.1735	**0.2126**	0.1163	0.0227	0.0144	0.0114	0.0096	0.0082	0.0054

Figure 3.4 shows the improvement graphically compared to our earlier baseline results at almost all recall points.

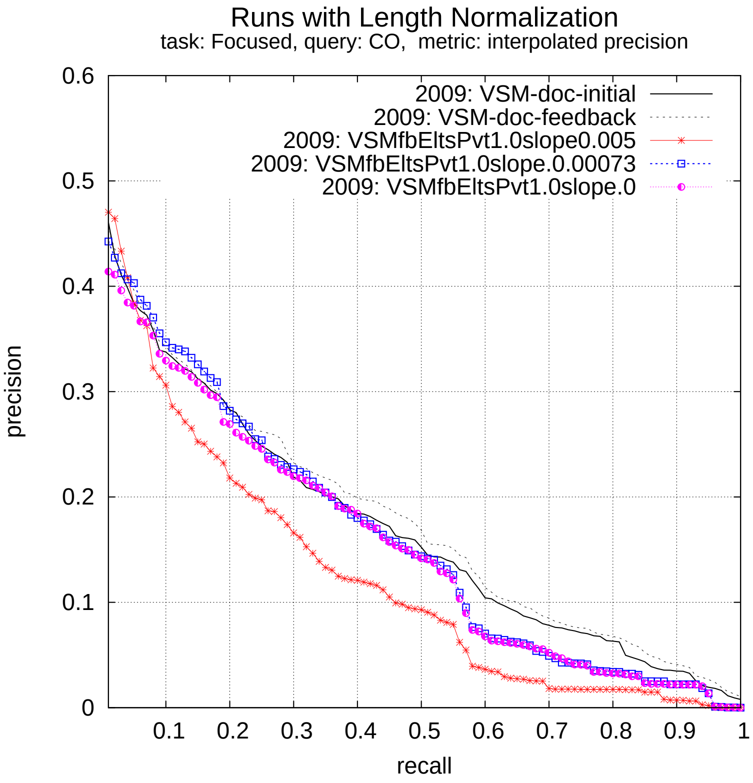


Figure 3.3: Interpolated P-R graph with INEX 2009 test collections

Document Retrieval

Does there exist an optimum value of the pivot-slope factor for XML document retrieval as well? If yes, what can be the magic-value here? We explore these two questions in a document retrieval

Table 3.8: **element-retrieval over result of *2008:VSM-doc-feedback*, pivot = 1**

slope	0	0.00073	0.05	0.13	0.17	0.20	0.23	0.26	0.31	1.00
iP[0.00]	0.5719	**0.6630**	0.6542	0.5894	0.5585	0.5438	0.5078	0.4865	0.4495	0.0668
iP[0.01]	0.5719	**0.6630**	0.4488	0.3290	0.2802	0.2690	0.2544	0.2313	0.1842	0.0067
iP[0.05]	0.5572	**0.6140**	0.2695	0.1283	0.1014	0.0811	0.0679	0.0605	0.0523	0.0007
iP[0.10]	0.5341	**0.5570**	0.1600	0.0399	0.0300	0.0279	0.0232	0.0224	0.0128	0.0000
MAiP	0.2520	**0.2921**	0.0468	0.0220	0.0176	0.0161	0.0145	0.0133	0.0111	0.0008

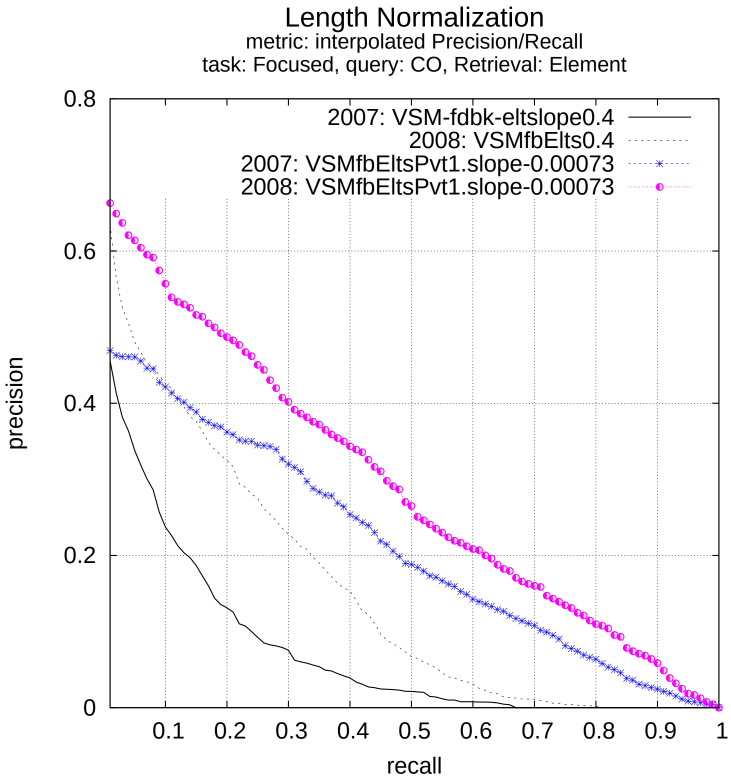


Figure 3.4: Interpolated P-R graph with INEX 2007 - 08 test collections

setting with the same INEX data. The treatment was exactly the same: we set $pivot = 1$ and varied the $slope$ value in the range 0 to 1.

The observation for baseline document retrieval (without applying feedback) with INEX 2009 data is summarized in Table 3.9. The same optimum value surprisingly worked well for document retrieval.

Table 3.9: **2009 doc-retrieval: variation of pivot-slope, pivot = 1**

slope	0	0.00073	0.005	0.10	0.20	0.30	0.40	0.50
MAP	0.2572	**0.2612**	0.1463	0.0250	0.0177	0.0149	0.0130	0.0121

Unlike earlier document retrieval scores, where we measured effectiveness in the element retrieval setting (in terms of $MAiP$), here we measure the score at the document level with Mean Average Precision (MAP). With feedback as described above, the best score *VSMfb-pvt1.0slope0.00073* improved by about 5% in MAP ($= 0.2743$) from its without feedback counterpart *VSMbase-pvt1.0slope0.00073* (Table 3.10).

Table 3.10: **2009: Document-level evaluation for the FOCUSED, CO task**

Run Id	MAP	% change
VSM-doc-feedback	0.2149	-
VSMbase-pvt1.0slope0.00073	0.2612	21.54
VSMfb-pvt1.0slope0.00073	0.2743	27.92

Both the runs *VSMbase-pvt1.0slope0.00073* and *VSMfb-pvt1.0slope0.00073* are significantly better than our earlier feedback run *VSM-doc-feedback* where $pivot = 80$ and $slope = 0.2$ were used.

Figure 3.5 shows the improvement in document retrieval pictorially for the entire range of recall points.

3.5 Discussion

Once we empirically identified the optimum value for the *pivot-slope* factor, we attempt to find through a study of length distribution why such a value works well. We sort the documents in the INEX 2009 corpus by length (measured as the number of distinct terms contained in a document) and divide the entire range of document lengths into a number of bins containing *nearly equal* number of documents. For each bin, we find the probability of relevance (= #relevant docs. in bin / total #rel. docs.) and probability of retrieval (= #retrieved docs. in bin / total #ret. docs.) and plot them against document length. Each bin is represented on the X-axis by the average document length for that bin. As postulated by Singhal [130], length normalization should be such that the probability of retrieval roughly follows the pattern of the probability of relevance when we plot

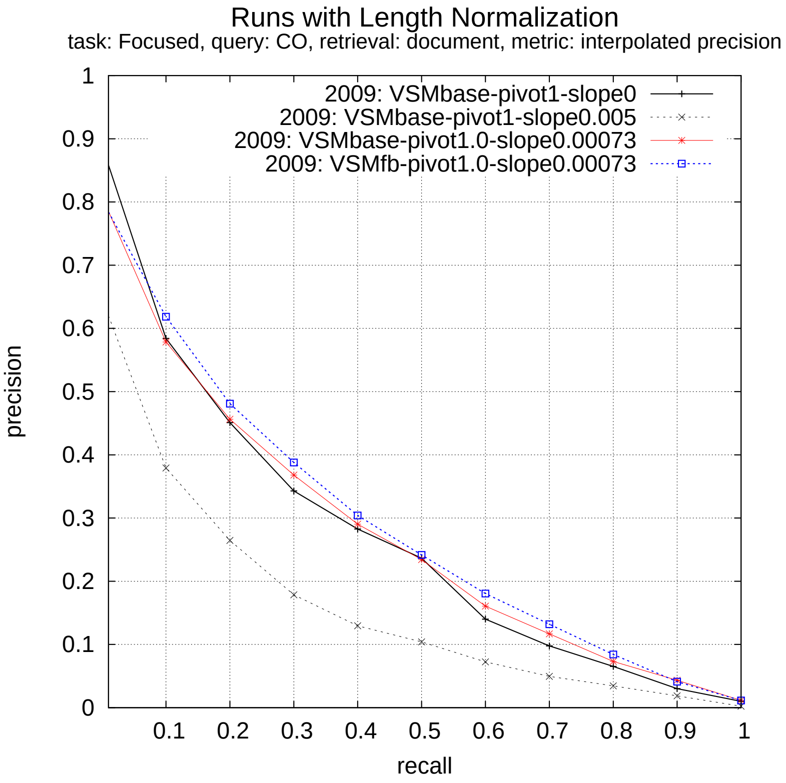


Figure 3.5: Interpolated P - R graph for document runs with INEX 2009 data

these values against document length.

For the document feedback run using our empirically found optimum pivot-slope factor (*VSMfb-pivot1-slope0.00073*), the retrieval and relevance curves are actually close together for a wide range of document lengths (Figure 3.6).

In contrast, our initial document feedback run using pivot = 110 and slope = 0.2 (*VSMfb-pivot110-slope0.2*) did not produce good results. It retrieved a high proportion of short documents, but did not retrieve an adequate number of long documents. Also, the two curves cross at about document length = 300, which should be used as the pivot (instead of the value 110 that we took initially).

Looking at it from another angle, if we used pivot = 300, and slope = 0.18, we would get the same pivot-slope factor = 0.00073 (See Eqn. 3.2).

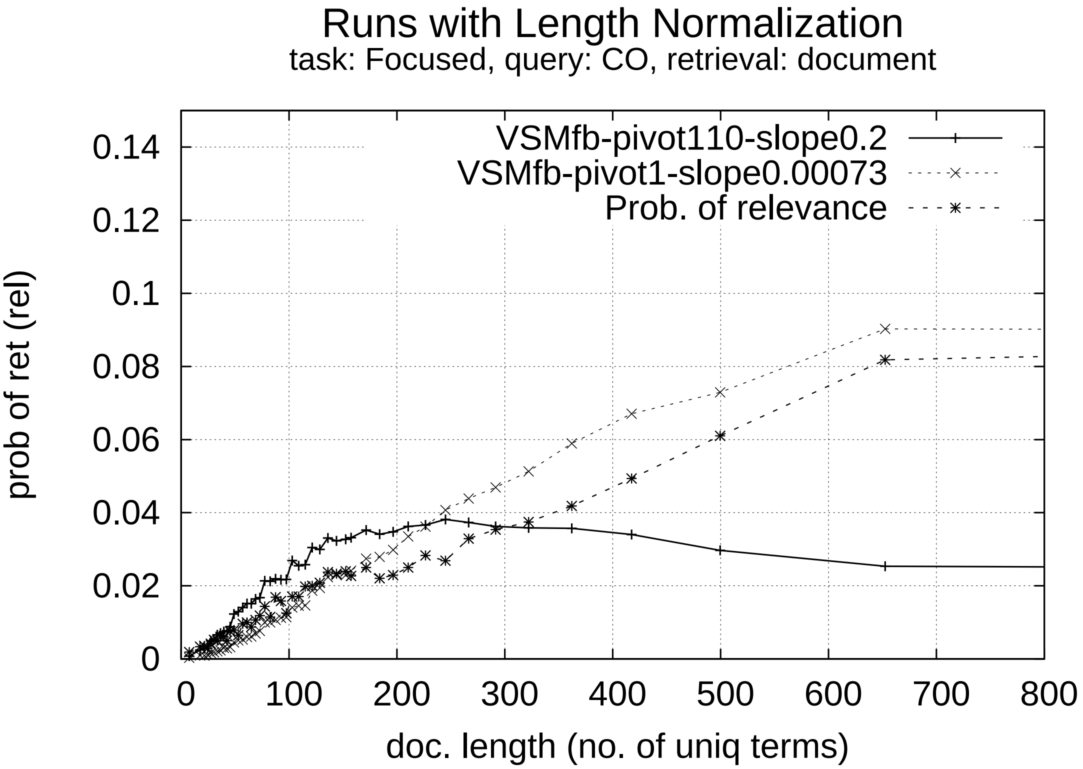


Figure 3.6: INEX 2009: Probability of retrieval/relevance vs. document length

Unfortunately, this approach to finding an appropriate pivot is not easily applicable to element retrieval, where retrieval granularity is not fixed. We, therefore, transformed pivot and slope to a composite parameter in order to achieve ease of tuning. Since the “optimum” value works well for both element and document retrieval across different years, we expect that this value may be used for any kind of XML retrieval using the Wikipedia collection.

3.6 Conclusion

This chapter described our retrieval experiments on XML document collections based on the Vector Space Model. The work centered around our participation in the INEX adhoc tasks, specifically the

THOROUGH and FOCUSED subtasks using CO queries. We used the INEX 2006 - 2009 adhoc test collections. Since SMART as a text retrieval system does not natively support sub-document level retrieval, we customized the system to enable it for XML element retrieval. We also explored effective indexing and term-weighting strategies for XML retrieval using VSM. We started with retrieval of only leaf-level nodes in the XML tree. Next, we incorporated the modifications necessary for retrieval of intermediate nodes. The results obtained for element retrieval were not satisfactory since we had not tuned the parameters used in pivoted length normalization. We therefore studied the issue of *length normalization*, in the light of XML retrieval. Wikipedia XML corpus used in INEX is syntactically and semantically different from news genre corpus used in TREC based on which parameters of pivoted normalization, namely *pivot* and *slope* were set earlier. Since length of an element is widely varying in XML documents depending on its level in the XML document structure, using a fixed average length as *pivot* for all elements is not a good idea. On the other hand, using variable pivot for different elements is too complicated to handle. We therefore, tried to combine *pivot* and *slope* to a single composite parameter. We experimentally found an ‘optimum’ value for the said parameter which yielded substantial improvements on different XML collections used at INEX during 2006 - 2009. The optimum value is shown to work well for both document- and element-level retrieval.

Chapter 4

Query Refinement

As mentioned in the Introduction, we investigate two ways to improve retrieval effectiveness for XML data - one, by using length normalization technique within the VSM; and two, through effective query processing. In Chapter 3 we discussed the first technique where we found for adhoc XML retrieval an ‘optimum’ pivot-slope factor which works quite well at both document- and element-level. In this chapter we look into the second approach of improving retrieval performance. Indeed, it is generally observed that a user finds it easier to formulate a complex information need in natural language rather than using terse keyword queries or using a formal query language. In XML, search queries can be very precisely formulated to retrieve precise document components, but the user often makes mistakes in framing the query due to the complicated syntax of query languages. Thus, for example, in INEX topics, we observe that a user often vividly states in the narrative (N) section, not only her information need, but also what she *does not* want. Most text retrieval systems do not handle this *negative information* properly. In this chapter we investigate whether this information can help improve retrieval performance. Initially, we adopt a naive approach of manually labelling *negative information* in INEX topics and then removing them from the N section. Such modified queries are used for retrieval from the INEX adhoc corpus. We observe that retrieval performance improves by 4-6% across different INEX corpora compared to a baseline in which the complete original queries are used. This observation sets the ground to explore whether the process of negation detection and removal can be automated. We use a maximum entropy classifer to segregate positive and negative sentences in the narrative sections of INEX topics. Retrieval with queries after automatic removal of negative sentences yields *equivalent* performance compared to the manual method (difference not statistically significant).

4.1 Introduction

For good retrieval, we need to start with a good query. As Lewis and Jones observed, however, *“many end users have little skill or limited experience in formulating initial search requests and modifying their requests after observing failure. Even while relevance feedback is available, it needs to be leveraged from a sensible starting point”* [70]. The problem is much worse in the field of semi-structured retrieval. For example, while formulating INEX 2003 structure-conscious (CAS) queries, as many as 63% of the queries were detected to contain syntactic errors [127]. A notable point is that these queries were mostly prepared by experienced IR researchers. One can only wonder how difficult the job is for casual users. In fact, an end-user always thinks in natural language but she is expected to formulate queries either with some keywords (for document retrieval) and/or in some specified format (structured or semistructured format). What the user has in mind is therefore not always properly reflected in the query. Had she been allowed to pose her query in natural language, she can verbosely express the information need: what she actually wants and what she does not.

Since XML allows retrieval at lower granularities, users expect a retrieved unit to be specific to the information need with as little irrelevant content as possible from a document. Hence, most of the INEX topics are very detailed and verbose in the narrative section. Along with their information needs, users also explicitly express there what they do not want or what documents or texts are not relevant to them. The narrative section (N), where this *negative* information mainly occurs is, however, either not used during retrieval or is not handled properly by present-day IR systems. For example, consider INEX query 439 (*interactions and relations of Japanese and Jews during the Second World War*) which specifies in the narrative: *An element that describes both Jews and Japanese separately without making any connections is irrelevant. An element is irrelevant if its topic is insignificant. For example, if there is a minor pop music album whose lyrics include*

Japanese and Jews during the Second World War, the element is irrelevant. An element is also irrelevant, if from the element alone it's unclear when the interactions happened.

We observe that there are three types of terms in the narrative section. Terms like ‘Jews’, ‘Japanese’ ‘interaction’, “second world war” are important terms that need to co-occur in a document to be relevant. These can be considered as ‘positive’ terms which the topic creator stresses, and whose presence in a relevant document is highly likely. Terms like ‘music’, ‘album’, ‘lyrics’ are unrelated or antithetical to the focus of the topic. We call these terms ‘negative’. Also there are terms like ‘insignificant’, ‘minor’, ‘unclear’ which are ‘weak’ terms, and we can ignore them during retrieval.

In general, longer, verbose queries are found to improve retrieval results [7]. However, if the increase in length of a query is due to the addition of negative terms, we expect it to yield poor results. Thus, if we completely ignore the narrative section, we miss some information as to what makes a unit relevant. On the other hand, if we treat all these terms in the narrative in the same fashion, there is a risk of diluting the focus of the query. Again, consider the following complete query:

```
<topic id="2009034" ct_no="219">
  <title>the evolution of  the  moon</title>
  <castitle>//*[about(.,  the  evolution  of  the moon)]</castitle>
  <phrasetitle/>
  <description>Find  information about the origin and evolution of the
               moon.
  </description>
  <narrative> Human exploration of the  moon has found that there are
              a  lot  of  mysterious  phenomena,  and  now scientists
              haven’t yet found a reasonable explanation. If  we want
              to  understand  those  mysterious phenomena, we need to
              first make clear the origin and evolution of the  moon.
              Articles talking about the origin and evolution of  the
              moon  are  relevant.  Articles  introducing  mysterious
              phenomena  happened  on  the  moon  are  relevant, too.
              Articles  introducing human exploration experiences  of
              the moon or relevant organizations, instruments, events
              are irrelevant.
  </narrative>
</topic>
```

Notice that the title (T) and the description (D) do not contain terms related to ‘mysterious phenomena’, ‘scientists’ or ‘explanation’ which are important positive terms and only occur in the narrative (N). On the other hand, ‘human exploration’, ‘organizations’ and ‘instruments’ are negative terms. Intuitively we can say that if we include these positive terms but exclude the negative terms from the narrative during retrieval, performance can potentially be improved.

We have seen that a TDN query (i.e. a search query formed using all three fields - T, D, N - of a topic) usually results in better retrieval effectiveness compared to a TD query (one that uses terms from only the T and D fields). This typically happens because the narrative contains new positive terms or reinforces the importance of positive terms occurring in the T or D fields. The above example shows, however, that the N field for several queries contain negative terms. Most IR systems do not identify or distinguish positive terms from negative ones, and treat them equally during query processing. When negative terms receive the same weightage as positive a completely irrelevant document containing only the negative terms may be retrieved at high ranks. Both precision and recall suffer when non-relevant documents achieve high rank and relevant documents are not retrieved or get lower ranks because of topic dilution or topic drift caused by the presence of negative and weak terms. In XML retrieval, where the narrative for a substantial number of topics contains negative information, the issue is not negligible and therefore worth-investigating.

Our objective is to take into account the detailed specification provided in a verbose query and to explore ways to effectively and automatically make use of such detailed specifications to improve retrieval performance. Though the hypothesis of performance augmentation through the removal of negative terms seems intuitive, we need to experimentally verify it. Secondly, if there indeed is any improvement, the challenge is how to automatically detect positive and negative terms from the queries and to design an effective weighting scheme for these positive and negative terms so that overall retrieval performance improves.

We attempt to address these issues in this chapter. The following sections discuss related work (4.2), the approach taken (4.3) and then results obtained (4.4). We discuss the limitations and scope of future work in Section 4.5 and finally conclude in Section 4.6.

4.2 Related Work

The history of handling Natural Language Queries (NLQs) and tackling related problems is quite old. One of the earliest studies is by Warren and Perreira [146]. The paper described their prototype natural language question answering system. With the aid of a logic-based grammar formalism, English questions were translated into Prolog logic-subset, then transformed into efficient Prolog queries which are executed to yield the answer. The work was limited to a small set of English queries.

Another attempt was reported in the early nineties by Nakkouzi et al. [86] which affect both IR systems and their users. They rightly recognized the problem of NLQs with negation. The paper presented an algorithm to transform Boolean queries with negated terms into equivalent queries without them based on a hierarchical thesaurus. However the work, by their own admission, considered a thesaurus with a specific structure and was not supported by adequate experimental evaluation.

The problem with negation in IR was seen as early as the initial TREC adhoc tasks. Dumais [28, 29] observed that many of the early TREC queries (TREC-1, TREC-2) contained a lot of “NOTS”. At TREC-1, Bellcore’s performance was brought down because their Latent Semantic Indexing (LSI)-based technique could not handle negated queries [28]. When they manually removed negated phrases from TREC 2 adhoc queries, performance improved marginally (about 2%) [29].

McQuire et al. [80] developed a prototype system for handling negation in NLQs. The algorithm used was based on the result of a survey where for each query containing negation, several possible choices were given to remove negation. However, the approach was not implemented in any IR system.

Techniques for detecting negation in text have long been applied in the medical domain. Negation identification helps determine the absence of a medical condition in a particular patient report. Lior et al. [107] identified the negative concepts by training a classifier using regular expression patterns. Other methods have also been successfully used in negation detection, including regular expression matching [21], lexical analysis and parsing [85], and machine learning algorithms like

Naive Bayes and Support Vector Machines (SVM). Goryachev et al. provide a survey of such efforts [40]. In most cases, the medical reports were marked up using UMLS (a medical domain mark-up language) and these mark-up tags act as cues to the algorithms to detect the negation of a medical condition. However, the nature of the underlying data is specific to a particular domain and therefore different from the XML queries in free text provided in INEX tasks.

In the semi-structured domain, some work was done as a part of the Natural Language Processing (NLP) track of INEX. The task was to explore the potential of extracting NEXI (Narrowed Extended XPath I) specifications from NLQs. Tannier's approach [136] was grammar-rule based. A part-of-speech (POS) tagged query was first analyzed with the help of a set of predefined reduction rules to identify specific structural and content part. The parts are then synthesized to eventually form a structured query. The NLPX system (Woodley and Geva [153], [152], [154]) uses POS-tagging and chunking, and matches tagged NLQs to previously constructed query templates. Matching templates are then merged to form NEXI queries. From the query text, the authors identified some term classes in NLQs like STRENGTHENERS (e.g. 'major' or 'particularly'), REVERSE BOUNDARIES ('talked about' or 'described'), NEGATORS (like 'except' or 'user does not want'). Both Tannier as well as Woodley and Geva applied standard NLP techniques to analyze NLQs and can be considered to be related to the work described in this chapter. However, these efforts were confined to structured query formulation, and thus have a different focus from our work.

Some studies on long queries [5, 8] have reported on the use of query re-weighting, query reduction, and the identification of key concepts in a query. Our approach also uses a simple query reduction process. The difference is that we try to automatically identify the negative constraints in the narrative section of the query using an off-the-shelf Maximum-Entropy classifier. We discard negative information with the hope that this would remove negative terms from the query, thereby improving retrieval performance.

4.3 Approach

Our approach can be broadly divided into two steps: (i) identifying the constructs (sentences / clauses / phrases) in the detailed statement of information need that specify positive and negative constraints; and then (ii) effectively utilising positive and negative constraints separately in the retrieval process. We start by identifying negative constraints in a verbose query (specifically the *narrative* section of the INEX topics), and show how retrieval results improve when these negative portions are simply removed from the user's query.

4.3.1 Positive and Negative Constraints

Sentences specifying what the user wants are termed as *positive*, while those which specify what the user is not looking for are termed *negative*. Thus, the sentence *I need X but I don't want Y*, when spliced, gives a positive part *I need X* and a negative part *don't want Y*. A sentence of the form *Z is not irrelevant* is labeled as positive because of the presence of double negation. A query broken into its positive and negative parts is shown below in Table 4.1:

Table 4.1: **Positive and negative parts of query 500.**

I have asthma and want to know if it is pollen triggered. I am looking for allergen plants and the symptoms of these allergies. Interesting results can treat allergies caused by plants or allergen plants. Food allergy is considered as irrelevant. I am not interested in drugs or treatments for allergies.

Positive part

I have asthma and want to know if it is pollen triggered. I am looking for allergen plants and the symptoms of these allergies. Interesting results can treat allergies caused by plants or allergen plants.

Negative part

Food allergy is considered as irrelevant. I am not interested in drugs or treatments for allergies.

The entire exercise is done in two stages. First, we take a manual approach to substantiate our claim that negation removal indeed *significantly* improves retrieval performance. Next, the process of negation identification is automated with the help of supervised learning. We show that the second approach also yields a performance boost *equivalent* to that provided by the manual approach.

4.3.2 Manual Separation

We use the adhoc queries from 4 years' (2007 – 2010) INEX test collections. The narrative of each query is broken into sentences as shown above. These sentences are then manually labelled as positive and negative. Corresponding to a given query q, we construct a positive query q_p by combining the title (T), the description (D), and all positive sentences from the narrative (N) of q after discarding the negative sentences. A set of new queries is thus formed from each query collection, we call this the positive topic set (P_{MANUAL}). The original verbose query set with complete T, D, N is called Q. Retrieval is done using the SMART system with $Lnu.ltn$ term-weighting strategy with the original verbose query set (Q) as well as positive query set (P_{MANUAL}). Two separate sets of retrieval runs are evaluated: a baseline run and a run using automatic query expansion based on Rocchio blind feedback. The feedback is done with 20 most frequent terms and 5 most frequent phrases occurring in the top 20 pseudo-relevant XML documents with settings $(\alpha, \beta, \gamma) = (4, 4, 2)$ as explained in the previous chapter (Chapter 3).

4.3.3 Automatic Separation

Since the manual removal of negative constraints from the verbose query results in significant improvements in retrieval performance (cf. Section 4.4.1), we next explore ways to automate the process of labelling sentences as positive or negative.

We use the Stanford classifier [101], a Java implementation of Maximum Entropy classifier as the classifier is seen to perform best with our data. A set of manually labelled sentences from the narrative of INEX topics is used to train the classifier, which is then used to classify a test-set of sentences from the narratives of a different INEX topic-set into one of the two classes. The negative sentences thus identified are purged from the respective narrative sections of INEX topics. The set of topics containing original title (T), description (D), and only the machine-identified positive sentences from the narrative thus created is denoted as P_{MAXENT}. The experiment described in 4.3.2 that is done with P_{MANUAL} or Q is repeated with P_{MAXENT} as well.

4.4 Results

4.4.1 Manual Separation

We verify our hypothesis on the INEX 2008 - 2010 datasets as they have the same evaluation setup with similar test collections. Only those queries with qrels available are included in our study as no quantitative comment can be made on the queries without qrels. However not all the queries with qrels have clear negative constraints. As a part of the manual separation process, we identified queries for which clear demarcation between positive and negative constraints is possible. Statistics for the 2008, 2009, and 2010 query sets are shown in Table 4.2.

Table 4.2: **INEX adhoc qrels**

Year	**# queries**		
	Total	**With qrels**	**Manually separated**
2008	135	70	44
2009	115	68	36
2010	107	52	26

We observe that among the queries with available qrels, 50% or more queries have clear negative constraints in the narrative section.

When we manually remove the negative constraints from the queries that contain negation, keeping the other queries untouched, we notice an improvement in the overall performance compared to the baseline TDN runs (Table 4.3). There is a performance gain either in MAP or in the number of relevant documents retrieved or both.

When we focus only on the subset of queries containing negation, the improvement is naturally

Table 4.3: **Performance of the original query set (Q) and the manually separated set ($P_{\textbf{MANUAL}}$) over complete qrels**

Year (#reldocs)	MAP (#rel-ret) with Q	MAP (#rel-ret) with $P_{\textbf{MANUAL}}$	Change in MAP
2008 (4887)	0.3032 (3988)	0.3103 (4010)	+ 2.3%
2009 (4858)	0.2496 (3613)	0.2568 (3609)	+ 2.9%
2010 (5471)	0.2657 (3802)	0.2746 (3853)	+ 3.4%

somewhat more substantial (Table 4.4). Note that the figures in this table are based on the scores of only those queries that have negative constraints (Table 4.2, last column). For both the runs (Q and P_{MANUAL}), pseudo-relevance feedback is used for automatic query expansion.

Table 4.4: **MAP of the original query set (Q) and the manually separated set ($P_{\textbf{MANUAL}}$) for queries with negation**

Year	Q	$P_{\textbf{MANUAL}}$	Change
2008	0.2706	0.2818	+ 4.1%
2009	0.2424	0.2561	+ 5.7%
2010	0.3025	0.3204	+ 6.0%

We would like to mention that similar improvements are observed for the baseline runs (without feedback) as well.

Figures 4.1, 4.2, and 4.3 provide detailed profiles of the per-query changes in AP when negation detection and removal are done manually. We make two important observations:

- The performance of about 75% of queries where negative constraints are present improves if those *negatives* are summarily removed (see Table 4.5).
- The improvement is typically overwhelmingly greater than the deterioration , i.e., in general, average improvement is much higher than average drop in performance. Figures 4.1 - 4.3 also show that sometimes the score shoots up by much more than 100%. On the other hand, degradation, whenever it occurs for a few queries, is not more than 70%.

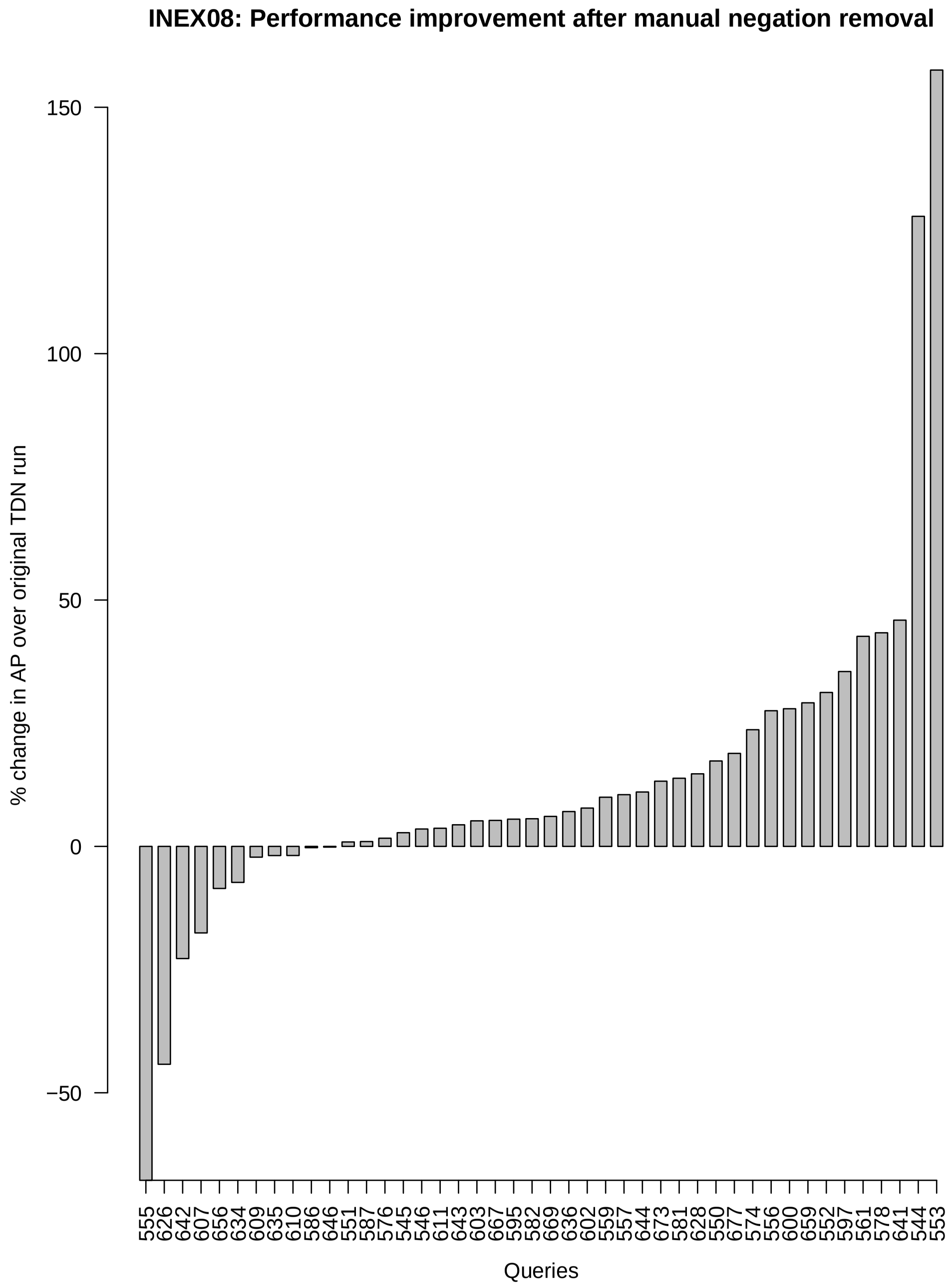


Figure 4.1: INEX 2008: Performance change when negation removed

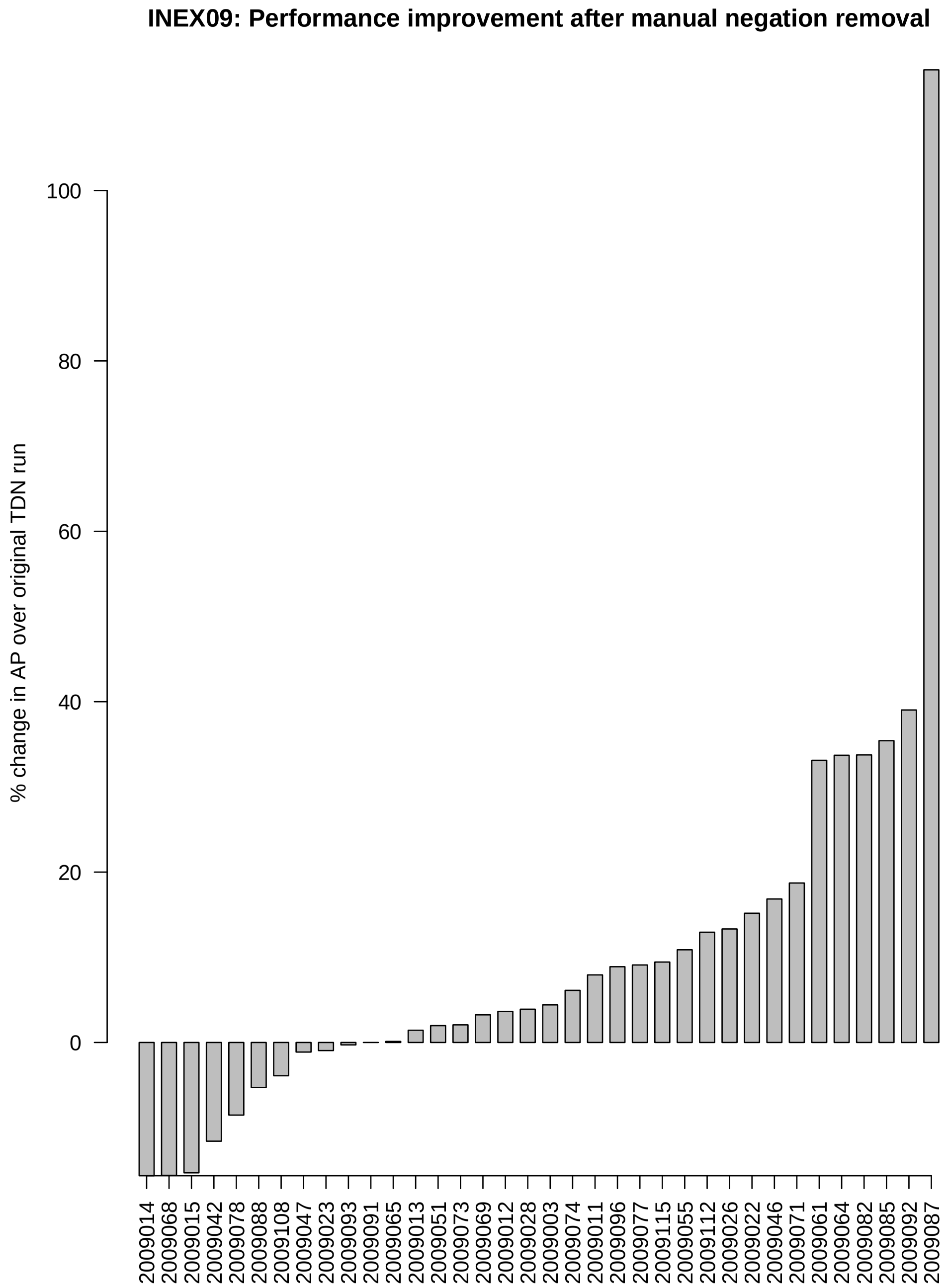


Figure 4.2: INEX 2009: Performance change when negation removed

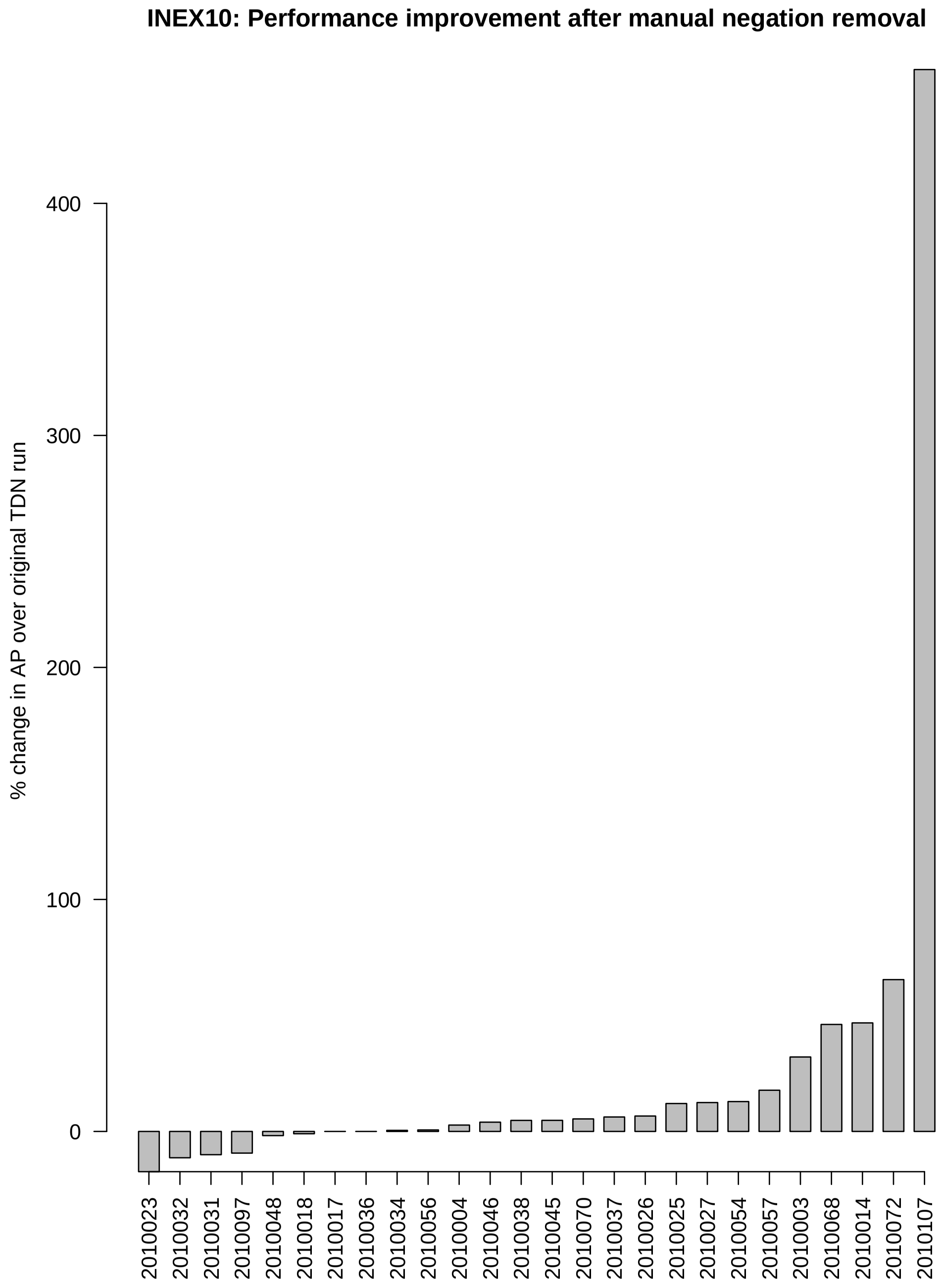


Figure 4.3: INEX 2010: Performance change when negation removed

Table 4.5: **Per-query performance of P_{MANUAL} over Q**

Year	# queries when performance in		
	$P > Q$	$P = Q$	$P < Q$
2008	33	0	11
2009	25	1	10
2010	18	2	6

The improvements obtained by using P_{MANUAL} over Q for each of the years (2008 - 2010) were found to be statistically significant using a paired Wilcoxon test at 95% level of significance (we took a non-parametric test since the underlying distribution did not seem to be normal).

4.4.2 Automatic Separation

The fact that negation removal significantly increases the performance for a substantial number of INEX queries sets the premise for our attempt to automate the process of negation detection and negation removal from the narratives of INEX topics.

We keep the title (T) and description (D) untouched as in the original query since negative constraints, when they are present, mostly occur in the narrative (N) part. The narrative for each query is broken into a set of sentences. These sentences are classified into either positive or negative by the Maximum Entropy Classifier based on the initially learned data.

Table 4.6: **Classifier performance**

Test set	Training set	# of training sentences	Accuracy
2008	2007	589	90.4%
2009	2008	679	89.1%
2010	2009	516	93.8%

Table 4.6 shows the performance of the classifier. Though the classifier identifies most of the sentences correctly (accuracy around 90%), there are some sentences which are wrongly classified (the number of *false positives* and *false negatives* together comprise about 10%). Hence, some

queries in Q, which have negation, and can be found in P_{MANUAL}, may not be included in P_{MAXENT}. Also, there are a few queries which appear in P_{MAXENT} but not in P_{MANUAL} because of *false positive*s. Note that Table 4.6 shows the baseline performance of the classifier. Accuracy will increase when the size of the training set is increased (e.g. when 2009 queries is trained with 2007 & 2008 queries and 2010 query set with that of 2007 - 2009).

Table 4.7: **MAP of the automatically detected positive (P_{MAXENT}) over original (Q) query sets for queries with negation**

Year	$\lvert P_{\text{MAXENT}} \rvert$	Q	P_{MAXENT}	**Change**
2008	31	0.2638	0.2748	+ 4.2%
2009	30	0.2573	0.2581	+ 0.3%
2010	20	0.2922	0.2983	+ 2.1 %

The document level retrieval performance for a blind feedback-based run using the automatically separated queries is given in Table 4.7. Note that the average scores are slightly different from those in Table 4.4. The difference crops up because of the difference between the sets P_{MAXENT} and P_{MANUAL}.

Table 4.8: **Per-query performance of P_{MAXENT} over Q**

Year	**# queries when performance in**		
	$P_M > Q$	$P_M = Q$	$P_M < Q$
2008	23	0	8
2009	19	0	11
2010	12	0	8

We also compare the performance of automatic separation vis-a-vis manual separation query-by-query (Table 4.8 and Figures 4.4, 4.5 and 4.6). For most of the common queries (i.e. queries in $P_{\text{MANUAL}} \cap P_{\text{MAXENT}}$), automatic and manual separation yield comparable results.

Indeed, the small difference between the performance of P_{MAXENT} and P_{MANUAL} is not found to be statistically significant (Wilcoxon test at 95% confidence interval run on $P_{\text{MANUAL}} \cap P_{\text{MAXENT}}$).

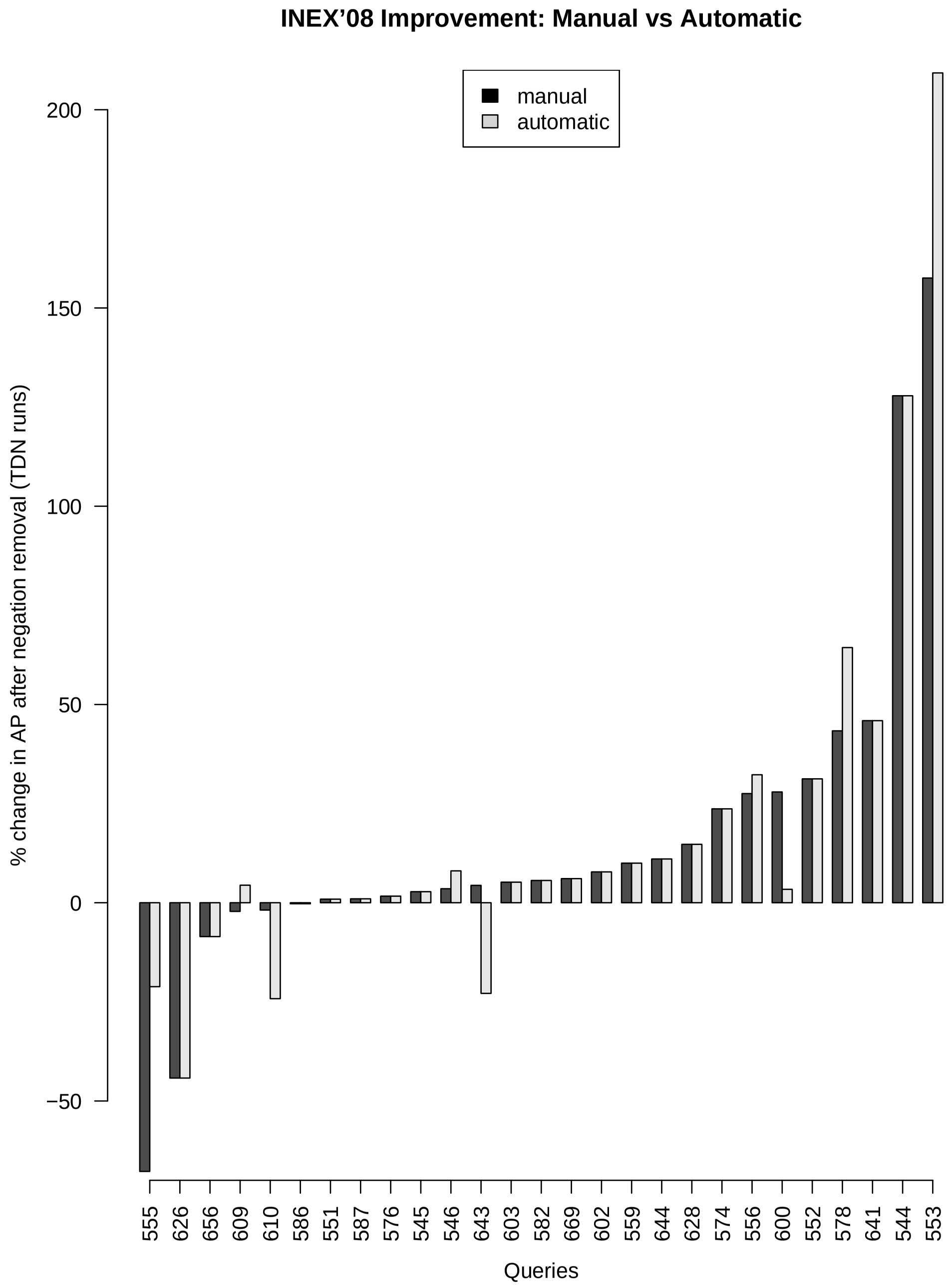


Figure 4.4: INEX 2008: Performance change when negation removed

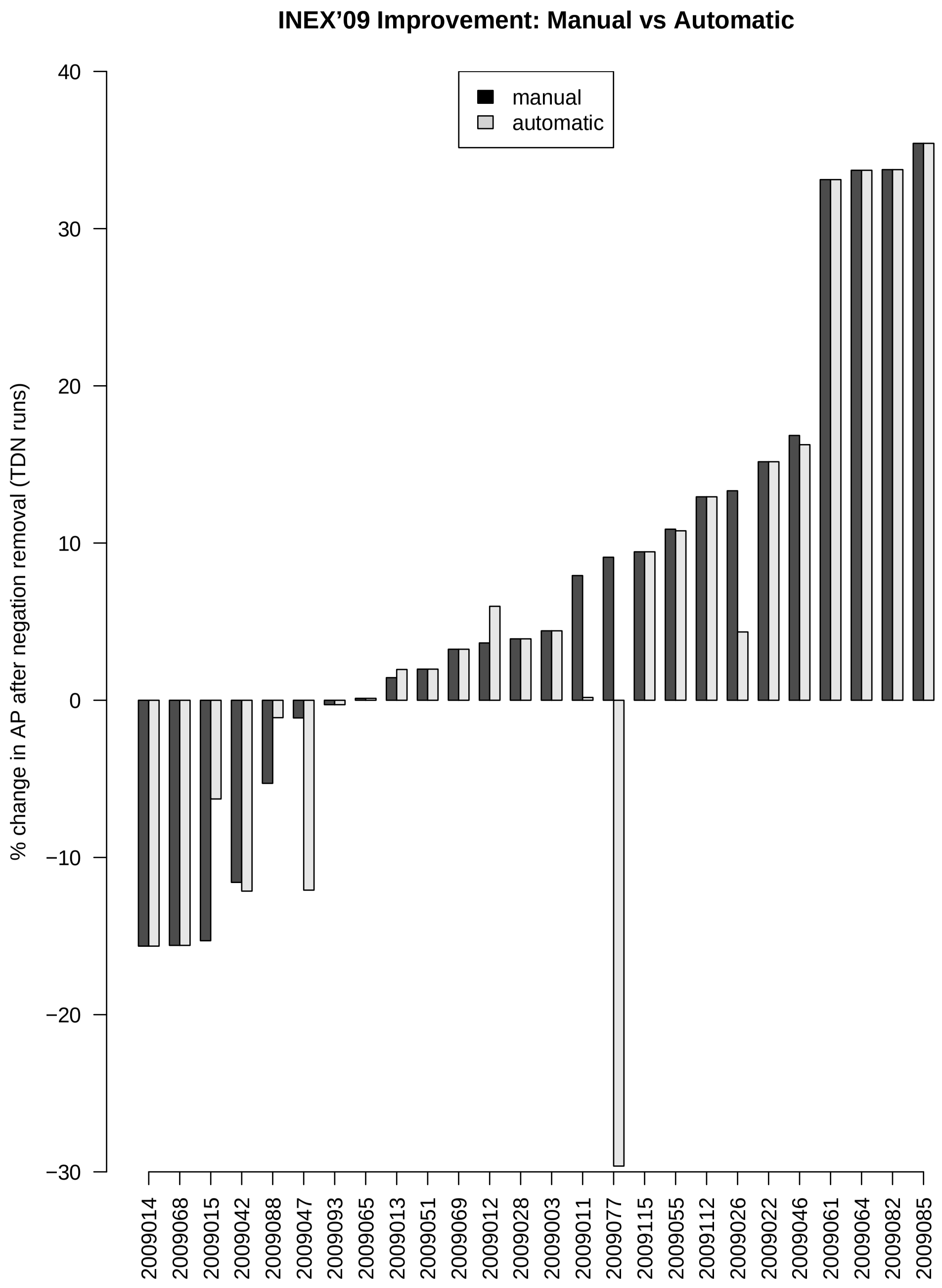


Figure 4.5: INEX 2009: Performance change when negation removed

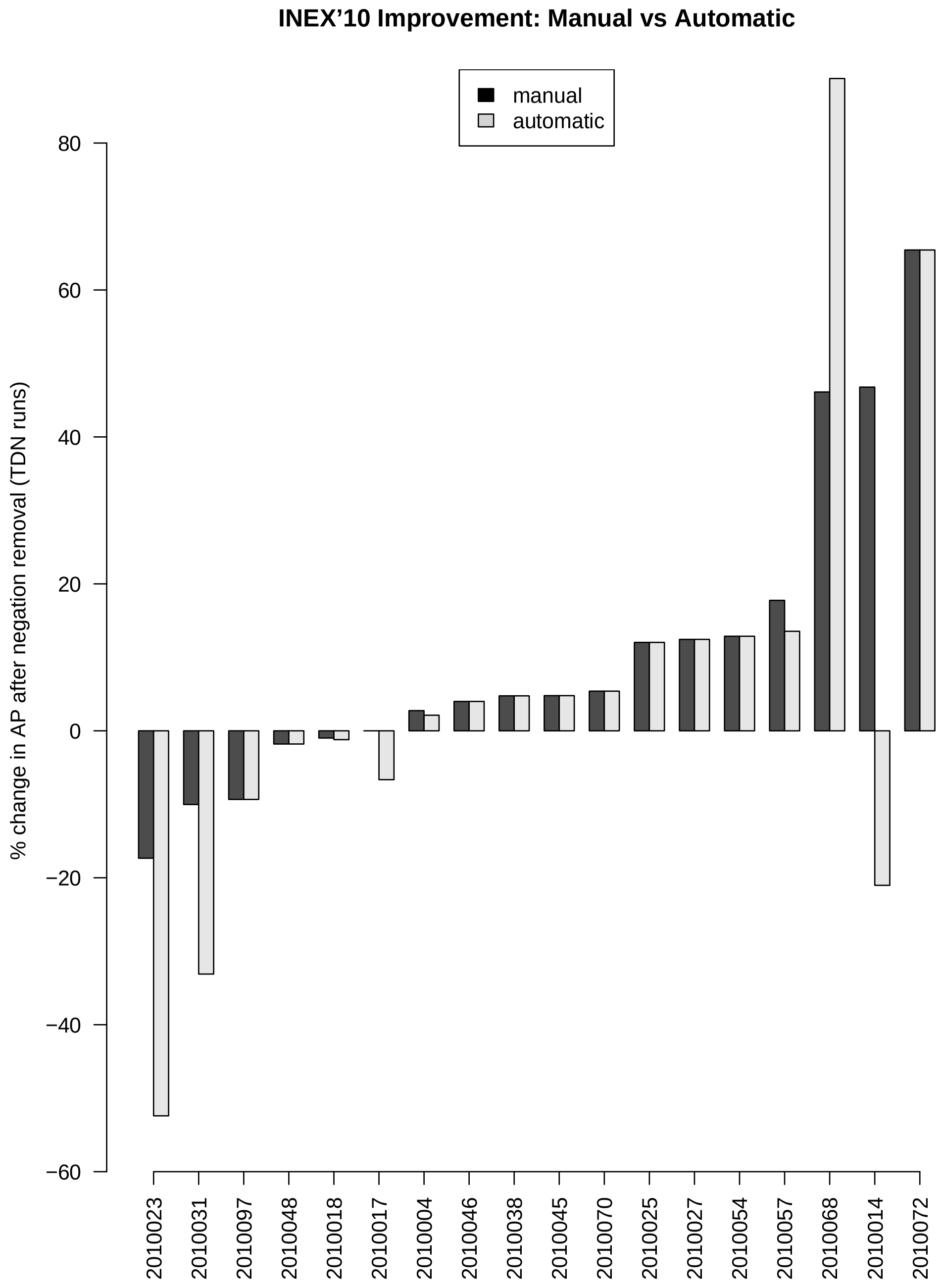


Figure 4.6: INEX 2010: Performance change when negation removed

Finally, the effect of automatic negation removal over all the queries present in the qrels, irrespective of whether a query contains negation or not, is shown in Table 4.9.

Table 4.9: **Performance for original query set (Q) and the automatically processed (P_{MAXENT}) over complete qrels**

Year	MAP (#rel-ret) with Q	MAP (#rel-ret) with P_{MAXENT}	Change in MAP
2008	0.3032 (3988)	0.3081 (4006)	+ 1.6%
2009	0.2496 (3613)	0.2499 (3608)	+ 0.1%
2010	0.2657 (3802)	0.2680 (3861)	+ 0.8%

We note that the overall impact of automatic removal of negation is brought down (compared to Table 4.3) by the large number of queries where either negation was not present or could not be detected automatically. Nevertheless, despite very small overall improvements in terms of MAP, the increase in the number of relevant documents found is comparable to that for manually separated queries (cf. Table 4.3). This is particularly important for element retrieval, since a good document retrieval with high recall provides a strong foundation for improved element retrieval (as observed in Chapter 3).

4.5 Limitation and Future work

It is generally observed that longer, verbose queries improve retrieval results and overall retrieval scores are found to increase as query length is progressively increased in T, TD, TDN runs respectively. In our approach, query length decreases from Q to P_{MANUAL} or P_{MAXENT}. Inspite of this shortening of length, the fact that retrieval performance significantly improves, unfolds a subtle but important caveat: query-length increase ensures improvement in retrieval results *only if* there is no negative terms. Presence of negation, on the other hand, will be counter-productive.

In case of automatic separation, the results reported here only shows the baseline improvement in retrieval performance as we used a single year's query set as training data. We believe that the classifier's performance as well as that of retrieval will improve if size of training data is increased.

However this hypothesis is not tested and can be verified in future.

In the entire demonstration in this chapter, we have simply discarded the negative constraints identified either manually or automatically. We believe, however, that the negative information can be used more proactively by a retrieval system to further improve performance. For example, the query may be modified suitably by assigning proper weights to different kinds of query terms: positive, negative and weak. How to automatically identify and weight these terms can be explored in future.

Finally, we have focused only on the verbose INEX adhoc queries (specifically queries from INEX 2008-10). This work thus appears to apply only to scenarios where users formulate precise and detailed queries corresponding to their information needs, rather than in a Web-search-like setting where the overwhelming majority of queries are very short and correspond to casual information needs.

4.6 Conclusion

A naive user finds it easier to formulate a complex information need in natural language rather than using terse keyword queries or using a formal query language. In XML retrieval this is particularly important because search queries are often very specific, and targeted towards finding precise document components. At INEX, we observe that topic creators often vividly state, besides the information need, what she *does not* want in the verbose narrative section. Most text retrieval systems do not handle this *negative information* properly. In this chapter, we investigate whether this information can help to improve retrieval performance. We first manually remove negative information from the INEX queries and observe that retrieval performance improves by 4-6% across different INEX collections compared to using the original queries containing negation. The improvement is found to be statistically significant. We also explore whether the process of negation detection and removal could be automated. With the help of an off-the-shelf maximum entropy classifer, we segregate positive and negative sentences in the narrative section of INEX topics. We form a set of positive queries taking only the positive sentences. This query set yields *equivalent* retrieval performance compared to the manual method (difference not statistically significant). The improvement we achieve needs to be viewed in the backdrop of diminishing effect due to query-length shortening. Generally longer queries improve retrieval performance which also holds true in XML domain. But the positive queries we use here are shorter than whole queries. Our performance gain over the whole queries in spite of this diminishing effect proves the merit of negation removal. However we believe that negative information can be used to further fine-tune queries, which can potentially yield further improvements.

Chapter 5

Pool Sampling

Since 2007, INEX has used a set of precision-recall based metrics for its adhoc tasks. Our aim in the next four chapters, is to investigate the reliability and robustness of these retrieval measures, and of the INEX pooling method. We investigate four specific questions.

1. How reliable are the metrics when assessments are incomplete, or when query sets are small?

2. What is the minimum pool / query-set size that can be used to reliably evaluate systems?

3. Can the INEX collections be used to fairly evaluate 'new' systems that did not participate in the pooling process? And,

4. for a fixed amount of assessment effort, would this effort be better spent in thoroughly judging a few queries, or in judging many queries relatively superficially?

First we study the sensitivity and stability of different precision-recall based metrics under incomplete assessments (Chapter 5). We simulate incomplete assessments using two different techniques. Keeping the number of queries unchanged, the set of judgments is reduced first *randomly* and then *systematically* per query.

Our findings validate properties of precision-recall-based metrics observed in document retrieval settings. Early precision measures are found to be more error-prone and less stable under incomplete judgments. However, system rankings are more affected when the incomplete assessments are obtained *randomly*, but remain largely unaffected if the incomplete (or partial) assessments are done *systematically*.

5.1 Introduction

INEX [1], set up in 2002, has been responsible for creating a Cranfield-style infrastructure for evaluating the effectiveness of content-oriented XML IR systems. INEX provides large test collections, topic sets and relevance judgments. As at traditional document retrieval evaluation forums, the relevance assessments used for evaluation at INEX are based on a pool generated from results submitted by participants. However, as the retrieval unit for XML search systems can be an element of arbitrary granularity and length, evaluation has been a challenge at INEX. Evaluation measures used in traditional IR, where a whole document is typically considered either relevant to a user query or not, are no longer tenable as the aim here is to locate the most relevant document-part(s) and not complete documents. Various evaluation measures have been tried over the years at INEX. The official metrics used at INEX 2002 [41] (calculated by the `inex_eval` program) were modified for INEX 2003 and 2004 [43] (cf. the `inex_eval_ng` program). Again, at INEX 2005, three new *Cumulated Gain*-based [51] metrics were adopted as the official metrics [60, 61]. These metrics were also used at INEX 2006. Since 2007, however, an arbitrary passage that may span more than one XML element has also been accepted as a valid retrievable unit for the focused adhoc task. This modified definition of the task necessitated a metric that could be used to evaluate both passage-retrieval and element-retrieval systems in the same manner. This gave rise to a family of metrics that were derived from the traditional interpolated *precision-recall* metrics. However, these metrics are defined in terms of text-length expressed in characters, rather than the number of documents that are retrieved and/or relevant (see Sec. 2.6.6 for details). Five of these metrics, viz. $iP[0.00]$, $iP[0.01]$, $iP[0.05]$, $iP[0.10]$ and AiP, were used in the official reports for the focused adhoc tasks. Among these, $iP[0.01]$ was taken as the official measure to rank the competing systems [54].

Since these measures are extensions of their counterparts in the standard document retrieval setting, they may be expected to have similar properties. However, the evaluation set-up used at INEX is markedly different from that used at other forums in certain ways.

- First, at INEX, retrieval granularity is at the XML element or passage level, but pooling is done at the document level. Thus, even when a small passage is retrieved from an article, the complete article is included in the pool. This leads to considerable diversity and robustness in the pooling.

- Second, TREC generally uses a fixed pool-depth (typically the top 100 documents from each contributor are pooled) and pools vary in size across queries. In contrast, INEX pools all runs (both valid and invalid) using top-n pooling and a fixed pool-size, i.e. the pool-depth is dynamically chosen for each query so that a pool size of around 600 documents is reached. The dynamically chosen pool-depth, which is at least 30 in terms of articles (and substantially higher in terms of elements), is assumed to be deep enough to cover a large fraction of the relevant articles for the majority of topics.

- Third, relevance in the XML domain is defined at the sub-document level. A relevance judgment file (or *qrels*) contains more information than just a boolean indicator about whether a document is relevant or irrelevant for a given topic. The qrels lists, for each topic, the documents that contain relevant passages, and precisely specifies the relevant elements and/or passages within each document. Relevant elements are identified by their *xpath*[1] and relevant passages either by a combination of *xpath* and start and end positions, or simply by their length and character-offsets from the beginning of the article. This relevance judged pool or qrels is then used for evaluation.

- Finally, there are no independent and dedicated topic creators or assessors at INEX. Participants are responsible for creating search topics and assessing the pools generated from submissions, with the assessment for a particular topic being generally done by the participant who created the topic. Since this job is a voluntary service by the participants, we need to keep the assessment effort to the minimum possible. From an operational point of view, this is possibly the most important difference between INEX and other evaluation forums.

[1] W3C, XPath-XML Path Language(XPath) Version 1.0, http://www.w3.org/TR/xpath

Given these differences, it is an open question whether the INEX metrics indeed have similar properties with regard to reliability and robustness as their document retrieval counterparts. There are several issues related to the chosen metrics that we investigate in our evaluation-related work. One of the questions that we specifically attempt to address in this chapter is: how reliable are the various metrics in ranking competing systems when assessments are incomplete (i.e. when some relevant documents have not been included in the judged set, and have therefore been assumed to be non-relevant)? On a related note, what is the minimum pool size that can be used to reliably evaluate systems?

Our experiments and analyses are restricted to the INEX focused task. The task allows us to look at early precision ($iP[0.00]$, $iP[0.01]$, $iP[0.05]$, $iP[0.10]$) as well as overall performance (AiP, $MAiP$) of a set of systems. The early precision metrics used in the task are known to be unstable and they need to be used with caution when comparing the performance of systems [34, 52]. Thus, in a sense, this task is the 'weakest link' among the INEX tasks, and needs to be thoroughly investigated. Further, for many groups, the focused results form the basis for submissions to the other adhoc tasks, suggesting that our choice of the focused task is a reasonable one.

In the next section, we review past work that provides the background for the rest of the chapter as well as for other evaluation related chapters. Section 5.3 describes our experimental set up and design of two different experiments. The results and observations are detailed in the context of two different approaches in the next two sections. Section 5.6 presents a comparative analysis. We conclude the chapter in Section 5.7.

5.2 Previous Work

Evaluating evaluation metrics and methodologies has a well-established history in the field of document retrieval. In the context of TREC, Zobel [158] examined the fairness and trustworthiness of the pooling-based evaluation methodology of IR experiments.

Buckley and Voorhees [12] proposed a novel way to examine the accuracy of various evaluation measures and validated a number of traditional rules of thumb that address issues such as the minimum number of queries required for reliable evaluation, which measures to use, and the notion of “significant” difference in the scores between two competing systems.

The above work was extended with the study of evaluation measures under incomplete and imperfect relevance judgments [13]. The authors examined the stability of system rankings produced by a metric when the size of the relevance-judged pool is gradually reduced, as well as when the topic-set size is reduced. They showed that MAP is both stable and discriminatory for evaluating document-level retrieval from a static document collection.

Sakai and Kando [113] studied the effect of incomplete assessments on different metrics like *b-pref*, RBP (Rank-Biased Precision, proposed by Moffat and Zobel [84]), Q-measure, AP, $nDCG$ and their *condensed-list* variants. The *condensed-list* variant of a metric is calculated using a ranked list obtained by removing unjudged documents from the original ranked list. The *condensed-list* variants AP', Q' and $nDCG'$ consistently performed better than either RBP or their original counterparts under unbiased incompleteness in terms of discriminative power and ranking stability. However, under shallow pooling or *pool-depth bias*, AP, $nDCG$ and Q-measure are better than their condensed-list variants [111].

The stability of system rankings in the above studies was mainly measured by Kendall’s rank correlation coefficient (τ). According to a widely used rule of thumb, two rankings are considered to be similar if their τ is 0.9 or higher. Sanderson and Soboroff [122] warn that the threshold of 0.9 should be taken with caution. In more recent work, Yilmaz et al. [156] showed that τ penalizes ranking differences at both high ranks and low ranks equally. However, the IR community is usually more concerned about differences in the top ranks than those at the bottom. Their proposed coefficient, AP correlation (τ_{AP}) gives more weight to differences at high rankings. It yields smaller values than Kendall’s τ for differences in the top ranks, but equals τ when the differences are uniformly distributed over the entire ranked list. To the best of our knowledge, there are no reported rules of thumb regarding the range of values of τ_{AP} for which two rankings may be regarded as essentially the same. Therefore, it is prudent to use τ_{AP} along with τ, rather than using either one alone.

All the work discussed above was based on document-level retrieval using mainly TREC/NTCIR data. Kazai and Lalmas [60] first studied the evaluation of XML retrieval. Their work used the XCG-based metrics (e.g. *MAep*, *nxCG*, *MAnxCG*, etc.) and some other older metrics like Q, R and `inex_eval`, with the INEX 2004 submissions (where only XML elements were permissible as units of retrieval).

Trotman et al. [137] experimented with the INEX pooling and assessment strategies during the INEX 2006 workshop. One of the questions addressed in their work was whether the INEX pool can be reduced in size using a shallow random pool (around 100 documents, taken in alphabetical order, per topic for 15 topics). This experiment was done for the Relevant-in-Context task, but was not comprehensive or conclusive. The correlation between system-rankings using the shallow pool and the original pool was highly positive (Spearman's rank corr $= 0.97$) on one hand; but for the ten best systems, it was near zero (Spearman's rank corr. $= -0.03$).

Piwowarski et al. [100] provide details about the INEX pooling and assessment exercises and their evolution during 2002-2006, along with an in-depth analysis. The authors explain why INEX shifted from an element-level pooling strategy to a document-level pooling strategy, and show that a pool containing the top-ranked 500 distinct documents per topic is large enough to ensure stable evaluation.

However both these studies used a different experimental setup. INEX has evolved much in several respects since then. In 2006, the test collection was changed — a collection of technical articles published by IEEE was replaced by a January 2006 dump of the English Wikipedia. The evaluation metrics have also changed at regular intervals. Prior to 2007, only XML elements were considered retrievable units at INEX. Handling overlap among the elements was an issue during this period. Since 2007, the retrieval of arbitrary, non-overlapping passages was permitted, and a new set of evaluation metrics was introduced. The track overview papers [34, 52] discuss evaluation results related to the Focused, Best in Context, and Relevant in Context tasks of the adhoc track. The authors found that the official metric used for the focused task ($iP[0.01]$) was unstable; further, no significant differences were found among the 10 best systems based on the metric (one tailed t-test, 95% confidence level). However, no information was provided about the other reported metrics, and to our knowledge, there are no reports of any analysis of their characteristics in the context of

XML retrieval. Similarly, there is no reported study of the robustness of the INEX pool, which is unique in several respects.

In the next few chapters including this one, we present a study of these new metrics of XML retrieval using the INEX 2007 and INEX 2008 collections. Our motivation is to analyse the general characteristics of the measures to find out which of the measures is the most robust, stable, and least erroneous in reliably comparing a set of XML retrieval systems. We are also interested in examining the pool creation process, and in looking at the trade-offs between the effort involved in creating the relevance assessments and the quality of the resultant collection. For example, we are interested in quantities like the minimum number of topics that should be used, the minimum number of documents to be judged per query, the minimum pool depth to be used, etc. We believe these observations can help in building test collections that use a much larger corpus, where the completeness assumption in Cranfield-based pooling is likely to be seriously challenged and possibly compromised.

Our work in this chapter is in line with the earlier work of Buckley and Voorhees [12, 13]. In most of our comparative analyses, our objective is to look at the stability of system rankings, rather than the absolute values of the various measures. Changes in ranking are quantified using the conventional Kendall's τ, as well as the more recently proposed τ_{AP} measure proposed by Yilmaz et al. [156]. Our results validate observations from the document retrieval domain in the context of focused retrieval, thus underlining the intrinsic properties of the metrics used.

5.3 Design of Experiments

As discussed in Section 5.1, INEX does not use a fixed pool depth for all queries. Instead, an assessor judges about 600 documents per topic. Our aim in this chapter is to study how results are affected if the assessment effort is reduced by judging a smaller pool of documents per topic. Naturally, when fewer documents are judged, the absolute values of various metrics will change. If the relative ranks of various runs remain largely unaffected, then the smaller set of assessments can still be used for evaluation. By progressively reducing the pool size, we can estimate the minimum amount of effort that yields results comparable to the current results.

Assessment effort can be reduced in two ways. First, the pool is generated as usual, but the assessors do the judgments on a best-effort basis. In this scenario, the pool for a particular topic may end up being partially judged. In the second case, the reduced pool size is fixed a priori, i.e. a smaller pool is created at the outset, and given to assessors. The following sections describe experiments that study how evaluation results are affected in these two scenarios.

5.3.1 Test Collection

We use the INEX 2007 and 2008 adhoc test collections in our experiments. These test collections consist of a corpus with an XML-ified version of the English Wikipedia as described in Table 2.2 of Chapter 2. For INEX 2007, though original topic set contained 130 queries (Table 2.4), but relevance judgments were available for only 107 topics (Table 2.5), so the remaining 23 queries were not part of our experiments. Similarly, for INEX 2008, the topic set consisted of 135 queries (544-678) but relevance judgments were available for only 70 queries.

The focused task of the adhoc track expects participating systems to return, for each topic, a ranked list of non-overlapping document parts (either passages or XML elements) that are most focused with respect to the information need expressed in the topic. For 2007, among the submitted runs, 79 were reported in the INEX 2007 website as valid runs. For 2008, this number was 61. Each such run was supposed to retrieve 1,500 passages or elements per topic, and list them in decreasing order of their relevance to the topic. The effectiveness of a strategy for a single topic is computed as a function of the ranks of retrieved and relevant texts and their relative lengths. The effectiveness of the strategy as a whole is then computed by taking into consideration its effectiveness across all the topics.

5.4 Random Sampling

The first set of experiments that we did may be taken to correspond to the following scenario. Pools are constructed in the usual way, and distributed to participants, but a participant is not able to assess all documents assigned to her. Can the partial assessments be used for evaluation? To simulate this situation, a random fraction of the qrels is discarded — these entries are regarded as unjudged and therefore assumed to be non-relevant — and the reduced qrels are used for evaluation. Our aim here is to see how small the random sample can be so that overall evaluation reliability is not compromised for a given set of queries.

5.4.1 Experiments

The experiment is designed as follows. First, 80% of the relevant documents for each query are selected at random from the original qrels without replacement.[2] All adhoc focused runs from both INEX 2007 and 2008 are evaluated using this reduced set of assessments, and ranked on the basis of each metric in turn. Rank correlation (both τ and τ_{AP}) values are computed between these new rankings, and the ranking produced by the corresponding metric with the original (100%) pool. The process is repeated with 10 different random samples. The entire exercise is then repeated at 60%, 40% and 20% sampling levels.[3] From the INEX 2007 adhoc focused task, 107 topics and 78 runs were used, while from the INEX 2008 focused task, there were 70 topics and 61 runs.

These experiments also address a flaw in our earlier experiments reported in Pal et al. [91], where random samples were chosen directly from the entries in the qrels. Since each entry specifies relevance for a single element, it was possible for a sample to include a relevant element from a document, but exclude another relevant item from the same document (which would then be regarded as non-relevant). This is an unrealistic situation, since judgments are done one document at a time, rather than one element at a time, i.e., an assessor is given a whole document for assessment, and (s)he highlights all the relevant passages/elements in it. Thus, given a particular document, all its relevant items should either figure in the pool, or be excluded from the pool.

[2] Though the original qrels contain assessed non-relevant units as well, these entries do not figure during the computation of precision-scores, and are therefore ignored in these experiments.

[3] For a few topics there were less than 5 relevant documents. For such topics, one relevant document was included in the reduced qrels at 20% sampling level.

Table 5.1: **INEX 07: Stability of system rankings for random pool sampling (107 topics, 78 systems, 6,460 rel. docs. in 100% qrels)**

Pool%	$iP[0.00]$ Avg.		$iP[0.01]$ Avg.		$iP[0.05]$ Avg.		$iP[0.10]$ Avg.		$MAiP$ Avg.	
(#reldocs)	τ	τ_{AP}	τ	τ_{AP}	τ	τ_{AP}	τ	τ_{AP}	τ	τ_{AP}
20 (1,298)	0.65810	0.64635	0.62234	0.66353	0.67128	0.68743	0.72921	0.73541	0.82164	0.80468
40 (2,589)	0.74463	0.74416	0.72792	0.76200	0.80305	0.78626	0.84945	0.84263	0.90080	0.88253
60 (3,871)	0.82469	0.81336	0.80086	0.81458	0.84878	0.83947	0.88492	0.87710	0.93062	0.91990
80 (5,164)	0.88633	0.88824	0.87556	0.88596	0.90913	0.90385	0.93584	0.92027	0.96450	0.95152

Table 5.2: **INEX 08: Stability of system rankings for random pool sampling (70 topics, 61 systems, 4,887 rel. docs. in 100% qrels)**

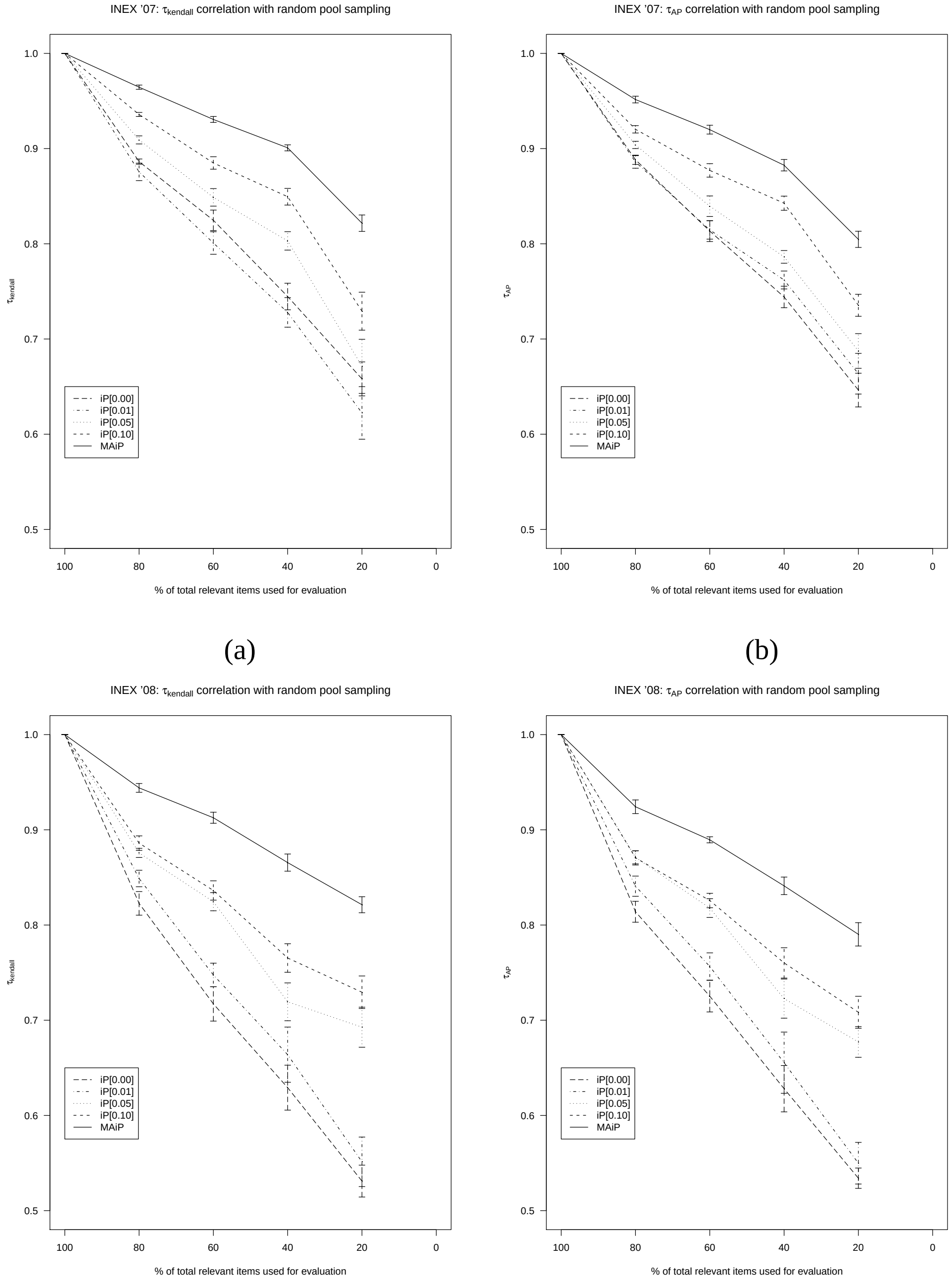
Pool%	$iP[0.00]$ Avg.		$iP[0.01]$ Avg.		$iP[0.05]$ Avg.		$iP[0.10]$ Avg.		$MAiP$ Avg.	
(#reldocs)	τ	τ_{AP}	τ	τ_{AP}	τ	τ_{AP}	τ	τ_{AP}	τ	τ_{AP}
20 (971)	0.53104	0.53407	0.55126	0.54982	0.69268	0.67718	0.72940	0.70835	0.82131	0.79022
40 (1,942)	0.62918	0.62809	0.66383	0.65540	0.71934	0.72270	0.76536	0.76021	0.86557	0.84114
60 (2,918)	0.71716	0.72525	0.74754	0.75644	0.82448	0.81782	0.83628	0.82575	0.91268	0.88951
80 (3,889)	0.82273	0.81393	0.84885	0.84080	0.87574	0.87112	0.88601	0.87057	0.94404	0.92422

5.4.2 Results

The means of the τ and τ_{AP} values across 10 random samples for each sampling level are shown in Tables 5.1 and 5.2. The same values along with the standard error at each sampling level are shown in Figure 5.1.

For all the graphs, as the sampling level decreases, the correlation between the original rankings produced by a metric and the rankings obtained with reduced assessments decreases in general, so each of the curves droops. One obvious reason is that with reduced assessments, the precision-score is affected non-uniformly across the systems, depending upon the ranks of retrieved relevant texts that are missing in the reduced pool. This phenomenon leads to changes in comparative ranks. Further, Kendall τ drops for $iP[0.00]$ and $iP[0.01]$ at a much faster rate than it does for $iP[0.05]$, $iP[0.10]$ or $MAiP$. Among the metrics, $MAiP$ clearly shows the least variation in τ values across different pool-sizes and across the samples at a particular pool-size.

Error-bars for each curve tend to increase as pool-size reduces. The reason can be attributed to the fact that at smaller pool sizes, the overlap among the samples reduces. This affects the precision



(a)

(b)

(c)

(d)

Figure 5.1: Rank correlation between original system rankings and rankings obtained with randomly reduced pools.

Table 5.3: **Minimum no. of reldocs required in the random pool to get $\tau \geq 0.9$ (INEX 2007: 107 topics, 6,460 reldocs in 100% pool; INEX 2008: 70 topics, 4,887 reldocs in 100% pool)**

Metric	INEX 2007	INEX 2008
$iP[0.00]$	$> 5,164$ ($> 80\%$)	$> 3,889$ ($> 80\%$)
$iP[0.01]$	$> 5,164$ ($> 80\%$)	$> 3,889$ ($> 80\%$)
$iP[0.05]$	$> 3,871, < 5164$ ($\sim 75\%$)	$> 3,889$ ($> 80\%$)
$iP[0.10]$	$> 3,871, < 5164$ ($\sim 65\%$)	$> 3,889$ ($> 80\%$)
$MAiP$	$< 2,589$ ($< 40\%$)	$< 2,918$ ($< 60\%$)

scores of different systems in a very irregular fashion. This irregularity causes widely varying system-rankings across the samples leading to wide variation in τ.

τ_{AP} values are, on the whole, slightly lower than the corresponding τ values. When a relevant document, selected by a few systems at high rank (it is likely that these systems are toppers) is randomly excluded from the pool, early precision scores for such systems drastically drop. For other systems which do not retrieve the document (many of these are among the poorly ranked systems), the scores remain unaffected. This leads to substantial changes at high positions of the system-ranking. Since τ_{AP} puts more weight on changes at the top ranks compared to τ (Kendall τ has the same weightage for changes in ranks across the entire list), it is affected to a greater extent than τ.

On a closer look, the curves in Figure 5.1(a) (INEX 2007) look smoother and more regular compared to their INEX 2008 counterparts (Figure 5.1(c)). There are two reasons for this: (i) the INEX 2007 dataset contains a larger number of valid runs (78 compared to 61 for INEX 2008) and (ii) the INEX 2007 qrels consist of a greater number of queries than the INEX 2008 qrels (107 topics compared to 70). The changes in τ-values are smoothed out through averaging over a higher number of systems, resulting in smoother curves. Similarly, rankings disagree to a greater extent when a smaller number of queries is involved. This leads to a sharp fall in τ-values, which is particularly acute for the early precision metrics ($iP[0.00]$ and $iP[0.01]$) in 2008 (see Figure 5.1(c)).

In summary, with about 50% sample of the qrels, system rankings at INEX 2007 and INEX 2008 are not significantly affected ($\tau \geq 0.9$) if $MAiP$ is used as the ranking metric. If $iP[0.10]$ is used to rank systems, τ remains above 0.9 for 65% samples in case of INEX 2007, and an 83% pool for INEX 2008. For the other metrics, even a 20% random reduction in the judged pool results in significant ranking changes (see Table 5.3).

5.5 Reducing Pool-depth

Our second goal is to estimate the minimum pool size that can be used to reliably evaluate a set of runs. First, pools of varying sizes are generated from a set of submissions by varying the pool depth for each query. Note that, in these experiments, a smaller pool is always a proper subset of a bigger pool. The maximum possible pool size is limited by the size of the original pool. Submissions are then evaluated on the basis of assessments generated from the reduced pools.

5.5.1 Experiments

Since some submission files were changed by participants after the pooling process was completed, we were not able to exactly replicate the original pool. We therefore take as our starting point a pool created from all valid and invalid submissions (98 for INEX 2007 and 76 for INEX 2008) in the focused category only. To create this pool, we guess the original pool depth (d_Q) for each topic Q as follows: d_Q is taken to be the minimum depth at which the number of distinct documents in the generated pool is greater than (or equal to) the original pool size for Q. The restriction of the original qrels to this generated pool is taken to be the initial (or 100%) qrels. Although this is actually a subset of the original qrels, it is a close clone (both Kendall's τ and τ_{AP} for system rankings obtained using the original qrels and our simulated 100% qrels are over 0.99 in most cases, with the minimum value being 0.97).

The reduced pools are also created in a similar way. The X% pool ($X = 5, 10, 20, \ldots, 90$) is generated by first guessing an appropriate pool depth ($d_Q^{(X)}$) for each topic Q. At this pool depth, the pool size for Q equals (or just crosses) X% of the original pool size for Q. We refer to the corresponding qrels as the X% qrels.

Table 5.4: **INEX 07: Stability of system rankings on reducing pool depth (107 topics, 78 systems, 5,610 rel. docs. in 100% qrels)**

Pool%	$iP[0.00]$		$iP[0.01]$		$iP[0.05]$		$iP[0.10]$		$MAiP$	
(#reldocs)	τ	τ_{AP}	τ	τ_{AP}	τ	τ_{AP}	τ	τ_{AP}	τ	τ_{AP}
5 (1,123)	0.89426	0.87066	0.85995	0.87111	0.84251	0.79643	0.85910	0.81760	0.91543	0.89109
10 (1,667)	0.94394	0.90673	0.91312	0.91531	0.86520	0.81984	0.88807	0.87771	0.94363	0.91281
20 (2,497)	0.97432	0.95700	0.93247	0.93669	0.90621	0.86521	0.91579	0.89033	0.95781	0.95560
30 (3,184)	0.98432	0.98250	0.95265	0.94696	0.93224	0.91840	0.94647	0.92976	0.96848	0.96489
40 (3,757)	0.99049	0.98733	0.95697	0.95258	0.94546	0.93777	0.96164	0.95646	0.98182	0.98316
50 (4,247)	0.99232	0.98894	0.95697	0.95343	0.95662	0.94926	0.97098	0.96050	0.98766	0.98420
60 (4,697)	0.99516	0.99464	0.96898	0.96580	0.96431	0.96574	0.97765	0.96333	0.99166	0.99219
70 (5,108)	0.99750	0.99643	0.97515	0.97138	0.97565	0.97408	0.98532	0.98273	0.99366	0.99402
80 (5,375)	0.99900	0.99784	0.97849	0.97895	0.97965	0.98495	0.98716	0.98741	0.99767	0.99711
90 (5,520)	0.99933	0.99827	0.98783	0.99030	0.98815	0.99149	0.99283	0.99178	0.99900	0.99857

All valid submissions are evaluated with the reduced qrels, and the correlation between the rankings obtained using the X% and 100% qrels — measured using Kendall's τ as well as τ_{AP} — are computed for each of the INEX metrics.

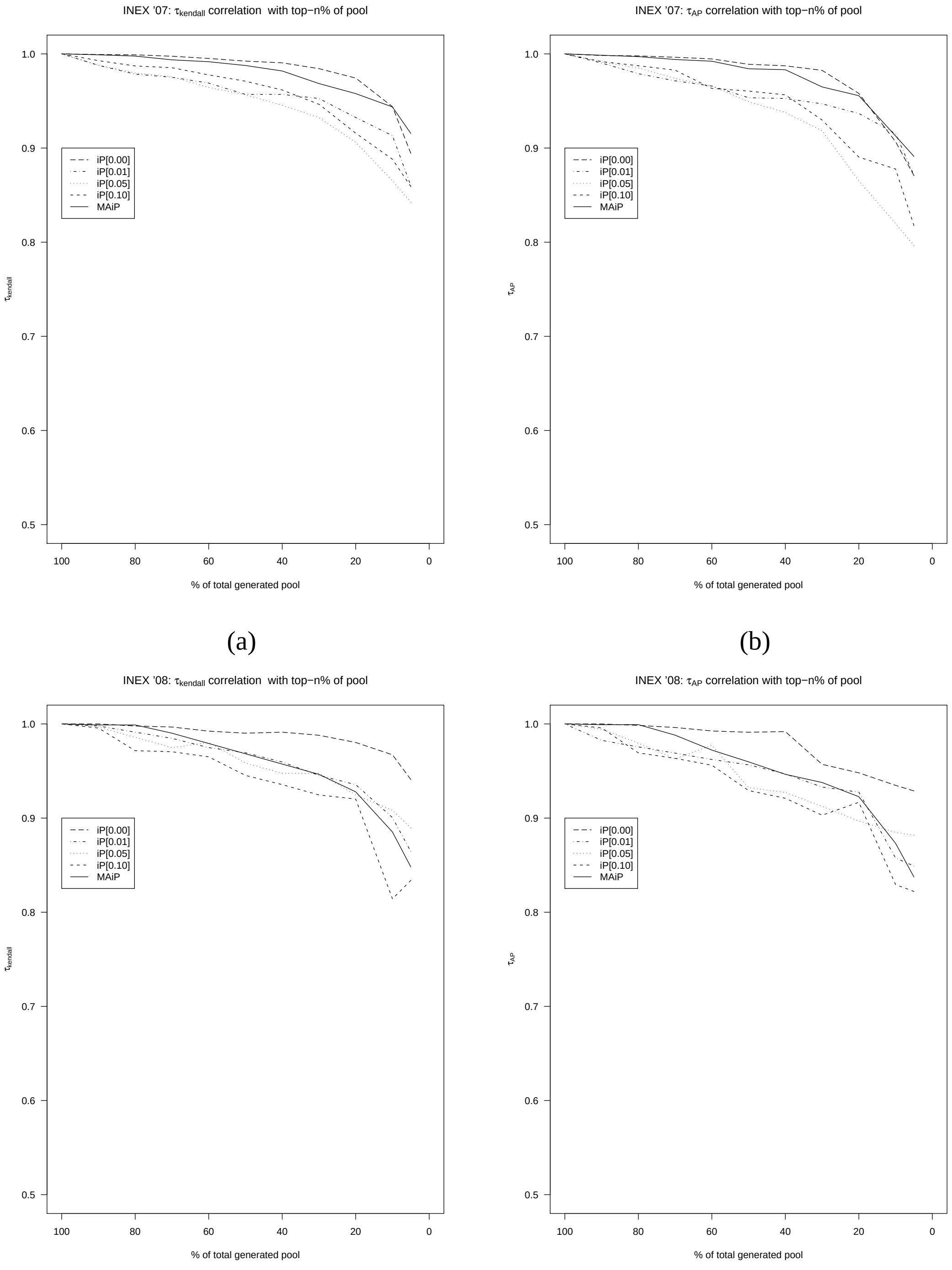
5.5.2 Results

Tables 5.4 and 5.5 show the variation in τ and τ_{AP} as X varies from 5 to 90. Both τ and τ_{AP} are very high (over 0.9 in almost all cases even at $X = 20\%$). This shows that system rankings are not significantly affected if a shallower pool is used for evaluation. Thus, the system rankings obtained using any of the INEX evaluation metrics is reasonably reliable even when just 20% of the original pool size is assessed. The fact that τ and τ_{AP} are in close agreement further signifies that ranking changes are more or less uniformly distributed over the entire ranked list, irrespective of the metric used for ranking.

Figure 5.2 displays the same information graphically. Two trends are visible from the graphs.

1. In general, correlation values decrease from $iP[0.00]$ to $iP[0.10]$.

2. Also, correlation decreases as shallower pools are used for evaluation.



(a)

(b)

(c)

(d)

Figure 5.2: Rank correlation between original system rankings and rankings obtained with reduced

Table 5.5: **INEX 08: Stability of system rankings on reducing pool depth (70 topics, 61 systems, 4,667 rel. docs. in 100% qrels)**

Pool%	$iP[0.00]$		$iP[0.01]$		$iP[0.05]$		$iP[0.10]$		$MAiP$	
(#reldocs)	τ	τ_{AP}	τ	τ_{AP}	τ	τ_{AP}	τ	τ_{AP}	τ	τ_{AP}
5 (850)	0.94098	0.92886	0.86448	0.84909	0.88962	0.88182	0.83388	0.82205	0.84809	0.83739
10 (1,307)	0.96721	0.93479	0.90055	0.85759	0.90820	0.88480	0.81421	0.82937	0.88525	0.87316
20 (1,992)	0.98033	0.94805	0.93552	0.92756	0.92459	0.89686	0.92022	0.91679	0.92787	0.92271
30 (2,530)	0.98798	0.95712	0.94536	0.93295	0.94754	0.91241	0.92459	0.90303	0.94645	0.93782
40 (2,960)	0.99126	0.99187	0.95956	0.94695	0.94754	0.92723	0.93552	0.92104	0.95738	0.94647
50 (3,386)	0.99016	0.99118	0.96940	0.95657	0.95847	0.93234	0.94536	0.92924	0.96831	0.95972
60 (3,754)	0.99235	0.99254	0.97486	0.96224	0.98033	0.97733	0.96503	0.95615	0.97924	0.97223
70 (4,094)	0.99672	0.99623	0.98470	0.96899	0.97486	0.96283	0.97049	0.96344	0.99017	0.98809
80 (4,380)	0.99781	0.99845	0.99126	0.97566	0.98579	0.97958	0.97159	0.96925	0.99891	0.99928
90 (4,584)	1.00000	1.00000	0.99781	0.98258	0.99672	0.994520	0.99563	0.99580	0.99891	0.99921

Precision values at early recall levels are generally determined by top-ranked documents. Even when shallow pools are used, these top-ranked documents are usually included in the smaller pool. The assessments for these top-ranked documents are therefore mostly unaffected whether we use shallow pool or the original pool. Thus, precision values at early recall levels do not change much as pool size is reduced, and the curves for $iP[0.00]$ and $iP[0.01]$ are relatively flat. On the other hand, a relevant document which is not highly-ranked by any of the systems will be excluded from the pool at smaller pool depths. Such a relevant document will then remain unjudged, and will therefore be considered non-relevant. If the rank of such a relevant document varies across systems (as is likely), the precision values at higher recall levels will be affected differently for different runs, and their relative ranks may change, leading to a drop in τ or τ_{AP}.

As pool size decreases, the number of such documents — which were marked relevant in the original qrels, but are regarded as non-relevant in the reduced qrels — increases, leading to greater discrepancies in ranking. $MAiP$ is an average of precision values at 101 recall levels, and is thus affected to an intermediate degree when pool size changes.

5.6 Discussion

Of the two methods that were tried in this chapter, reducing pool-depth is clearly the more logical, systematic and safe way to obtain smaller pools. However, in some cases, the judged pool may be small due to forces of circumstance, rather than by design. For example, at TREC, CLEF and FIRE, the pool of documents to be judged is given to an assessor in order of document ids. This is done to avoid any potential bias of assessors against low-ranked documents. In such a scenario, if an assessor ends up partially judging a topic, can the judgments be of some use? The Random Sampling section attempts to give a quantitative answer to this question. This approach also provides a baseline that highlights the usefulness of reducing pool size by reducing pool-depth.

5.7 Conclusion

We carry out a number of experiments with a common objective of finding a reliable, “optimum” evaluation setup in terms of the assessment effort required, and the evaluation measures used to rank a set of XML retrieval systems at INEX. This chapter demonstrates the first set of such experiments.

The issues we attempted to address here are summarized below along with our findings:

- *How reliable are the various metrics in ranking competing systems when assessments are incomplete?*

  The results of our experiments validate properties of *precision-recall*-based metrics that were originally observed in a document retrieval setting. For example, our experiments reaffirm that early precision measures ($iP[0.00]$, $iP[0.01]$) are more error-prone and less stable under incomplete judgments, whereas AiP is the least vulnerable among these metrics. Specifically, $MAiP$-based rankings remain largely unaltered ($\tau \geq 0.9$), even when evaluation effort is randomly reduced to half of the original.

- *On a related note, what is the minimum pool size that can be used to reliably evaluate systems?*

Our experiments show that rankings similar to the official rankings can be obtained even when pooling is limited to about 15% of the currently used pool depth. The following observation emphasises how small this pool really is. If we construct the top-10 pool (i.e., a pool consisting of the top 10 documents returned by each participating system) for the INEX 2007 task, it contains about 30% of the actual pool (with 184 documents being judged on average per query). Similarly, for INEX 2008, the top-10 pool constitutes 22% of the overall pool, with 135 documents being judged on average per query Thus, it is fair to say that the 15% pool mentioned above is smaller than even the top-10 pool.

Of course, it is important to keep in mind that such a small pool-size works well because evaluation at INEX is strongly dependent on a relatively small set of top-ranked results, and the measures used are mostly early-precision oriented (as are those considered in our study).

Chapter 6

Query Sampling

This chapter deals with the second set of experiments that are performed in our study of INEX adhoc task evaluation. We use the same experimental setting as in Chapter 5, but investigate the problem of finding an “optimum” evaluation setup from another angle. In order to investigate the reliability, sensitivity and stability aspects of different INEX adhoc metrics, the number of queries are *randomly* reduced, but the number of assessments per query remains unchanged. We study the changes in system rankings provided by different metrics under reduced query set sizes. Also, we attempt to *quantitatively* measure the errors that the metrics cause as the number of queries is reduced.

Combining and comparing the observations made in Chapter 5, we infer that for a fixed amount of effort, judging shallow pools for many queries is better than judging deep pools for a smaller set of queries. However, when judging only a random sample of a pool, it is better to completely judge fewer topics than to partially judge many topics. These findings corroborate results reported for the document retrieval scenario.

6.1 Introduction

As discussed in Chapter 5, a set of *precision-recall* based metrics were introduced in the INEX adhoc task from 2007. Though the metrics were adopted from the standard document retrieval setting, they were modified to suit the need of sub-document level retrieval. Also, the evaluation framework followed at INEX is markedly different from that at other evaluation forums in several ways (see Section 5.1 for details). A thorough study of the reliability, sensitivity and stability of these metrics for XML retrieval was thus needed.

In continuation of the study described in Chapter 5, we specifically investigate the following questions in this chapter.

- How reliable are the various metrics in ranking competing systems if the query set size is small? Unlike Chapter 5 we reduce the number of queries here, keeping the assessments per query unchanged. In particular, what are the *error rates* of the various metrics as query set size changes? (The error rate quantifies the chance of arriving at a wrong conclusion when comparing two systems using a particular set of queries.) What is the minimum number of queries that should be used to keep the error rates for the various metrics within a maximum allowable upper bound?
- For a fixed amount of assessment effort, would this effort be better spent in thoroughly judging a few queries, or in judging many queries relatively superficially?

We attempt to address these questions based on the official submissions to the INEX 2007 and INEX 2008 adhoc focused task.

In the next section, we review relevant past work that provides the background for the chapter (for other related work see Section 5.2) Our design of experiments is discussed in Section 6.3. The results and observations are detailed in the context of two different approaches in the next two sections. Section 6.6 presents a comparative analysis. We conclude the chapter in Section 6.7.

6.2 Previous Work

Since this chapter is closely related to the earlier chapter, most of the relevant work is discussed in Chapter 5.

In addition to pool sampling (Chapter 5), Buckley and Voorhees [12] also introduced the concept of an *error rate* for an evaluation measure. By repeating retrieval runs using different variations of the same query sets and comparing pairs of systems across query variations, they showed that Average Precision is a more stable measure than measures based on early precision. The authors studied the effect of topic set size on error-rates more extensively in [141]. Using the TREC results and Mean Average Precision (MAP) as the metric, error-rates were directly computed for topic sets of size upto 25. The rates were then extrapolated for larger topic set sizes. The study indicated a caveat when two retrieval systems are compared to determine one's superiority over the other, and cautioned that error-rates must be taken into consideration along with the number of topics. Sanderson and Zobel [123] extended the study on error-rates of evaluation metrics using MAP and $P@10$ in the light of significance tests, and found the bounds (lower and upper) of these error-rates. They observed that, given a set of relevance judgments, MAP is more reliable than $P@10$. To estimate the discriminative power of various IR metrics, Sakai [110] used the Bootstrap hypothesis test with NTCIR collections and compared it with the *swap*-method (method of finding error-rate introduced in [141]). Both approaches suggest that Average Precision is one of the best metrics, so far as discriminative power is concerned. The power of a metric can also be seen as the statistical power of a hypothesis test as proposed by Webber et al. [147]. The authors determine the minimum number of topics necessary to detect a certain degree of superiority of one system over another by estimating between-system score differences and their standard deviations. Interestingly, the paper also concludes that greater statistical power is achieved for the same relevance assessment effort by judging a shallow pool for a large number of topics rather than a deep pool for a small number of topics.

Carterette et al. [19] conducted a noteworthy experiment with topic-set size. Using the TREC Million Query Track data, the authors found that evaluation based on a large number of queries with fewer judgments is more cost-effective than, but equally reliable as, using fewer queries with more judgments.

All the work discussed above was based on document-level retrieval using mainly TREC/NTCIR data. So far as our knowledge goes, there has not been any study to investigate the properties of these new metrics in the XML retrieval domain. (Kazai and Lalmas [60] worked on XML evaluation metrics used at the early INEXes [2002-04] which had a smaller test collection comprising IEEE technical articles).

Our work in this chapter is in line with the earlier work of Buckley and Voorhees [12, 141, 13]. For error-rates of different measures we also considered [123]. To measure changes in system rankings, we used both Kendall's τ and τ_{AP} like in Chapter 5.

6.3 Design of Experiments

Since INEX does not have a dedicated pool of assessors and assessment is done by participants, some topics end up not being assessed completely, or at all. The initial topic set provided for run submission thus gets downsized after assessment, and this reduced topic set is used for evaluation (see Table 2.4 and Table 2.5). In the last chapter, we have studied the effect of partial assessment of the pool for a query. But is the reduced number of topics enough to ensure stable and robust evaluation of systems? What is the minimum number of topics required for reliable evaluation based on different metrics used in XML retrieval? How do these new metrics behave under reduced query size? We investigate these issues from two different angles.

In the first approach, we randomly reduce the size of the query set and study the changes in system rankings due to smaller query sets (Sec. 6.4). Secondly, we compute the *error* committed by different metrics on average when comparing systems at different query set sizes (Sec. 6.5).

we now turn to the details of the two approaches, the results and our observations.

6.4 Random Sampling

6.4.1 Experiments

We simulate smaller topic set by eliminating all assessment details for a randomly chosen subset of topics. The setup for this task is quite similar to that for *random pool sampling* (Sec. 5.4). First, a random 80% sample of the total set of queries in the qrels (107 for INEX 2007 and 70 for INEX 2008) is selected. For each selected topic, all available assessment information is considered. Once again, correlation is measured between the system rankings produced by each metric using the complete set of queries and the reduced query-set. The process is repeated for 10 random samples. The whole exercise is repeated with 60%, 40% and 20% of the query set.

6.4.2 Results

The behaviour of the metrics as query-set size varies is shown in Figure 6.1. The curves exhibit the same drooping nature as the query-set size is progressively reduced. Early precision measures ($iP[0.00]$ and $iP[0.01]$) perform poorly compared to late precision measures ($iP[0.05]$ and $iP[0.10]$). $MAiP$ emerges as a clear winner both in terms of its resilience to the reduction in size of the topic-set and variation across samples (smallest error bars).

Further, τ_{AP} values are in general slightly smaller than τ for $MAiP$ and $iP[0.10]$ for both INEX 2007 (Figures 6.1(a) and (b)) and INEX 2008 (Figures 6.1(c) and (d)), but this trend is not so prominent for early precision metrics.

The top-ranked systems at both INEX 2007 and INEX 2008 are very similar (see Figure 6.2, for example). Thus, for $iP[0.01]$, there is no statistically significant difference in the performance of the top 10 runs at INEX 2007 [34]. Paired t-test results for the top 10 systems at the INEX 2008 focused task are also similar [52]. Since these top-ranked systems are not significantly different to each other, even small changes in the qrels (due to removal of some topics) cause more swaps in the relative positions of the top-ranked systems than the low-performing ones. Since τ_{AP} puts more weight to swaps in the top ranks than in the low ranks compared to Kendall's τ, τ_{AP} values

Table 6.1: **Number of queries required to get $\tau \geq 0.9$**

Metric	INEX 2007	INEX 2008
$iP[0.00]$	86 (80%)	59 (84%)
$iP[0.01]$	86 (80%)	58 (83%)
$iP[0.05]$	75 (70%)	53 (75%)
$iP[0.10]$	59 (55%)	44 (62%)
$MAiP$	37 (35%)	32 (45%)

are smaller.

Between the two years, the curves for INEX 2007 are flatter (i.e. correlation with the official system rankings drops more slowly) than their INEX 2008 counterparts, one of the reasons being that INEX 2007 has a larger number of participating runs (78 compared to 61). Moreover, INEX 2007 curves also have smaller error-bars (range of τ-values is smaller) compared to INEX 2008 curves. This is most likely due to a higher number of queries at the same percentage point (the INEX 2007 topic set has a total of 107 queries, against 70 queries in the INEX 2008 set).

One important observation is that $MAiP$-based rankings remain largely unchanged ($\tau \geq 0.9$) even with only 37 queries (about 35% of the original number) for INEX 2007, and 32 queries (45%) for INEX 2008.

Table 6.1 shows the minimum number of queries required to obtain a τ value of 0.9 or greater with the original ranking, when ranking is done on the basis of the various INEX metrics.

6.5 Error Rates

The behaviour of the various metrics as topic set size changes is studied from a different perspective in this section.

6.5.1 Motivation

The measured effectiveness of a system, among others, depends very much on the query set used to measure effectiveness [6]. For a particular set of queries, system A may outperform system B, while for a different set of queries, their relative performances can be in the opposite order. However, if two such systems are evaluated using a large number of randomly chosen sets of queries, then it is expected that the "truly better" system will outperform the other for a majority of the query sets. The remaining cases, where the better system performs worse, can be regarded as *errors*. How well a metric captures the intrinsic quality of systems is reflected in how often it leads to an erroneous conclusion when used to compare two systems. The fewer the errors, the better is the metric. The motivation behind the experiments reported in this section was to study the error-rates of various metrics at different topic-set sizes, and then to estimate the minimum number of topics required to keep the error-rate within a stipulated limit. These experiments are based on the work of Buckley and Voorhees [141].

6.5.2 Computing Error Rates

The basic procedure to compute the error rate is as follows. We take two retrieval systems A and B, and two subset of the topic set of equal size z, and compute the value of a particular evaluation metric for each system-topic-set pair. The mean scores of the two systems are compared for each topic set. We then check whether the scores on the topic sets agree as to which of the runs is better. If they do not agree, i.e. A's score is higher than B's by at least a minimum margin p on one topic set, but B's score is higher (by at least the same minimum margin) on the other set, we mark this as a *swap* (or disagreement). By repeating the exercise n times with different topic sets of the same size, we calculate the proportion of swaps for a particular pair of runs. The average proportion of swaps over all possible pairs of runs is called the *error rate* for that particular topic set size.

Note that the subset of the topic set could be chosen in two different ways. The problem here is of *query sampling* either *with replacement* or *without replacement*. Initially we considered the first approach. This approach generally yields lower error rates since two topic sets of the same size may overlap, leading to a reduction in the error rate [123]. For the results reported here, the second approach (which ensures disjoint topic sets, and thus results in higher error rates) has been chosen. The core of the algorithm for calculating error rates is similar to that in [141].

```
foreach topic set size $z$ from 5 ... $numQ/2$ do
    Set all counters to 0;
    foreach $trial$ from 1 ... 50 do
        Select two disjoint topic-sets $A$, $B$ of size $z$;
        foreach diff $p$ from 0, 5, 10, 20, 30 do
            foreach run-pair $X, Y$ from INEX dataset do
                if ($M(X, A) \sim M(Y, A) \geq p$%) then
                    $dA = M(X, A) - M(Y, A)$;
                end
                if ($M(X, B) \sim M(Y, B) \geq p$%) then
                    $dB = M(X, B) - M(Y, B)$;
                end
                if ($dA$ * $dB < 0$ ) then
                    increment swap-counter for $p$ ;
                end
                increment counter ;
            end
        end
    end
    foreach diff $p$ from 0, 5, 10, 20, 30 do
        error-rate for size $z$ = (swap-counter for $p$)/(counter) ;
    end
end
```

Algorithm 1: Algorithm for finding error-rates of evaluation measures

The minimum topic set size we take is 5. The maximum size is roughly half of the number of topics available in the qrels (50 for INEX 2007 and 35 for INEX 2008). The number of iterations (n) is 50. We consider five different values for the tolerance p (p = 0, 5%, 10%, 20%, 30%). The measures M considered are the five official measures, viz. $iP[0.00]$, $iP[0.01]$, $iP[0.05]$, $iP[0.10]$, and AiP. For INEX 2007, there are $\binom{78}{2} = 78 \times 77/2 = 3,003$ pairwise comparisons (though there were 79 systems, two of the systems were identical), and for INEX 2008, there are $\binom{61}{2} = 61 \times 60/2 = 1,830$ pairwise comparisons for each topic-set size ranging from 5 to 50 and 5 to 35 respectively for five different measures at 5 different percentage points. The whole exercise leads to a set of error curves, based on the error rates actually computed by the above algorithm.

6.5.3 Extrapolating to Larger Topic-set Sizes

As explained above, error-rates can be experimentally calculated for topic-sets that contain at most half the total number of queries in the qrels. The error-rate vs. topic-set size graphs (see Figures 6.3, 6.4) are initially plotted from these empirically determined error-rates, and then extrapolated in order to estimate the error-rates for larger topic-sets. For each line, we fit a curve to the observed data using the FUDGIT package [77]. As observed by Buckley and Voorhees [141], the data seems to fit an 'exponential decay' family of functions, given by the following equation:

$$Y = A_1 \cdot \exp(-A_2 \cdot X) \tag{6.1}$$

where Y is the error-rate, and X is the size of the topic-set ($X \in \{5, 6, \ldots, 100\}$ for INEX 2007, and $X \in \{5, 6, \ldots, 70\}$ for INEX 2008). A_1 and A_2 are parameters to be estimated using the observed values of Y for $X \in \{5, 6, \ldots, 50\}$ (INEX 2007) or $X \in \{5, 6, \ldots, 35\}$ (INEX 2008). We re-write Equation 6.1 as

$$\ln Y = \ln A_1 - A_2 \cdot X \tag{6.2}$$

and fit a linear least squares regression model to the observed data corresponding to each line. To check 'goodness-of-fit', χ^2 values for the parameters are also calculated and observed to be in the range $0 - 9.13$ for both INEX 2007 and INEX 2008. The maximum allowable value for $\chi^2_{0.995,50}$ (degrees of freedom for both INEX 2007 and 2008 = number of iterations, 50) is 27.991. The model thus fits the data with at least 99.5% confidence.

6.5.4 Results

Figures 6.3 and 6.4 show the results of our experiments on error rates. The initial part of each line is plotted based on observed error rate values; the lines are then extrapolated as explained above. The error rate plots follow the same decaying pattern. While the error rates for $iP[0.00]$ are slightly higher than those for $iP[0.01]$, the $iP[0.05]$ and $iP[0.10]$ curves lie in between the corresponding $iP[0.01]$ and AiP curves. The graphs exhibit the following trends, as expected.

1. Error rates are maximum when the tolerance is 0%, and fall off as the tolerance increases.
2. Error rates are generally high with smaller query-sets, and progressively decrease as query-set size increases.
3. Error rates are higher for the early precision-metrics ($iP[0.00]$, $iP[0.01]$), and least for $MAiP$.

The INEX 2008 curves have higher error rates for the same topic set size because of the smaller number of total runs.

As discussed in Section 6.5.2, disjoint topic sets are used when computing error rates. For a particular metric, the obtained error rates are thus generally higher than when overlapping topic are used[1].

Table 6.2 shows the approximate minimum topic-set sizes for which error rates are 5% or less, for the various metrics. For both INEX 2007 and INEX 2008, an error-rate of less than 5% with 0% tolerance is only achievable with $MAiP$ as the metric, and indeed, for the topic-set size used at INEX 2007 and INEX 2008, the error-rate for $MAiP$ is consistently less than 5% for all tolerance values considered in our experiments. In comparison, the INEX results based on the official metric $iP[0.01]$ can be taken to contain fewer than 5% errors when $p = 5\%$ or higher , i.e. if we regard a performance difference of 5% or less as insignificant.

[1] When we considered sampling *with replacement* [91], the error rates were lower, even though the overall pattern was very similar to that reported here.

Table 6.2: **Minimum #topics required to achieve less than 5% error**

Data	% tolerance	$iP[0.00]$	$iP[0.01]$	$iP[0.05]$	$iP[0.10]$	$MAiP$
INEX 2007	0	$\gg 100$	$\gg 100$	$\gg 100$	> 100	**70**
	5	60	65	65	56	**45**
	10	35	35	35	35	**25**
	20	13	15	15	17	**13**
	30	7	7	8	10	**7**
INEX 2008	0	$\gg 70$	$\gg 70$	$\gg 70$	> 70	**58**
	5	48	46	43	40	**35**
	10	22	23	22	22	**18**
	20	8	10	10	10	**9**
	30	< 5	< 5	< 5	6	**5**

6.6 Discussion

Sampling experiments (both pool and query) were conducted with one common set of objectives: first, to study the relative stability of the INEX metrics, and second, to study the effect of reducing assessment effort on the overall evaluation results. Assessment effort can be reduced by reducing the number of queries judged, or by reducing the number of documents judged per query. The results in Chapter 5 (Pool Sampling) show the effect of reducing the number of documents judged per query, while keeping the number of queries unchanged; whereas the results here (Random Query Sampling) shows the effect of reducing the number of queries, keeping the number of documents per query in the qrels unchanged. In this section, we compare these results to find out the safest way to reduce assessment effort without compromising the reliability of the evaluation results. This question is of importance as the corpus used at INEX grew significantly in size since 2009.

Table 6.3: **Kendall Tau values at 60% sampling for INEX 08**

Sampling	$iP[0.00]$	$iP[0.01]$	$iP[0.05]$	$iP[0.10]$	$MAiP$
Random pool	0.717159	0.747541	0.824481	0.836284	0.912678
Random query	0.791913	0.823716	0.871038	0.899344	0.92612

6.6.1 Reducing Pool Size vs. Topic set Size

Since the number of documents judged per query is roughly constant (just over 600 articles), a given sample ratio (say $x\%$) corresponds to similar assessment effort for both pool sampling and query sampling. For example, at INEX 2008, a 60% sample of the pool contains 25,363 ($42,272 \times 0.6$) documents, while a pool corresponding to 60% of the query set contains roughly 25,200 articles ($70 \times 0.6 \times 600$).

Table 6.3 suggests that, for a given amount of assessment effort, the system rankings obtained with a smaller query set are closer to the original rankings than the rankings obtained when a subset of the documents are judged at random. This is also confirmed by a closer look at the results in Chapter 5 (Pool Sampling). This reveals that, in general, the curves for random query sampling are slightly more stable in comparison to their counterparts in random pool sampling (for example, see Figure 5.1 and Figure 6.1). One likely explanation for this is that, in the query sampling experiments, if a topic is used for evaluation, the complete relevance judgments for the topic are considered. Thus, unlike in *random pool sampling*, the query contributes to the precision scores of all systems uniformly; the reduction in τ is caused by the variation of system performance across topics.

However, much higher τ values are obtained when assessment effort is reduced by reducing pool depth, compared to when it is reduced by reducing the total number of queries (see Table 6.4).

In summary, if one wants to minimize the total amount of assessment effort, it is better to judge shallow pools for many queries, than to judge deep pools for fewer topics. This is in complete agreement with the observations from the document retrieval domain or the recent findings from

Table 6.4: **Kendall Tau values at 60% sampling**

Year	Sampling	$iP[0.00]$	$iP[0.01]$	$iP[0.05]$	$iP[0.10]$	$MAiP$
2007	random query	0.855752	0.831144	0.883361	0.904501	0.94191
	pool depth	0.995161	0.968979	0.964310	0.977652	0.991661
2008	random query	0.791913	0.823716	0.871038	0.899344	0.92612
	pool depth	0.992350	0.974863	0.980328	0.965027	0.979235

the TREC Million Query track [19]. If, however, assessors are likely to end up partially judging their assigned queries (at random), it may be better to reduce the workload by giving them larger pools for fewer topics and ensuring that, if they start judging a query, they complete the assessment for that query.

6.6.2 On error rates

Our experiments on *error rates* follow the methodology introduced by Buckley and Voorhees [141]. One can argue that this method lacks a solid mathematical foundation. Another disadvantage of this method is that it can measure error rates using only upto half the topic set size. In spite of these limitations, it does provide similar results as theoretically more sound methods such as the *Bootstrap sensitivity* method [110, 113]. The Bootstrap method has the advantage that it can sample upto the full topic set size unlike our method. This is why the Bootstrap has been more widely used in recent work [23, 105, 114, 148].

We did not explicitly measure the discriminative power of the metrics concerned, but interestingly, the way we found error rates can be easily improvised to provide some idea about discriminative power. The number of cases where system pairs are considered equivalent for two disjoint topic sets (score-difference $<= p\%$) can be obtained from Algorithm 1. This can be used to compute the proportionality of ties (PT). The lesser this ratio (PT), the higher is the discriminative power for a metric. Again, PT is seen to be generally inversely proportional to the error rates [110].

Of course, a more formal study on discriminative power of metrics could be done as suggested by Webber et al. [147] to investigate the number of topics (n) required to achieve a certain level of statistical power ($1 - \beta$) at a given level of significance (say, $\alpha = 0.05$).

6.7 Conclusion

This chapter outlines the second set of experiments that we conducted on the evaluation methodology adopted for the INEX adhoc focused task. In our pursuit of finding an "optimum" evaluation set up we attempted here to find answers to the following questions.

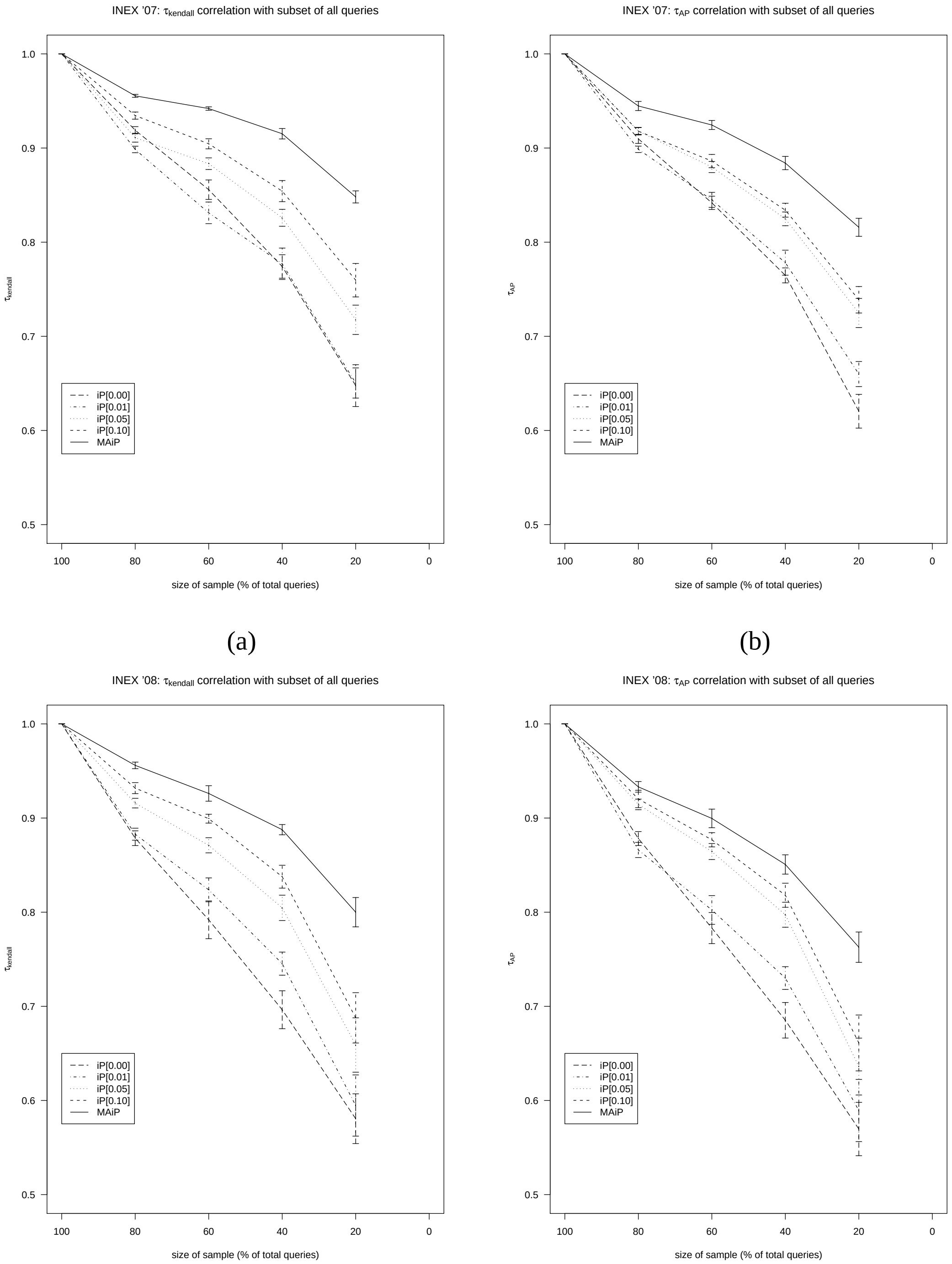
- *How reliable are the various metrics in ranking competing systems if the query set size is small? . . . What is the minimum number of queries that should be used to keep the error rates for the various metrics within a maximum allowable upper bound?*

  As seen in the last chapter (Chapter 5), we find that early precision measures ($iP[0.00]$, $iP[0.01]$) are more error-prone, and less stable if a small topic set is used, whereas AiP is the most stable. Our experiments also suggest that the pool size and number of queries used since INEX 2007 are large enough to reliably evaluate all submissions, i.e. the INEX results generally contain less than 5% error for all the metrics reported.

- *For a fixed amount of assessment effort, would this effort be better spent in thoroughly judging a few queries, or in judging many queries relatively superficially?*

  We observe that, to reduce the amount of effort required to create usable qrels, it is better to judge shallower pools for all topics, rather than reduce the number of topics that are judged. However, it is better to completely judge a smaller number of topics, than to randomly judge many topics.

We believe these findings (along with those in the next two chapters) should be useful while formulating the evaluation strategy to be used with much larger text collections.



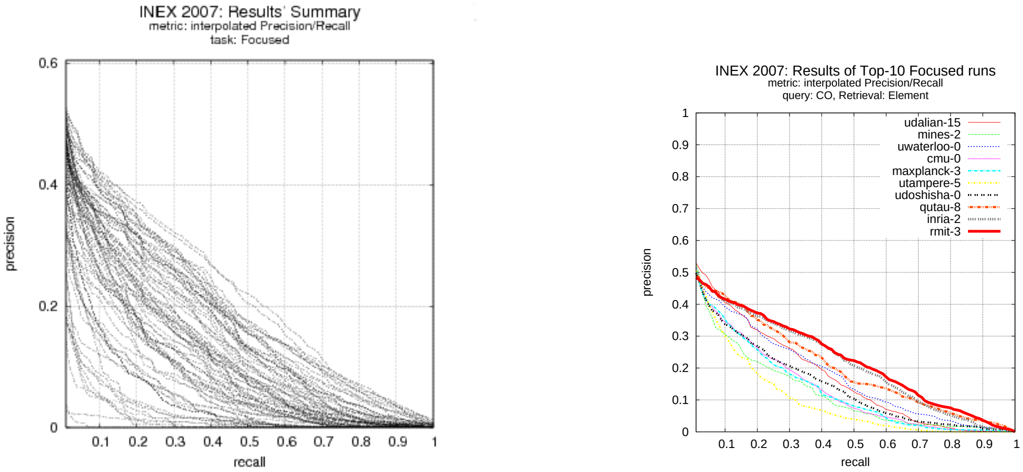
(a)

(b)

(c)

(d)

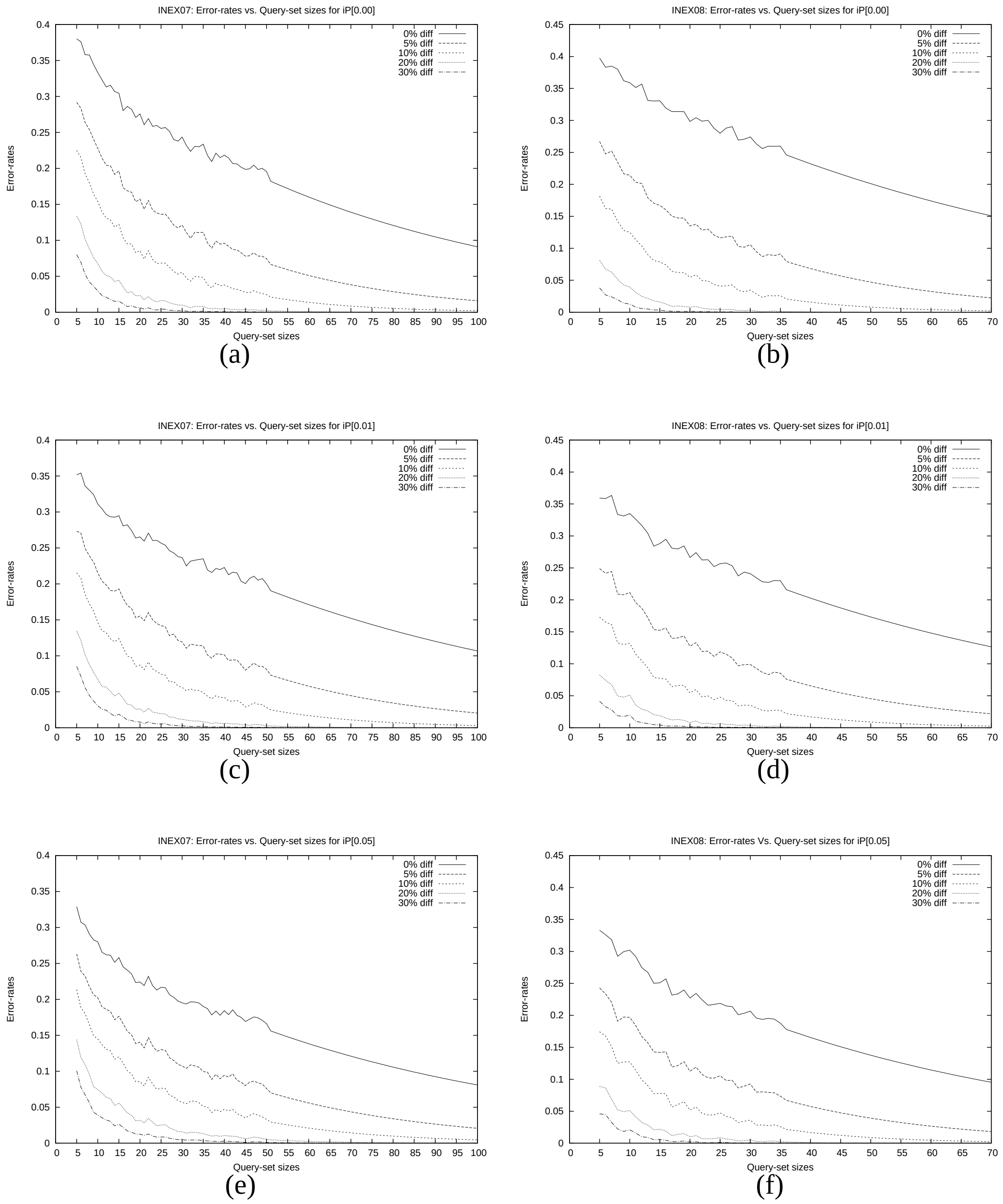
Figure 6.1: Rank correlation between original system rankings and rankings obtained with ran-



(a)

(b)

Figure 6.2: INEX 07 Precision-Recall curves for focused runs and top-10 systems.

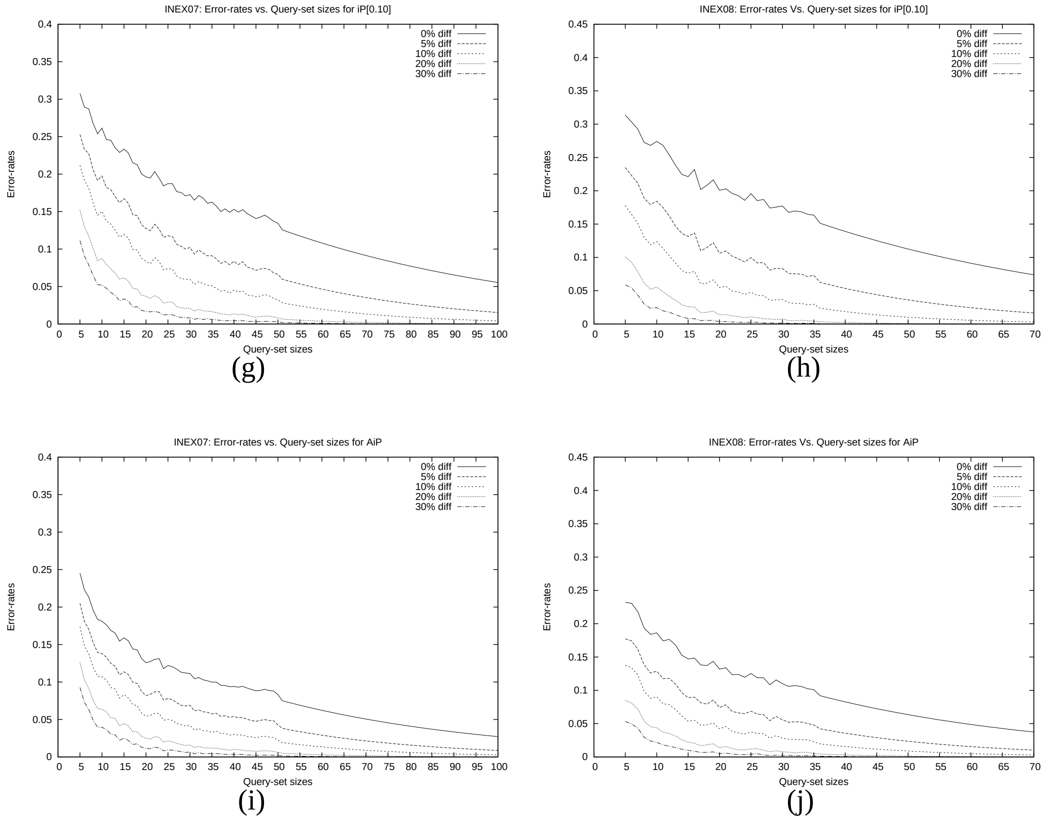


(a) (b)

(c) (d)

(e) (f)

Figure 6.3: Error-rates for $iP[0.00]$ $iP[0.01]$ãnd $iP[0.05]$ as query-set size changes (INEX 2007 and 2008).



(g)

(h)

(i)

(j)

Figure 6.4: Error-rates for $iP[0.10]$ and AiP as query-set size changes (INEX 2007 and 2008).

Chapter 7

Leave One Group Out

This chapter deals with the third set of experiments related to INEX adhoc task evaluation. We explore the reusability of the INEX adhoc pool. Using similar experimental data as in Chapter 5, and Chapter 6, we specifically attempt to answer a broad question: can the INEX collections be used to fairly evaluate 'new' systems that did not participate in the pooling process? What advantage does a system get during evaluation by contributing to the INEX pool? We simulate the evaluation of a 'new' system by removing from the pool all contributions from the run(s) of an organization, and evaluating those run(s) with assessments based on the remaining pool. The differences in raw score and ranking among the systems are observed.

We find that the bias towards a system due to its contribution to the pool is mostly *insignificant*. This suggests that a 'new' system which did not contribute to the INEX 2007 or INEX 2008 pool can be fairly evaluated on the basis of the corresponding qrels.

7.1 Introduction

Like other evaluation forums, INEX also uses pooling-based evaluation, where systems' responses to a pre-defined set of topics are used to generate ground-truth for the evaluation of the same set of systems. Unlike other evaluation forums, however, the systems at INEX retrieve a set of document components, i.e., elements or passages from XML documents. As the retrieval unit can be an element of arbitrary granularity and length, the evaluation exercise is markedly different at INEX (see Sec. 5.1). Though retrieval is done at the sub-document level, both pooling and assessment are done with complete documents, i.e., if a system retrieves a fragment of a document, however small, the whole document is included in the pool and the entire document is judged to identify all relevant elements or passages in it. While this feature is expected to add diversity and robustness to the pool, this needs to be confirmed through a systematic investigation. In the earlier two chapters, we simulated incomplete relevance assessments in two different ways, but the total number of systems contributing to the pool and the evaluation remained constant. Thus, the amount of bias introduced by the contributing systems in the INEX evaluation framework was not studied. Is the bias *significant* overall or at the per query level? Can the pool be reused? In other words, can a 'new' system which did not contribute in pooling be fairly evaluated using the same test collection? We investigate these issues in this chapter using the INEX 2007 and 2008 adhoc test collections.

In the next section, we review relevant past work that provides the background for the chapter (for other related work, see Sec. 5.2). The motivation of this study is discussed in Section 7.3. Next, Section 7.4 describes the experimental procedure. Section 7.5 presents results, observations and analyses. Section 7.6 outlines a the limitations and the scope for future work in this line. We conclude the chapter in Section 7.7.

7.2 Previous Work

Reusability is one of the important long-term goals while creating standard bench-mark test collections. Although collections are often used for evaluating participating systems in the short-term, it is desired that these collections can be used to measure the effectivity of a new technique in future. Quantifying reusability has long been an area of research in IR. Zobel [158] conducted a series of experiments to study the fairness and trustworthiness of the pooling-based evaluation methodology in the context of TREC activities. His work included experiments on pooling to study how system performances get reinforced due to pooling, and how the omission of a system's contribution from the pool affects performance measures (popularly known as 'leave-one-out' experiments).

Büttcher et al [14] revisited the 'leave-one-out' experiments and applied two statistical learning techniques (KLD and SVM) to infer whether an unjudged document coming from a "new" system which did not take part in pooling is relevant or not.

Sakai and Kando [113] also conducted the leave-one-group-out experiments with different metrics like AP, $nDCG$ and Q, and their condensed-list variants AP', $nDCG'$ and Q'. The *condensed-list* variant of a metric is calculated using a ranked list obtained by removing unjudged documents from the original ranked list. AP', $nDCG'$ and Q' seemed to overestimate a 'new' system's performance while AP, $nDCG$ and Q underestimate the 'new systems' [112]. However, the extent of overestimation is higher than the extent of underestimation.

All these experiments were done in a document retrieval setting on TREC and/or NTCIR data. To our knowledge, no prior work has been reported on the reusability study of the INEX pool.

Our leave-one-out experiments are motivated by the same objective as in Zobel [158]. The methodology has been slightly modified, however, in accordance with the findings of Büttcher et al [14]. Both these studies [158, 14] looked at the overall impact of the pooling bias, while in reality, the pooling bias varies from query to query. Besides the overall effect, we also look at the effect of pooling bias on a per-query basis.

7.3 Motivation

One of the assumptions underlying pooling-based evaluation is completeness: the contributing systems are together successful in finding all relevant items. Thus, an unjudged document is assumed to be non-relevant. If we use a set of assessments to evaluate a technique that did not contribute to the underlying pool, and the technique actually succeeds in finding unjudged, relevant documents that were not found by any of the systems contributing to the pool, the new technique does not get credit for this. Its effectiveness can thus be underestimated. In the current set of experiments, we try to quantify the extent of this problem within the INEX setting.

7.4 Experiments

These experiments are inspired by the work of Zobel [158] on TREC data. Given a set of N submissions that contribute to a given pool, each run's contribution to the pool is removed in turn, i.e., the pool is constructed on the basis of the remaining $N - 1$ runs. The qrels corresponding to this reduced pool are used to evaluate the system. The difference between the original and new scores can give us an idea about the robustness of the pool.

However, there are two practical issues here. One, runs from a particular group are quite often similar in strategy. Second, the INEX adhoc pool is made up of runs from three separate subtasks. It is also a fact that participants tend to submit variants of runs to each subtask. Thus, leaving one focused run out of the pool, while including other related submissions from the same participant might not change the pool much, and the stability of the metric may be overrated when the difference between the original and new scores is computed.

We therefore follow the suggestion of Büttcher et al. [14], and 'leave-one-group-out' at a time in our experiments. We start with the 100% pool reconstructed from the INEX focused submissions (see Section 5.5 in Chapter 5) as our baseline. For each participating group, a pool is re-created by excluding all the focused runs from that group. Each excluded run is then evaluated using the assessments generated from this pool. We compare the new scores with those computed from the baseline qrels for each query and for each run. We also check whether the overall change in score for a particular run is significant, using a paired t-test.

7.5 Results

Table 7.1 and Table 7.2 summarize the distribution of relative changes in scores. Table 7.1 provides the distribution of relative changes in overall scores for the participating systems. A negative (positive) change signifies that the new score is higher (lower) than the baseline score. As explained above, the effectiveness of a system that does not contribute to the assessed pool may, in general, be underestimated. Thus, for most systems, the score obtained with the 'leave-one-group-out' pool is somewhat lower than that obtained with the 100% pool (see the row corresponding to the $(0, 10]$-%

Table 7.1: **Distribution of relative change in scores for 'leave-one-group-out' (overall)**

Data	% change	$iP[0.00]$	$iP[0.01]$	$iP[0.05]$	$iP[0.10]$	$MAiP$
INEX 2007	$(-\infty, -50]$	0	0	0	0	0
	(-50, -10]	0	0	0	0	0
(out of	(-10, 0)	0	2 (2.6%)	2 (2.6%)	3 (3.8%)	4 (5.1%)
78	0	13 (16.7%)	9 (11.5%)	7 (9%)	8 (10.3%)	8 (10.3%)
systems)	(0, 10]	63 (80.7%)	67 (85.9%)	69 (88.4%)	67 (85.9%)	66(84.6%)
	(10, 20]	1 (1.3%)	0	0	0	0
	(20, 50]	1 (1.3%)	0	0	0	0
	(50, 100]	0	0	0	0	0
INEX 2008	$(-\infty, -50]$	0	0	0	0	0
	(-50, -10]	0	0	0	0	0
(out of	(-10, 0)	0	2 (3.3%)	6 (9.8%)	8 (13.1%)	7 (11.5%)
61	0	18 (29.5%)	7 (11.5%)	4 (6.6%)	4 (6.6%)	0
systems)	(0, 10]	43 (70.5%)	50 (81.9%)	51 (83.6%)	49 (80.3%)	54 (88.5%)
	(10, 20]	0	2 (3.3%)	0	0	0
	(20, 50]	0	0	0	0	0
	(50, 100]	0	0	0	0	0

change in score).

On a closer examination of these results, we found that the effect of a system's contribution to the pool is query-specific, and is not uniform across queries. The changes in overall system scores shown in Table 7.1 hide these interesting details. In Table 7.2, therefore, we report the distribution of relative changes in scores across all possible query-run pairs.

As above, for a number of queries, the score of a system drops when the 'leave-one-group-out' pool is used. In the vast majority of cases, however, the score remains unaffected, no matter what pool is used.

On the whole, the results are reassuring, as they suggest that new systems may be reliably evaluated using the INEX qrels. The actual number of queries for which there are no changes decreases from $iP[0.00]$ to $MAiP$. This suggests that the top-ranked items retrieved by any system are also retrieved by at least one other system, although possibly at different ranks. As one goes down the ranked list, the number of unique contributions to the pool increases. Thus, precision scores at

Table 7.2: **Distribution of relative change in scores for 'leave-one-group-out' (per-query-level)**

Data	% change	$iP[0.00]$	$iP[0.01]$	$iP[0.05]$	$iP[0.10]$	$MAiP$
INEX 2007	$(-\infty, -50]$	0	2	4	6 (0.1%)	1
	(-50, -10]	0	5	7 (0.1%)	13 (0.2%)	24 (0.3%)
(out of	(-10, 0)	0	6 (0.1%)	17 (0.2%)	27 (0.3%)	368 (4.4%)
8,307	0	7,890 (95%)	7,736 (93.1%)	7,612 (91.6%)	7,580 (91.2%)	6,869 (82.7%)
cases)	(0, 10]	191 (2.3%)	326 (3.9%)	451 (5.4%)	494 (6.0%)	727 (8.8%)
	(10, 20]	61 (0.7%)	74 (0.9%)	79 (1.0%)	92 (1.1%)	135 (1.6%)
	(20, 50]	77 (0.9%)	94 (1.1%)	91 (1.1%)	60 (0.7%)	117 (1.4%)
	(50, 100]	88 (1.1%)	64 (0.8%)	46 (0.5%)	35 (0.4%)	66 (0.8%)
INEX 2008	$(-\infty, -50]$	0	1	7 (0.2%)	7 (0.2%)	2
	(-50,-10]	0	2	7 (0.2%)	5 (0.1%)	30 (0.7%)
(out of	(-10, 0)	0	1	9 (0.2%)	20 (0.5%)	284 (6.7%)
4,270	0	4,135 (96.8%)	4,049 (94.8%)	3,955 (92.6%)	3,911 (91.6%)	3,395 (79.5%)
cases)	(0, 10]	84 (1.9%)	146 (3.4%)	217 (5%)	253 (5.9%)	465 (10.9%)
	(10, 20]	15 (0.4%)	26 (0.6%)	26 (0.6%)	38 (0.9%)	47 (1.1%)
	(20, 50]	17 (0.4%)	33 (0.8%)	34 (0.8%)	17 (0.4%)	27 (0.6%)
	(50, 100]	19 (0.5%)	12 (0.3%)	15 (0.4%)	19 (0.4%)	20 (0.5%)

Table 7.3: **Number of runs with significant difference in score (level of significance 0.05)**

Data	Significant diff.	$iP[0.00]$	$iP[0.01]$	$iP[0.05]$	$iP[0.10]$	$MAiP$
INEX 2007	YES	8 (10.2%)	14 (17.9%)	24 (30.8%)	15 (19.2%)	27 (34.6%)
	NO	70 (89.8%)	64 (82.1%)	54 (69.2%)	63 (80.8%)	51 (65.4%)
INEX 2008	YES	0	2 (3.3%)	5 (8.2%)	10 (16.4%)	15 (24.6%)
	NO	61 (100%)	59 (96.7%)	56 (91.8)	51 (83.6%)	46 (75.4%)

higher recall levels are affected for more queries.

More interesting is the fact that the performance of some systems actually improves when their contributions to the pool are omitted. This initially appears to be counter-intuitive. However, the following (somewhat extreme) example shows how this may happen. Consider a query for which there are two relevant documents. The first document is retrieved at rank one by all systems, while the second is retrieved by only one system at rank $R \gg 1$. The average precision for this system is $\frac{1}{2}(1 + 2/R) = 1/2 + 1/R$. When the system's contribution to the pool is omitted, there is only one relevant document for the given query, and the system gets a perfect score of 1.

On a closer examination of the results for individual queries, we find that such an improvement can also occur in another situation that is peculiar to the focused retrieval task (as opposed to the document retrieval task). Consider a system that retrieves only non-relevant passages/elements from a document that contains relevant material. If this system is the only one to contribute this document to the pool, its performance will improve when this contribution is omitted from the pool.

We also used a two-sided, paired t-test to check whether the overall change in score for each run is significant. Table 7.3 shows the number of runs for which the overall score is significantly affected, when the corresponding group's contributions to the pool are left out.

These results show that the early precision metrics are significantly affected for only a small number of systems. This can be accounted for in two ways. As stated above, most of the documents returned at top ranks are also retrieved by other systems. By and large, this consensus decreases as one goes down the ranked list. Since the $MAiP$ score is calculated using the entire ranked list, this metric is most significantly affected when a system's contribution is omitted from the pool.

Secondly, since the early precision metrics are relatively unstable, even large differences between two systems may not be regarded as significant on the basis of a t-test.

7.6 Limitations and Future Work

Our experiments in the Sections "Reducing Pool-depth" (Sec. 5.5 and here "Leave One Out" are based on a generated pool which is created from only the ad hoc focused submissions. This pool is not identical to the actual pool used at INEX, but it is a close clone. Though our results would have been slightly different if the original INEX pool could have been regenerated, we believe that this difference would be small.

Secondly, when reducing pool-depth, we found that rankings do not change significantly even when the pool size is substantially reduced (see Table 5.4 and Table 5.5). However, if such small pools were actually used in practice, it would be interesting to study whether new systems could still be reliably evaluated on the basis of the resultant relevance assessments. In other words, it might be instructive to redo the Leave One Out experiments using these reduced pools as a starting point.

In the last section, early precision metrics were observed to be less affected in the 'leave-one-out' experiments than precision at higher recall values. Two possible reasons for this are given. One, more consensus at the early ranks than at the lower ranks when one goes down the retrieved list; and two, relative instability of early precision metrics. It would be interesting to separate the contribution of these two factors (to the results in Table 7.3).

7.7 Conclusion

Evaluation is a gruelling challenge for XML retrieval research. Ever since the inception of INEX, its evaluation measures have changed at regular intervals. With the inclusion of arbitrary passages as valid retrieval units besides the usual XML elements, the need for a common set of effectiveness measures has gained importance. INEX 2007 therefore introduced a set of *precision-recall* based measures for its ad hoc tasks. The main aim of our evaluation related work was to investigate the reliability and robustness of these focused retrieval measures, as well as the pooling method used at INEX. We specifically look at the following issue in this chapter:

- *When a set of relevance assessments is used to evaluate a 'new' system that did not contribute to the pool used in the relevance assessment process, are the results biased against this system?*

  The findings reported in this chapter suggest that a 'new' system which did not contribute to the INEX 2007 or INEX 2008 pool can be fairly evaluated on the basis of the corresponding qrels. In most cases, contributing systems get insignificant advantages due to their contribution to the qrels. The fact that indexing and retrieval done at the sub-document level, but pooling and assessment are done at the document level (in the sense that the whole document is pooled and judged, although a single small passage/element is retrieved from it) bolsters the resiliency and reusability of the pool.

Chapter 8

Cost-effective Reusable Pooling

This chapter looks at the issue of reducing assessment effort from a different perspective: varying assessment effort across queries. Based on our observations on INEX evaluation data, we propose a simple and pragmatic approach for the creation of smaller pools for Cranfield-based evaluation of adhoc retrieval systems. Instead of using an apriori-fixed depth, for all queries, variable pool-depth based pooling is adopted. The pool for each topic is incrementally built and judged interactively. When no new relevant document is found for a reasonably long run of pool-depths, pooling is stopped for the topic. Depending on the available effort and required performance level, our proposed approach can be adjusted for optimality. Though spurred by our study of the evaluation of focused XML retrieval, the proposed solution generalizes to a document retrieval setting as well. Experiments with TREC-7, TREC-8 and NTCIR-5 data show its efficacy in substantially reducing costs without seriously compromising the reliability of evaluation.

8.1 Introduction

Like all other standard forums of Information Retrieval (IR) evaluation (TREC, CLEF, NTCIR, FIRE etc.) INEX has also adopted the Cranfield paradigm, where a test collection is built with three major components: 1) a set of documents (*corpus*), 2) a set of information needs (*topics*) and 3) a set of relevance judgments for each topic (*qrels*). Ideally, these qrels should be complete, i.e. all information items in the corpus (document for standard IR; passages/elements for focused retrieval) should be judged as relevant or non-relevant with respect to each topic in the topic set. For a large corpus it is infeasible to construct such a test collection because the amount of time and effort involved therein would be prohibitive. In practice, a more efficient approach called *pooling* is used. Pooling works as follows. Let Q be a particular topic from the topic set of a test collection. Let S_1, S_2, . . ., S_m be m ranked lists of documents retrieved in response to Q, by various retrieval systems / models / algorithms. Let D_{ij} represent the document at rank j in S_i. Then the pool P_Q for topic Q at depth k is given by

$$P_Q = \{D_{ij} | \quad 1 \leq i \leq m, \quad 1 \leq j \leq k\}.$$

The number k may be a constant (e.g. 100) whose value is fixed a priori. This is the practice followed at TREC, CLEF, NTCIR and FIRE. This k may also be determined in a query-dependent manner. For example, at INEX, k is chosen for each query in such a way that $|P_Q|$, the pool size, remains approximately the same across all queries in the query set. Each document in the pool is judged for relevance by at least one assessor. The information items in the pool that are judged relevant are assumed to be the only relevant items for the query; all unjudged documents (i.e. those outside the pool) are assumed to be non-relevant. Though this assumption is likely to be erroneous, the pooling method has proven reasonably robust over the years across different IR tasks at different evaluation forums. Despite some differences between focused retrieval and traditional document retrieval, pooling has been adopted for focused retrieval evaluation at INEX with suitable modifications (see Sec. 5.1 for details).

Even though pooling substantially saves assessment effort compared to exhaustive judgments, with the growing size of collections, creating relevance assessments has become immensely resource-intensive in terms of manpower and time even with pooling.

We demonstrate here a simple and pragmatic approach for the creation of smaller pools for the evaluation of adhoc retrieval systems. Instead of using a constant pool-depth for all topics, the pool for each topic is incrementally built by gradually increasing the pool-depth in steps, and judged interactively. When no new relevant document is found for a reasonably long run of pool-depths, pooling is stopped for the topic. The proposed technique serves two major purposes:

- it can reduce the total assessment effort without compromising the unbiasedness of the pool and
- it offers the possibility of having a better estimate of *recall* per query compared to the existing pooling technique.

In a nutshell, based on available effort and required performance level, the approach can be adjusted for optimality.

Unlike other low-cost evaluation strategies, where one cannot figure out a substantial number of relevant documents which *are* particularly instrumental in post-hoc fault analysis to improve system performance, our method is very much within Cranfield paradigm.

We test our algorithm first on the INEX focused task. Later, we extend and generalize it to a document retrieval setting using TREC-7, TREC-8 and NTCIR-5 data. The approach scales well in both the settings.

In the next section, we review relevant past work. The motivation behind our study is discussed in Section 8.3. Next, Section 8.4 details the experimental procedure. Section 8.5 discusses results. Section 8.8 presents the limitations of our approach and outlines the scope for future work in this line. We conclude the chapter in Section 8.9.

8.2 Previous Work

With the growing size of test collections, it has become imperative to reduce the amount of human effort required for creating pooling-based relevance assessments. There have been several studies in this direction since the late nineties. Zobel [158] first studied the efficacy of large scale evaluation experiments and pointed out that the TREC setup is able to identify at most 50–70% of all relevant items. He also indicated the possibility of adaptively fixing the pool-depth based on the concentration of relevant documents in the system rankings on a per query basis and maximizing recall. However, his work did not report any experiments related to reducing assessment effort.

Cormack et al. [25] first attempted to create a low-cost pool by their *move-to-front* (MTF) technique. Since all systems are not equally good in retrieving relevant items, systems that have retrieved relevant items recently were given a preference during pool-creation. Their technique thus introduced a bias in favor of systems which retrieve relevant documents at early ranks. The technique, although successful in identifying more relevant items using less effort compared to standard pooling, suffered from doubly favoring high-precision systems.

Soboroff et al. [131] attempted to bypass pooling and assessment by creating pseudo-relevance judgments through random sampling from the submitted runs. Though their technique was able to *roughly* rank the participating systems, the correlation of this ranking with a ranking based on actual human assessments was far from satisfactory.

Buckley and Voorhees [13] proposed a new metric called *bpref*, which is more resilient than Average Precision (AP) to incompleteness. For a dynamically changing collection like the Web or an incomplete pool, the authors showed that the measure could be effectively used to evaluate systems. However, Sakai [109] showed that other metrics (AP or Q-measure or nDCG) can also give comparable to better results if we consider *condensed lists* (obtained by removing unjudged documents from any ranked list).

Carterette et al. [18] proposed a mathematically elegant minimal test collection for evaluating a small set of runs. The test collection thus produced was more applicable for dynamic and transient corpora but had limited reusability. In follow-up work [17], the reusability issue was addressed with a set of judgments much smaller than usual pool. However both the approaches were measure-specific and based on a pairwise comparison of systems involving mathematically complex techniques.

Aslam and his team ([2], [155]) proposed some new measures based on statistical sampling which accurately estimated standard measures such as AP and R-Precision with low error using only a small number of judgments.

Moffat et al. [83] described a method that selectively judges a few documents based on their popularity. Each document was assigned a weight depending on its potential to be relevant. The potential of a document was determined by its highest rank and/or popularity among different ranked lists.

The pooling technique also used a projected rank-biased-precision (RBP) score for a document to decide whether to consider for judgment or not. In other words, the technique was biased to a particular metric.

Guiver et al. [45] proposed to choose a few good topics for evaluation which, according to the authors, are sufficient for reliably evaluating a set of systems.

Most of these recent low-cost evaluation techniques are either *biased* to some set of systems or specific to particular measure(s). In order to focus on the quick evaluation of a set of systems, the techniques have also, in general, seriously compromised on *completeness*. The strategies, therefore, lack an important aspect of pooling: they do not provide adequate data for system diagnosis. These new techniques may correctly estimate the performance of a system using a particular metric by observing only a few retrieved items. However, such pools do not provide enough information on the relevant documents which a particular system missed retrieving. To troubleshoot why a system is not able to retrieve relevant items in early ranks, one needs the knowledge of relevant items for the topic concerned. Traditional Cranfield-based pool provides such ground-truths which are useful during a post-hoc analysis, fault-diagnosis and/or tuning of the system. Also, this pool enables judging a ‘new’ system which did not participate in pooling. Moreover, this pool is independent of any metric and performance of a system can be measured using any one of the several metrics. In order to maintain these *reusability* aspects - which are intrinsic to the Cranfield-based technique, a new pooling technique should be equally fair or unbiased to all systems and to all metrics. Also, the pool should try to achieve completeness with maximum possible relevant items in it. These attributes are, however, either seriously compromised or completely neglected in newer low-cost evaluation techniques.

8.3 Motivation

In order to investigate how assessment effort can best be utilized, we finally look at the question of varying the amount of assessment effort across queries. We observe that the relation between pool-depth k and the number of relevant documents found for a query Q varies widely across queries.

In general, for any query, the rate of finding new relevant documents decreases as pool-depth is increased and eventually it drops to near zero after a threshold, i.e. one does not find a significant number of new relevant documents even though pool-depth is increased substantially beyond this point. For a query Q, we call this threshold (which is unique for each Q) the *critical pool-depth* (k_{cr}). The rate of finding new relevant documents and the critical pool-depth vary significantly from query to query. To ensure a reasonably good estimate of recall for a given Q, k should be no less than its critical pool-depth.

If the rate of finding relevant documents is high for a query (e.g. queries 570, 582 from INEX 2008 adhoc collection in Figure 8.1), and pooling for that query is stopped because the target pool size or pool depth has been reached, one may not reach critical pool-depth. For example, for query 582, pooling stops at about $k = 26$ as the pool size reaches 608 documents (the pre-determined number of documents to be judged is about 600 at INEX). Figure 8.1 suggests, however, that we cannot be sure to have reached critical pool-depth. Similarly, for query 570, pooling is stopped at a depth which is probably less than the critical pool depth (k_{cr}). In contrast, there are queries for which the rate of finding new relevant document is remarkably low (query 587, 634). For these queries, critical pool-depth appears to have been reached much before the pool-size reaches its target value. For query 634, critical pool-depth seems to be achieved when the total number of documents judged is near 180, while for query 587, this number is 480.

The above observation is not specific to INEX assessment data but quite commonplace across evaluation forums such as TREC and NTCIR as shown in Table 8.1. For example, topic 363 had 16 relevant documents ($nrels$) out of 1597 documents in the TREC-7 pool which was created taking pool-depth $k = 100$. However, when we incrementally re-built the pool in our experiment, we observed that the same number of relevant documents ($nrels$) is achievable at pool depth 20 (given by k_{cr}) when the pool-size is only 348. In other words, we could save the effort of judging $(1597 - 348) = 1249$ documents, if we stop pooling at k_{cr}. Similarly for query 31 of NTCIR-5 cross-lingual data, $nrels = 32$ could be achieved by creating a pool with depth 25 only. This can save effort for relevance judgment of $(1723 - 538) = 1185$ documents as this judgment does not produce any new relevant document. It is also found that the number of such queries with low critical pool-depth is quite substantial.

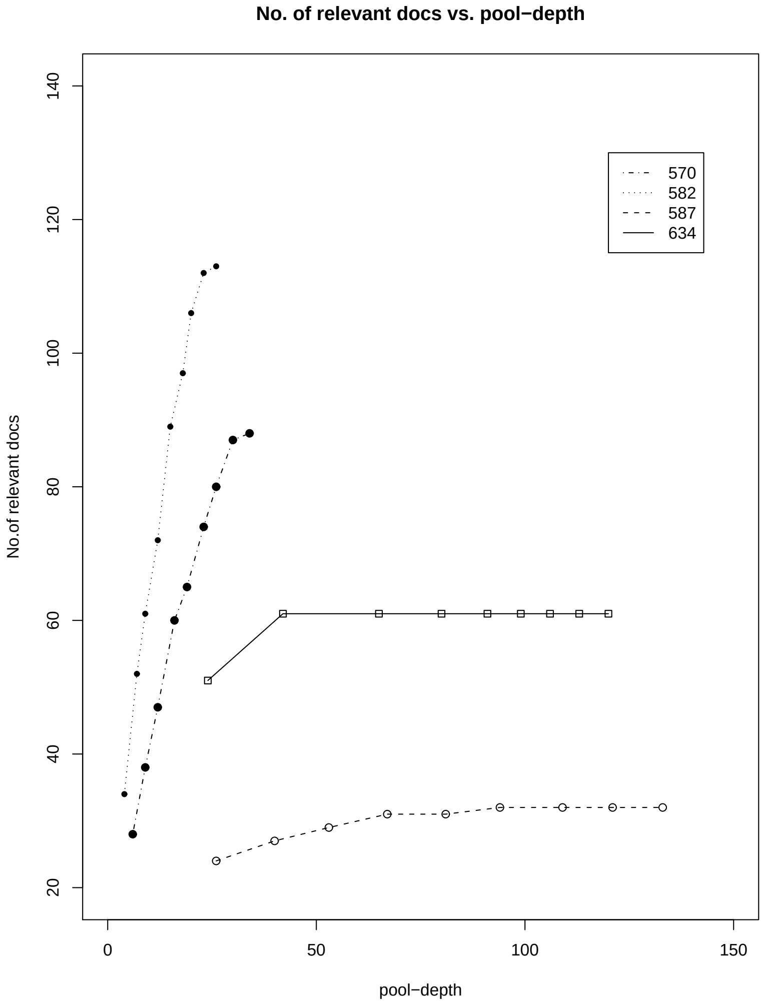


Figure 8.1: No. of Rel docs vs. Pool-depth.

Table 8.1: **Pool saturation at k_{cr}**

Test Collection	**topic-id**	k_{cr}	$nrels$	**pool-size at**	
				k_{cr}	$k = 100$
TREC-7 adhoc	363	20	16	348	1597
	384	76	51	926	1225
TREC-8 adhoc	403	14	21	148	1382
	410	47	65	943	2183
NTCIR-5 cross-lingual	31	25	32	538	1723
	4	20	10	451	1788

It would be nice if we could balance the assessment effort across queries, by judging fewer documents for queries like 634 (Fig. 8.1), and using the manpower thus saved to assess more documents for queries like 582. For a given amount of assessment effort, we would then be likelier to identify more relevant documents, thus obtaining a better estimate of recall.

This is of particular importance as larger corpora are being used nowadays for evaluation. Our proposed technique remaining within Cranfield paradigm also offers a trade-off between the overall assessment effort and fraction of total relevant documents found.

8.4 Experiments

Our approach is inspired by that of Zobel [158]. Zobel started with judgments for all topics for pools of some to some initial depth, and then used extrapolation to find the likely number of relevant documents ($reldocs$) for each query. He suggested that either the most promising topics should be further judged, or the least promising topics should be removed from the pool. Although Zobel indicated the possibility of maximizing recall and reducing effort, his work did not experimentally corroborate the claim.

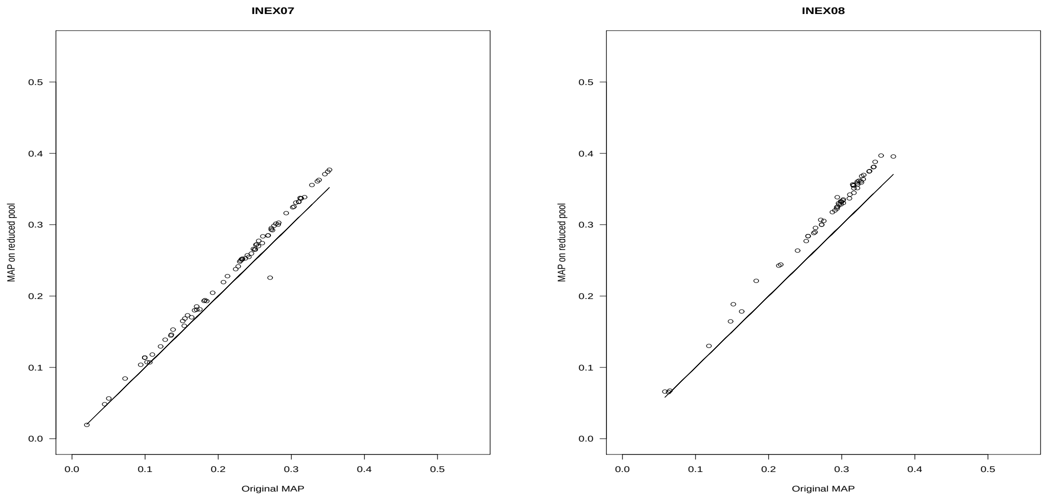
Zobel's work was based on TREC data, where a fixed pool-depth is used (typically the top 100 documents from each contributor are pooled) and pools vary in size across queries. In contrast, INEX uses a fixed pool-size, i.e. the pool-depth is dynamically chosen for each query so that a pool size of around 600 documents is reached. The dynamically chosen pool depth, which is at least 30 in terms of articles (and substantially higher in terms of elements), is assumed to provide enough coverage of the relevant articles for the majority of topics. Although INEX assessment is done at the sub-document (passage/element) level, the pool is created at the document level. Even if only a small single passage/element of an XML document is retrieved by any of the participating systems, the entire document is included in the pool. The whole document is judged for relevance and all the passages marked relevant by the assessor are included in the qrels. Thus, the judgments typically include a number of passages/elements which may not be retrieved by any of the participating systems (see Sec. 5.1).

We consider each of the topics used in INEX 2007 and INEX 2008 and build the pool incrementally. We start with pool depth $k = 1$, and increase k up to a suitable pool depth so that the pool size reaches just above 600 documents (see 5.5 for details); both the number of relevant documents ($nrels$) and pool size are noted at each k. As k increases, the count of new reldocs found at each k generally decreases. However, the rate of decrement is not uniform. It contains a few irregular bursts in-between. To estimate k_{cr}, thus, one needs to smooth the data. 2-stage smoothing is adopted. First, the raw nrels values are smoothed using moving averages over a window of size w (i.e. in the smoothed version, the raw nrels at any pool depth i is replaced by the arithmetic mean of w consecutive nrels, starting at $k = i$ and going up to $k = i + w - 1$). Then the rate of finding new reldocs (count for new reldocs at depth i, given by (nrels at $k = i+1$) - (nrels at $k = i$)) is also smoothed using moving averages with a window of size W. When this smoothed rate of finding new reldocs remains below a certain threshold t continuously for at least a pre-set number of pool depths (l), the corresponding depth is estimated as k_{cr} for the topic. A reduced pool is obtained based on k_{cr} for each topic in the topicset. The projection of the original qrels on the reduced pool provides a reduced qrels. Thus, each reduced qrels corresponds to a parameter setting for w, W, t, l and we consider $w = 6, 8, 10, 12, 14$; $W = 2, 3, 4, 5, 6$; $t = 0.05, 0.10, 0.20, 0.40, 0.80$; $l = 3, 4, 5, 6$; i.e. we consider $(5 \times 5 \times 5 \times 4) = 500$ qrels for a collection. We evaluate runs using the reduced qrels and compare the mean average precision (MAP) scores and ranking with those obtained using the original qrels.

2-stage smoothing is done as we found that the increments in nrels are not uniform as k increases. Even when the nrels values are smoothed, the rate of finding new reldocs is found to contain a few bursts which need a second level of smoothing.

8.5 Results

For INEX adhoc tasks, even if a small element/passage from a document occurs in the pool, the whole document is assessed to find all relevant passages from the document. In other words, INEX adhoc assessments are done at the document level. Therefore, it makes sense to check how many relevant documents occur in the judged pool. If we can ensure that our proposed reduced pooling yields almost the same number of relevant documents as the original pool, evaluation at the sub-document level will also provide similar results. We therefore evaluate our reduced pooling method at the document level and use MAP as the primary evaluation measure.

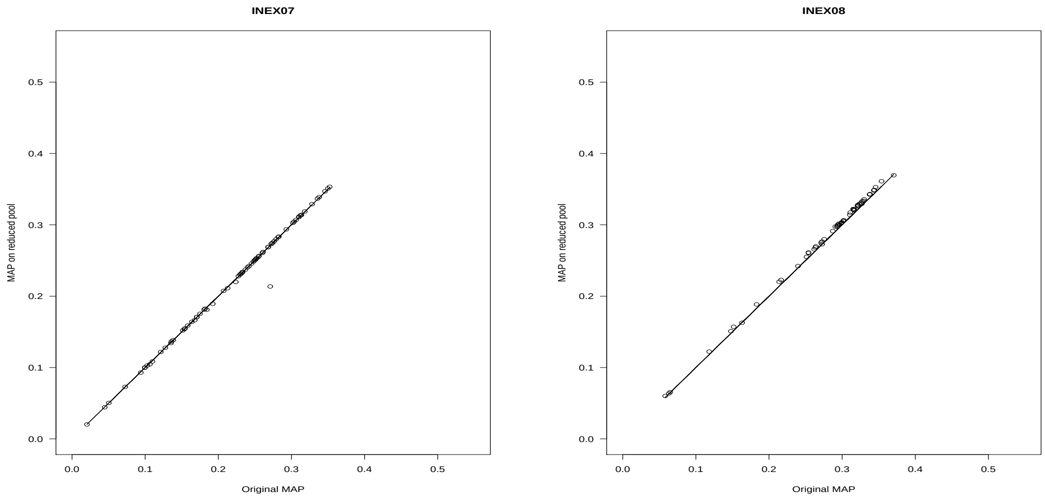


(a) INEX 2007

(b) INEX 2008

Figure 8.2: MAP values for reduced vs. original INEX pool (*aggressive stopping*: $w = 6$, $W = 2$, $t = 0.80$, $l = 3$)

The MAP values obtained for all the systems for a given set of values for (w, W, t, l) are plotted against original MAP values for these systems. As evident from Figure 8.2, MAP values obtained



(a) INEX 2007

(b) INEX 2008

Figure 8.3: MAP values for reduced vs. original INEX pool (*relaxed stopping*: $w = 14$, $W = 6$, $t = 0.05$, $l = 6$)

using the original and reduced pools are in close agreement. A reduced pool causes a slight underestimation of the *recall* base (total number of relevant documents for each topic) which in turn leads to overestimation of MAP-scores. This is particularly true for aggressive estimate of k_{cr}, typically obtained using small values of w, W, and l and a high value t.

The MAP values are much closer to the original when we relax the stopping criterion and allow a bigger pool (Figure 8.3). However, these bigger pools are still much smaller than the original ones.

Evaluation using shallow pools typically favors precision-oriented systems over recall-oriented ones, since a number of relevant documents are absent in such pools. Therefore MAP-based system ranking (where overall retrieval performance is looked at), may differ significantly when using a shallow pool, compared to using the full pool. It is generally seen that the shallower the pool, the larger is the difference. In our approach, the disagreement is most marked when the most aggressive stopping criterion is applied, leading to the shallowest pool. Table 8.2 summarizes the results for this “worst” case.

Table 8.2: **"Worst-case" performance in reduced pool (minimum τ and maximum RMS error)**

track	**Kendall's τ**			**RMS error(ϵ)**		
	τ_{min}	E	R	ϵ_{max}	E	R
INEX 2007	0.9671	0.5626	0.9387	0.0179	0.5573	0.9385
INEX 2008	0.9425	0.5275	0.9216	0.0321	0.5275	0.9216

E denotes the fraction of the original assessment effort required when using a reduced pool and R denotes the ratio of nrels in the reduced qrels to that in the baseline qrels [1].

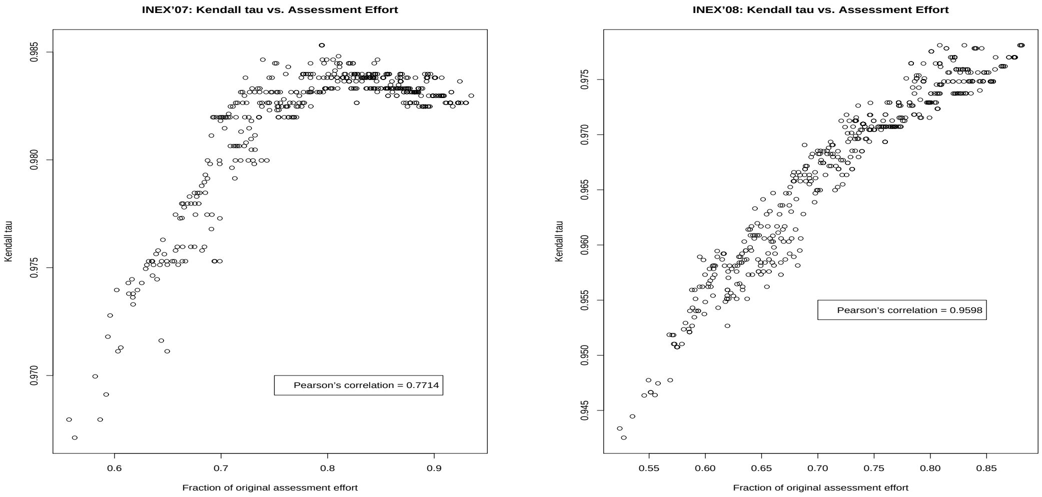
With about half of the original effort (a little over 50%), one can identify more than 90% of the reldocs, and obtain reliable system rankings (Kendall's $\tau > 0.94$ when compared to the baseline) and RMS error (ϵ) of the order of 0.03 in MAP values. Table 8.2 shows the minimum τ and the maximum ϵ (or ϵ_{max}) obtained for the 500 cases considered in each task. Needless to say, The τ-values are, in general, much above this minimum (τ_{min}). Actual ranges of Kendall's τ, AP correlation (τ_{AP}) and RMS error (ϵ) obtained for the 500 cases are shown in Table 8.3.

Table 8.3: **Ranges of τ, τ_{AP} and ϵ for INEX data**

Data	τ		τ_{AP}		ϵ	
	min	max	min	max	min	max
INEX 2007	0.9671	0.9853	0.9482	0.9862	0.0066	0.0179
INEX 2008	0.9425	0.9781	0.9132	0.9780	0.0048	0.0321

As expected, RMS error is found on the whole to be inversely proportional to the assessment effort, and Kendall's τ is proportional to the assessment effort as depicted in the curves (Fig. 8.4 - 8.5). The figures also show the Pearson's correlation coefficient between ϵ (resp. τ) and assessment effort, computed over the 500 cases we considered.

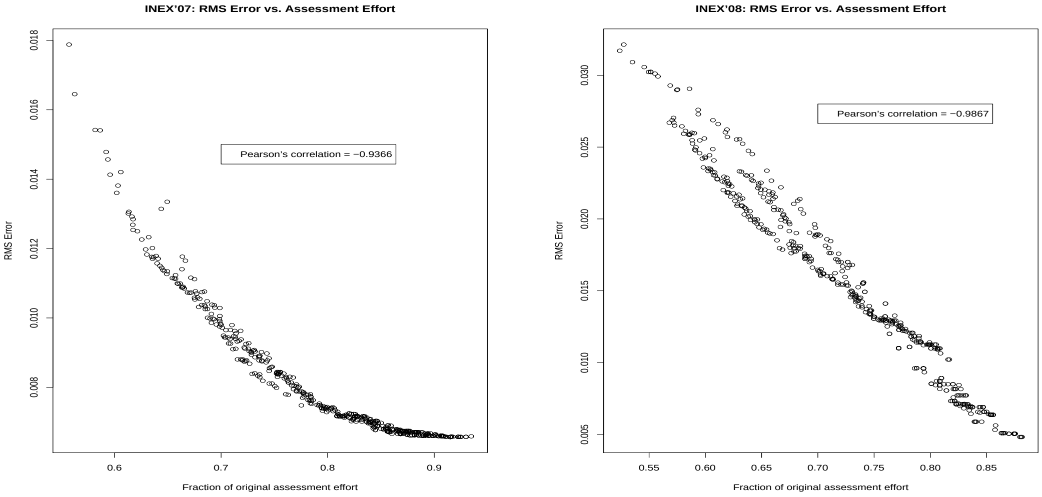
[1] a close clone of original qrels, see Sec. 5.5



(a) INEX 2007

(b) INEX 2008

Figure 8.4: Kendall's τ between rankings vs. assessment effort with INEX data



(a) INEX 2007

(b) INEX 2008

Figure 8.5: RMS error in MAP vs. assessment effort with INEX data

8.6 Generalization to Document Retrieval settings

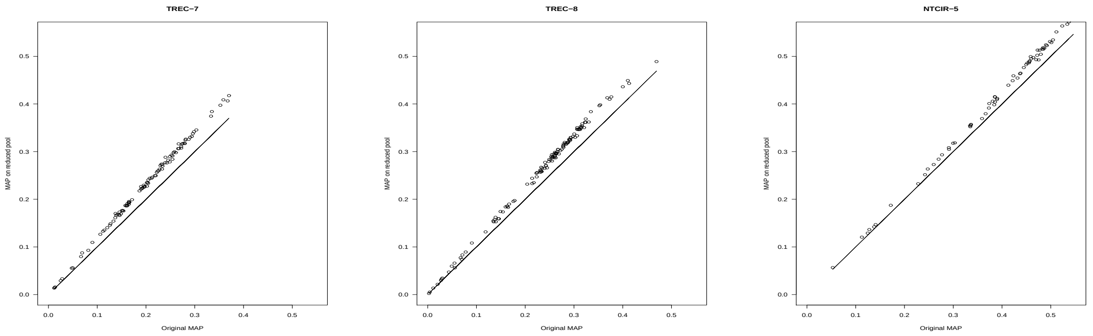
Interestingly, the above results are not restricted to an XML retrieval setting only. Following the observations in Section 8.3, we extend the work to a standard document retrieval setting using TREC and NTCIR data and the same experimental setup.

8.6.1 Data Used

Our algorithm is tested against the adhoc test collection of TREC-7 (topics 351-400, 103 runs), TREC-8 (topics 401-450, 129 runs) and adhoc cross-lingual test collections of NTCIR-5 (topics 1-50, 67 X-E runs, where X stands for Japanese, Chinese, Korean or English and E denotes English). The original qrels taken from the respective evaluation fora are used as the baseline judgments.

8.6.2 Results

The results are very much similar in nature to what we obtained for XML retrieval. Figure 8.6 shows the worst-case scenarios (most aggressive stopping of pool) for document retrieval evaluation with different collections.



(a) TREC-7

(b) TREC-8

(c) NTCIR-5

Figure 8.6: MAP values in reduced vs. original pool in document retrieval (*aggressive stopping*: $w = 6$, $W = 2$, $t = 0.80$, $l = 3$)

The document retrieval scenario is at least as promising as XML retrieval. As before, MAP values on the reduced pool closely follow the original baseline scores. This is reflected in higher Kendall's τ values and comparable RMS errors (Table 8.4).

Table 8.4: **"Worst-case" performance in reduced pool for document retrieval (minimum τ and maximum RMS error)**

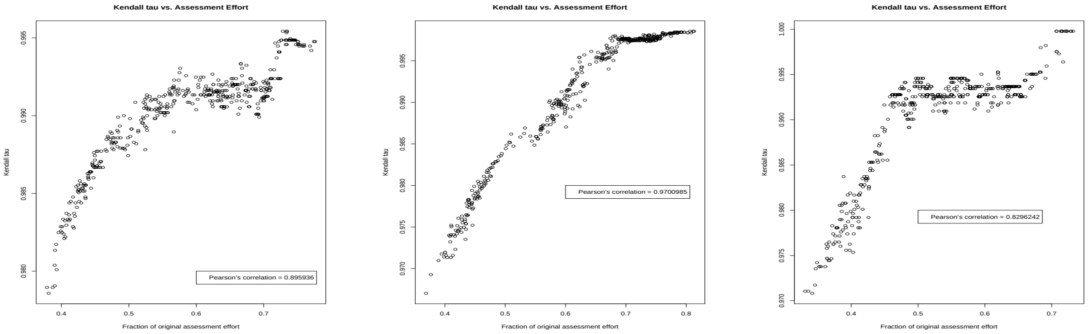
track	**Kendall's τ**			**RMS error(ϵ)**		
	τ_{min}	E	R	ϵ_{max}	E	R
TREC-7	0.979	0.381	0.847	0.033	0.379	0.846
TREC-8	0.967	0.368	0.821	0.030	0.369	0.821
NTCIR-5	0.970	0.341	0.850	0.026	0.331	0.846

As observed with XML data Kendall's τ in general is much above this minimum (τ_{min}). Actual ranges for Kendall's τ, AP correlation (τ_{AP}) and RMS error (ϵ) are shown in Table 8.5. Note that the performance is even better compared to XML retrieval setting (Table 8.3).

Table 8.5: **Ranges of τ, τ_{AP} and ϵ for document retrieval**

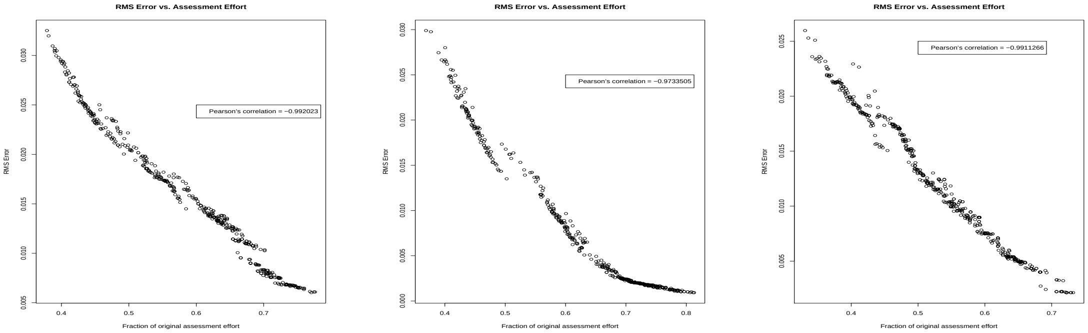
Data	τ		τ_{AP}		ϵ	
	min	max	min	max	min	max
TREC-7	0.979	0.995	0.974	0.994	0.006	0.033
TREC-8	0.967	0.999	0.961	0.998	0.001	0.030
NTCIR-5	0.970	0.999	0.972	1.0	0.002	0.026

The relation between Kendall's τ and assessment effort, and that between RMS error and assessment effort in the document retrieval scenario are also seen to resemble those in the XML retrieval evaluation setup (Fig. 8.7 - 8.8).



(a) TREC-7 (b) TREC-8 (c) NTCIR-5

Figure 8.7: Kendall τ vs. assessment effort in document retrieval scenario



(a) TREC-7 (b) TREC-8 (c) NTCIR-5

Figure 8.8: RMS error vs. assessment effort in document retrieval scenario

8.7 Discussion

Small pools generated through shallow pooling are generally used when early-precision is important and systems are evaluated using an early-precision metric. Estimates of recall take a beating here since a number of relevant documents are absent from such pools, and overall system performance cannot be reliably measured. How a smaller pool can be created without sacrificing recall is therefore an important question. In our approach, the above issue is reasonably addressed since we do not abruptly stop pooling by applying a *single-depth-for-all* queries. Rather, pooling effort is reduced on a *per-query-basis* based on the rate of finding relevant documents. The smaller pool thus generated is expected not to be unduly biased against recall-oriented systems. To verify this hypothesis, we evaluated and ranked the systems once again for each of the 500 different pools, using a recall-oriented metric, viz. recall@1000. Table 8.6 shows that the results obtained using this metric are similar to that obtained with MAP.

Table 8.6: **Ranges of τ, τ_{AP} and ϵ using Recall@1000**

Data	τ		τ_{AP}		ϵ	
	min	max	min	max	min	max
INEX'07	0.921	0.941	0.907	0.937	0.028	0.037
INEX'08	0.945	0.989	0.948	0.985	0.006	0.035
TREC-7	0.950	0.988	0.930	0.987	0.016	0.070
TREC-8	0.936	0.992	0.933	0.994	0.003	0.057
NTCIR-5	0.967	0.995	0.963	0.997	0.002	0.025

This confirms that the smaller pool generated by our method is not unduly biased against recall-oriented systems.

This pooling method also proves to be more effective in terms of assessment effort than our earlier shallow pooling technique (cf. reducing pool depth as part of pool sampling in Sec. 5.5). Here, we are able to extract more than 90% of the reldocs with about 50% effort, whereas earlier, shallow pooling (*single-depth-for-all* queries) would cost about 75% effort for the same (see Tables 5.4

and 5.5 in Chapter 5).

The results are even more promising in a document retrieval setting. With less than 40% of the original effort, one can identify more than 80% of the relevant documents, with reliable system rankings (Kendall's $\tau > 0.96$ when compared to the baseline which is slightly higher compared to XML retrieval evaluation) and RMS error in MAP less than 0.03 among n systems (Table 8.4).

Nevertheless, it can be argued that the proposed method devotes more judging effort to topics with a high number of relevant documents and less effort to topics with a low number of relevant documents (say about 20 or less). As a result, fewer relevant documents may be found for queries that have only a small number of relevant documents in the collection. Losing information about a single relevant document is much more detrimental in the case of these low-yield queries than for high-yield queries (having say 100+ relevant documents).

We investigate this problem in a document retrieval setting. An interesting pattern can be observed if the ratio nrels/pool-size (or, number of relevant documents found per document pooled) is plotted against pool-depth. For a large number of queries, the ratio starts with a moderate to high value and then falls off asymptotically, while for some queries, it makes a weak start and/or maintains a very low rate of decrease. Out of this latter class of queries, some are indeed low-yield ones, while for others, relevant documents are dispersed across the pool with long *dry runs* in between (i.e., no relevant documents are found for a long succession of pool-depths). When an aggressive stopping criterion is used, pooling tends to stop early for these queries, potentially causing problems in recall estimation.

We empirically find that the nrels/pool-size ratio remains less than or equal to 0.1 at a pool-depth of 20 (and decreases further for greater pool depths) for the majority of this second class of queries. If we leave aside these queries when reducing pool-depth on a per-query basis (i.e., we use the full pool for these queries), and consider only queries that cross the above threshold at depth 20 for pool reduction, we get better estimates of the total number of relevant documents compared to those obtained using an aggressive stopping criterion for all queries. The gain is particularly prominent for low-yield queries. This observation is true across the document collections (TREC-

7, TREC-8 and NTCIR-5).[2] Table 8.7 shows the E and R values obtained for various collections using this modified approach.

Table 8.7: **Correction for low-yield queries** (use depth = 100 if query has nrels-per-pooled-doc ≤ 0.1 at depth $= 20$)

Data	**% of low-yield queries**[1]	**Stop Criterion**	E	R
TREC-7	83.33 (15 out of 18)	most aggressive (6.2.0.80.3)	0.78	0.900
		most relaxed (14.6.0.05.3)	0.93	0.998
TREC-8	88.24 (15 out of 17)	most aggressive (6.2.0.80.3)	0.78	0.902
		most relaxed (14.6.0.05.3)	0.97	1.0
NTCIR-5	78.95 (15 out of 19)	most aggressive (6.2.0.80.3)	0.71	0.880
		most relaxed (14.6.0.05.3)	0.91	0.998

[1] roughly lower 33 percentile of queries ordered by nrels

If we look only at the fraction of the total number of relevant documents found (R), the overall gain in recall is costlier in terms of effort (E) for this modified approach (compare Table 8.4). The method still manages to save over 20% of assessment effort, however, while finding more than 90% of the known relevant documents. Importantly, this method safeguards against the danger of losing information about relevant documents for low-yield queries: for the identified low-yield queries, *all* relevant documents were found. Thus, this approach appears to be a practical means of reducing the effort required to obtain a robust and fair pool.

[2] Since INEX adhoc pools are constructed using much lower pool-depth values in general (little over 30 on average compared to 100 at TREC), we cannot expect to find a stable / comparable pattern for INEX queries. However, it may be possible to set the threshold in terms of a percentage of the original pool-depth.

8.8 Limitations and Future Work

At INEX, indexing, retrieval and evaluation are, in general, done at sub-document level. But pools are created and assessed at the document level. A document is included in the pool and judged in entirety even if a very small fraction of it is retrieved by any participating system. The pool therefore potentially contains some relevant document fragments which are not retrieved by any of the participating systems. While this contributes to the robustness and reusability of the pool, it also differentiates it from standard document retrieval pools. The effect of this subtle difference in the INEX pool, specifically on the evaluation of element retrieval, is not considered in our experiments. Our study here focuses only on the document level as pooling and assessments for the INEX adhoc tasks are done at that level. We believe that similar results would be obtained if evaluation were done entirely at the sub-document level (the observations from pool sampling in Chapter 5 support this claim); however, we are yet to experimentally ascertain this hypothesis.

Also, one can look into the reusability of the smaller pools obtained here for the evaluation of a 'new' system. In the line of Chapter 7, a 'leave-one-group-out' study can be done on the smaller pools. Given the robustness against both early precision and late recall metrics, however, we believe the pool can be safely used to judge 'new' systems.

8.9 Conclusion

Unlike other low-cost evaluation proposals, our method is not based on statistical sampling, neither is it specific to any measure(s), nor does it look for a few good topics. Within the traditional framework of the Cranfield paradigm – a time-tested general framework which provides reusability for judging 'new' systems as well as for post-hoc diagnosis of a system – the method offers an interactive pooling approach based on variable pool depth per query. It reduces assessment effort to a great extent for most of the queries where the pool saturates quickly. Again, for the queries where the rate of finding new reldocs is quite high, better estimates of recall can be ensured by going deeper in the pool ($k > 100$). By selecting the four parameters w, W, t and l suitably, an appropriate trade-off can be achieved between the cost of evaluation and its reliability. Even with the most aggressive stopping criterion (worst-case scenario with small w, W, l and high t), a highly reliable and fair pool can be obtained. To build a large test collection based on the Cranfield methodology, our simple approach can be cost-effective yet reliable.

Chapter 9

Conclusions

Since XML as a format for semi-structured data provides a very powerful and flexible data representation method that subsumes both traditional relational databases and HTML, it has emerged as a widely-used standard for data organization, representation and exchange on the Web and in digital libraries. Present day information repositories, which are a mixture of text, multimedia, and metadata, increasingly contain both long as well as heterogeneous documents, covering a wide variety of topics, e.g. books, user manuals, legal documents, and customer feedback data – and are marked up in XML. Owing to the mark-up, the documents may be regarded as aggregates of smaller, hierarchically structured entities that are separately indexable and retrievable [22]. How to effectively and efficiently store, index, query and retrieve these entities has been an active area of research for both the Database (DB) and Information Retrieval (IR) communities in recent times. For the IR community, XML retrieval poses a two-fold problem:

- finding effective techniques to retrieve appropriate or the most useful XML elements in response to a user query; and
- devising an appropriate evaluation methodology to measure the effectivity of such retrieval techniques.

In this thesis, we touched both the issues. First, we revisited the question of length normalization for the Vector Space Model [130] in the context of XML retrieval using a standard benchmark collection. We reduced two parameters used in pivoted length normalization to a single combined parameter and experimentally found its “optimum” value, which works well at both element and document levels for XML retrieval.

We observed that a substantial number of focused queries used in XML retrieval clearly state, besides the information need, what the user *does not* want. We demonstrated that this negative information, if not handled properly, degrades retrieval performance. We proposed a scheme for automatically removing negative information from XML queries, which led to significant improvements in retrieval results.

With regard to XML retrieval evaluation, we carried out an elaborate study of the pooling method and evaluation metrics used for the INEX adhoc tasks. focus was on the reliability and reusability of the pool and the metrics. We also proposed a cost-effective reusable pooling technique which can be used for evaluation of XML retrieval as well as traditional document retrieval.

9.1 Results

Length Normalization. Since several standard IR techniques have successfully been implemented in the realm of Web retrieval, there have been attempts to apply the techniques in XML retrieval as well. The Vector Space Model is an effective retrieval model that has been used for many years in document retrieval. Among the three components of the term-weighting formula, namely, *term frequency*, *document frequency* and *length normalization*, the third one largely depends on the nature of collection. Singhal [130] studied the issue in detail for document retrieval using TREC collections and proposed pivoted document length normalization with recommendations for the optimum values of its parameters, namely *pivot* and *slope*. Although the recommended values work well across different collections in the case of document retrieval, XML element retrieval is a different paradigm altogether. An element as a retrievable unit can range from a few words at the leaf node to the whole XML document. Using a single average length as *pivot* for these widely varying elements does not seem justified; also the Wikipedia XML collections that are used for the INEX adhoc task are both "syntactically" and "semantically" different from the TREC collections. We therefore needed a re-look into pivoted length normalization. We combined the two parameters – *pivot* and *slope* – and considered a single *pivot-slope* factor. We varied the factor and observed the element retrieval performance and found an optimum value. The optimum value not only worked well for element retrieval, it also yielded good performance for XML document retrieval across the collections (INEX 2006 - 2009).

Query Refinement. At INEX, we observed that topic creators often vividly state, besides the information need, what they *do not* want in the verbose narrative section. Most text retrieval systems do not handle this *negative information* properly. We investigated whether this information can help to improve retrieval performance by first manually removing negative information from the INEX queries. We observed that retrieval performance improved by 4-6% across different INEX collections compared to using the original queries. The improvement was found to be statistically significant. We then explored whether the process of negation detection and removal could be automated. With the help of a maximum entropy classifier, we segregated positive and negative sentences in the narrative sections of INEX topics. We formed a set of positive queries taking only the positive sentences. These queries yielded *equivalent* retrieval performance compared to the manual method (difference not statistically significant).

Evaluation. Evaluation is a gruelling challenge for XML retrieval research. Ever since the inception of INEX, its evaluation measures have changed at regular intervals. With the inclusion of arbitrary passages as valid retrieval units besides the usual XML elements, the need for a common set of effectiveness measures has gained importance. INEX 2007 therefore introduced a set of precision-recall based measures for its adhoc tasks. Our aim was to investigate the reliability and robustness of these focused retrieval measures, as well as the pooling method used at INEX. Our experiments were mainly driven by the common objective of finding a reliable, optimum evaluation setup in terms of the assessment effort required, and the evaluation measures used to rank a set of XML retrieval systems at INEX. The four specific questions that we investigated and observations we made with respect to them are summarized below.

1. *How reliable are the various metrics in ranking competing systems when assessments are incomplete?*

   The results of our experiments validate properties of precision-recall-based metrics that were originally observed in a document retrieval setting. For example, our experiments reaffirm that early precision measures ($iP[0.00]$, $iP[0.01]$) are more error-prone and less stable under incomplete judgments, whereas AiP is the least vulnerable among these metrics. Specifically, $MAiP$ -based rankings remain largely unaltered ($\tau \geq 0.9$), even when evaluation effort is halved.

On a related note, what is the minimum pool size that can be used to reliably evaluate systems?

   As evaluation is strongly dependent on a relatively small set of top-ranked results, rankings similar to the official rankings can be obtained even when pooling is limited to about 15% of the currently used pool depth.

2. *How reliable are the various metrics in ranking competing systems if the query set size is small? What is the minimum number of queries that should be used to keep the error rates for the various metrics within a maximum allowable upper bound?*

   As in (1) above, we find that early precision measures ($iP[0.00]$, $iP[0.01]$) are more error-prone, and less stable if a small topic set is used, whereas AiP is the most stable. Our experiments also suggest that the pool size and number of queries used since INEX 2007 are large enough to reliably evaluate all submissions, i.e. the INEX results generally contain less than 5% error for all the metrics reported.

3. *When a set of relevance assessments is used to evaluate a new system that did not contribute to the pool used in the relevance assessment process, are the results biased against this system?*

   Our investigation into the effect of bias towards a system due to its contribution to the pool suggests that a new system which did not contribute to the INEX 2007 or INEX 2008 pool can be fairly evaluated on the basis of the corresponding qrels. In most cases, contributing systems get insignificant advantages due to their contribution to the qrels.

4. *For a fixed amount of assessment effort, would this effort be better spent in thoroughly judging a few queries, or in judging many queries relatively superficially?*

   We observe that, to reduce the amount of effort required to create usable qrels, it is better to judge shallower pools for all topics, rather than reduce the number of topics that are judged. However, it is better to completely judge a smaller number of topics, than to partially judge many topics in random order.

   Available manpower may be utilised even more effectively by choosing the pool depth / pool size on a per query basis. We observed that the rate of finding new relevant documents decreases with increase in pool depth. The rate, however, varies widely across the queries. For a substantial number of queries, the rate drops to zero quite early. For these queries, if we can stop the pooling early, we can save a lot of assessment effort. The effort thus saved can be used for assessing queries that have a high rate of finding new relevant documents. This approach offers a possibility of obtaining better estimates of the 100% recall level for all queries. We proposed a heuristic technique to incrementally build an evaluation pool which can be adapted based on available assessment effort and the required level of reliability. We showed that our technique can save about 50% - 60% assessment effort without seriously compromising on the reliability of evaluation. The technique involved substantially less effort than usual shallow pooling methods needs to achieve the same level of reliability in system evaluation. We demonstrated that the method worked well in the evaluation of XML retrieval. Moreover, we observed that it generalises well to a document retrieval paradigm with even more promising results.

9.2 Recommendations

We recommend the following as a result of our study.

- The pivoted document length normalization factor in the *Lnu.ltn* scheme [129], should be considered for XML retrieval as

$$normalization = 1 + \alpha \times (\text{\# \textit{unique terms}})$$

where

$$\alpha = \frac{slope}{(1 - slope) \times pivot}$$

and element length is given by *# unique terms* in the element.

Without loss of generality, the $pivot$ parameter may be set to 1. A setting of $slope = 0.00073$ works well as the slope value for both element and document retrieval.

- Among the five metrics we studied in the context of INEX adhoc tasks, the early precision metrics ($iP[0.00]$, $iP[0.01]$) are not unstable and error-prone. If system ranking is done by any of these early-precision metrics, the result needs to taken with care. The problem is serious when the pool is incomplete, either due to incomplete judgments for some queries, or the pool contain less judgments for small number of topics. However, in both the situations, $MAiP$ performs in the most stable and reliable manner.

- The minimum number of queries required for reliable XML evaluation increases from a *stable* metric ($MAiP$) to *unstable* early precision metrics (in the order $iP[0.10]$, $iP[0.05]$, $iP[0.01]$, $iP[0.00]$). To ensure a reliable and stable system ranking with less than 5% error (tolerance in score -difference ± 5%) the recommended number of queries needed are summarized in Table 9.1:

Table 9.1: **Min. #queries reqd. for evaluation with stable system ranking ($\tau \geq 0.9$) and moderately acceptable (tolerance ± 5%) error rate ($<$ 5%)**

Metric	#queries
$iP[0.00]$	> 60
$iP[0.01]$	> 58
$iP[0.05]$	> 53
$iP[0.10]$	> 44
$MAiP$	> 35

- The following parameter settings may be used for *aggressive* stopping during pool creation using our incremental technique: These settings ensure reasonably good estimates of recall.

moving average window size for raw nrels,	w	=	6
moving average window size for rate of finding new reldocs,	W	=	2
threshold for minimum allowable rate of finding new reldocs,	t	=	0.8
run of pool depth where rate of finding reldocs $< t$,	l	=	3.

However, to ensure that early stopping does not result in fewer relevant documents found for queries that have only a few relevant documents, the technique should not be applied to queries which have nrels-per-pooled-doc < 0.1 at pool-depth = 20.

9.3 Future Directions

In this section, we summarize some issues arising out of our work that could be further investigated in the future.

- In our exploration of techniques for XML retrieval, we chose the Vector Space Model along with the pivoted length normalization scheme. We soon realized , however, that the wide range of retrieval granularity invloved in XML retrieval dictates a revisit to the issue of length normalization. The engineering solution we propose here works quite well for XML element and document retrieval. However, we believe that a thorough investigation into the term-weighting strategies of VSM in the XML retrieval context is need of the hour.
- In our study on query refinement, we simply discarded the negative constraints identified either manually or automatically. We believe, however, that the negative information can be used more proactively by a retrieval system to further improve performance. For example, the query may be modified suitably by assigning proper weights to different kinds of query terms: positive, negative and weak. How to automatically identify and weight these terms can be explored in future work.
- Our experiments with reduced pool – either reducing the pool depth for all queries *or* stopping pool creation on a per query basis – demonstrated that rankings do not change significantly even when the pool size is substantially reduced. However, if such small pools were

actually used in practice, it would be interesting to study whether new systems could still be reliably evaluated on the basis of the resultant relevance assessments. In other words, it might be instructive to redo the Leave One Out experiments using these reduced pools as a starting point.

Bibliography

[1] INitiative for the Evaluation of XML retrieval (INEX), 2011. https://inex.mmci.uni-saarland.de/.

[2] Javed A. Aslam, Virgil Pavlu, and Emine Yilmaz. A statistical method for system evaluation using incomplete judgments. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 541–548, New York, NY, USA, 2006. ACM.

[3] Ricardo Baeza-Yates, David Carmel, Yoelle Maarek, and Aya Soffer, editors. *Journal of the American Society for Information Science and Technlogy*, volume 53, 2002.

[4] Ricardo Baeza-Yates, Norbert Fuhr, and Yoelle S. Maarek. Second edition of the "XML and information retrieval" workshop held at SIGIR 2002, Tampere, Finland, Aug 15th, 2002. *SIGIR Forum*, 36:53–57, September 2002.

[5] Niranjan Balasubramanian, Giridhar Kumaran, and Vitor R. Carvalho. Exploring reductions for long web queries. In *Proc. 33rd ACM SIGIR*, pages 571–578. ACM, 2010.

[6] David Banks, Paul Over, and Nien-Fan Zhang. Blind men and elephants: Six approaches to trec data. *Inf. Retr.*, 1(1-2):7–34, May 1999.

[7] N. J. Belkin, D. Kelly, G. Kim, J.-Y. Kim, H.-J. Lee, G. Muresan, M.-C. Tang, X.-J. Yuan, and C. Cool. Query length in interactive information retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 205–212, New York, NY, USA, 2003. ACM.

[8] Michael Bendersky and W.B. Croft. Discovering key concepts in verbose queries. In *Proc. 31st ACM SIGIR*, pages 491–498. ACM, 2008.

[9] Michael K. Bergman. The Deep Web: Surfacing Hidden Value. `http://www.press.umich.edu/jep/07-01/bergman.html`, 2001.

[10] Chris Buckley. The importance of proper weighting methods. In M. Bates, editor, *Human Language Technology*. Morgan Kaufman, 1993.

[11] Chris Buckley, Amit Singhal, and Mandar Mitra. Using Query Zoning and Correlation within SMART: TREC5. In E.M. Voorhees and D.K. Harman, editors, *Proc. Fifth Text Retrieval Conference (TREC-5)*. NIST Special Publication 500-238, 1997.

[12] Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40, New York, NY, USA, 2000. ACM.

[13] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 25–32, New York, NY, USA, 2004. ACM.

[14] Stefan Büttcher, Charles L. A. Clarke, Peter C. K. Yeung, and Ian Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 63–70, New York, NY, USA, 2007. ACM.

[15] D. Carmel, Y. Maarek, and A. Soffer. XML and information retrieval: A SIGIR 2000 workshop. In *ACM SIGMOD Record*, volume 30(1), pages 62–65, 2001.

[16] David Carmel, Yoelle S. Maarek, Matan Mandelbrod, Yosi Mass, and Aya Soffer. Searching XML documents via XML fragments. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 151–158, New York, NY, USA, 2003. ACM.

[17] Ben Carterette. Robust test collections for retrieval evaluation. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 55–62, New York, NY, USA, 2007. ACM.

[18] Ben Carterette, James Allan, and Ramesh Sitaraman. Minimal Test Collections for Retrieval Evaluation. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 268–275, New York, NY, USA, 2006. ACM.

[19] Ben Carterette, Virgil Pavlu, Evangelos Kanoulas, Javed A. Aslam, and James Allan. Evaluation over thousands of queries. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 651–658, New York, NY, USA, 2008. ACM.

[20] D. Chamberlin. XQuery: An XML Query Language. In *IBM Systems Journal*, volume 41(4), pages 597–615, 2002.

[21] Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. *J. Biomed. Inform.*, 34(5):301–310, 2001.

[22] Yves Chiaramella. Information Retrieval and Structured Documents. In *ESSIR '00: Proceedings of the Third European Summer-School on Lectures on Information Retrieval-Revised Lectures*, pages 286–309, London, UK, 2001. Springer-Verlag.

[23] Charles L.A. Clarke, Nick Craswell, Ian Soboroff, and Azin Ashkan. A comparative analysis of cascade measures for novelty and diversity. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 75–84, New York, NY, USA, 2011. ACM.

[24] W. S. Cooper. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American documentation*, 19(1):30–41, 1968.

[25] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. Efficient construction of large test collections. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 282–289, New York, NY, USA, 1998. ACM.

[26] Carolyn J. Crouch. Dynamic element retrieval in a structured environment. *ACM Trans. Inf. Syst.*, 24(4):437–454, 2006.

[27] Ludovic Denoyer and Patrick Gallinari. The Wikipedia XML corpus. *SIGIR Forum*, 40(1):64–69, 2006.

[28] Susan T Dumais. *LSI meets TREC: A Status Report*, volume 500, pages 137–152. NIST, 1993.

[29] Susan T Dumais. *Latent semantic indexing (LSI) and TREC-2*, volume 500, pages 105–115. NIST, 1994.

[30] M. Dunlop. Time, Relevance and Interaction Modelling for Information Retrieval. In *Proceedings of the 20th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–213, Philadelphia, PA, USA, 1997. ACM Press.

[31] Khairun Nisa Fachry, Jaap Kamps, Rianne Kaptein, Marijn Koolen, and Junte Zhang. The University of Amsterdam at INEX 2008:Ad Hoc, Book, Entity Ranking, Interactive, Link the wiki and XML Mining Tracks. In *INEX 2008 Workshop Pre-Proceedings*, pages 66–91, Dagstuhl, Germany, 2008.

[32] Joel Fagan. *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods*. PhD thesis, Department of Computer Science, Cornell University, September 1987.

[33] Norbert Fuhr. A probabilistic framework for vague queries and imprecise information in databases. In *Proceedings of the sixteenth international conference on Very large databases*, pages 696–707, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[34] Norbert Fuhr, Jaap Kamps, Mounia Lalmas, Saadia Malik, and Andrew Trotman. Overview of the INEX 2007 Ad Hoc Track. pages 1–23, 2008.

[35] Norbert Fuhr, Mounia Lalmas, and Andrew Trotman, editors. *Comparative Evaluation of XML Information Retrieval Systems, 5th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2006, Dagstuhl Castle, Germany, December 17-20, 2006, Revised and Selected Papers*, volume 4518 of *Lecture Notes in Computer Science*. Springer, 2007.

[36] Debasis Ganguly, Johannes Leveling, Gareth Jones, Sauparna Palchowdhury, Sukomal Pal, and Mandar Mitra. DCU&ISI@INEX 2010: Adhoc, Data-centric and Feedback Tracks. In *INEX 2010 proceedings*, volume 6932, pages 182–193. Springer, 2010.

[37] Shlomo Geva, Jaap Kamps, Miro Lehtonen, Ralf Schenkel, James A. Thom, and Andrew Trotman. Overview of the INEX 2009 Ad Hoc Track. In Geva et al. [39], pages 4–25.

[38] Shlomo Geva, Jaap Kamps, and Andrew Trotman, editors. *Advances in Focused Retrieval, 7th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2008, Dagstuhl Castle, Germany, December 15-18, 2008. Revised and Selected Papers*, volume 5631 of *Lecture Notes in Computer Science*. Springer, 2009.

[39] Shlomo Geva, Jaap Kamps, and Andrew Trotman, editors. *Focused Retrieval and Evaluation, 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2009, Brisbane, Australia, December 7-9, 2009, Revised and Selected Papers*, volume 6203 of *Lecture Notes in Computer Science*. Springer, 2010.

[40] Sergey Goryachev, Margarita Sordo, Qing T. Zeng, and Long Ngo. Implementation and evaluation of four different methods of negation detection. Technical report, 2006.

[41] N. Gövert and G. Kazai. Overview of the INitiative for the Evaluation of XML retrieval (INEX) 2002. In *N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas, editors, Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX)*, pages 1–17, Dagstuhl, Germany, 2003.

[42] N. Gövert, G. Kazai, N. Fuhr, and M. Lalmas. Evaluating the effectiveness of content-oriented XML retrieval. Technical Report Technischer bericht, Computer Science 6, University of Dortmund, 2003.

[43] Norbert Gövert, Norbert Fuhr, Mounia Lalmas, and Gabriella Kazai. Evaluating the effectiveness of content-oriented xml retrieval methods. *Inf. Retr.*, 9(6):699–722, 2006.

[44] T. Grabs and H.-J. Schek. Generating vector spaces on-the-fly for flexible XML retrieval. In *Proceedings of the SIGIR 2002 Workshop on XML and Information Retrieval*, pages 4–13, 2002.

[45] John Guiver, Stefano Mizzaro, and Stephen Robertson. A few good topics: Experiments in topic set reduction for retrieval evaluation. *ACM Trans. Inf. Syst.*, 27:21:1–21:26, November 2009.

[46] M. Hassler and A. Bouchachia. Searching XML Documents - Preliminary Work. In *Pre-proceedings of the 4th workshop of the initiative for the evaluation of XML retrieval (INEX)*, pages 95–109, 2005.

[47] Bin He, Mitesh Patel, Zhen Zhang, and Kevin ChSen-Chuan Chang. Accessing the deep web: A Survey. *Communications of the ACM*, 50(5):94–101, 2007.

[48] Fang Huang, Stuart N. K. Watt, David J. Harper, and Malcolm Clark. Compact representations in xml retrieval. In Fuhr et al. [35], pages 64–72.

[49] G. Hubert. XML retrieval based on direct contribution of query components. In *Pre-proceedings of the 4th workshop of the initiative for the evaluation of XML retrieval (INEX)*, pages 138–149, 2005.

[50] D. Hull. Using Statistical Testing in the Evaluation of Retrieval Experiments. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 329–338, Pittsburgh, PA, USA, 1993. ACM Press.

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

[52] Jaap Kamps, Shlomo Geva, Andrew Trotman, Alan Woodley, and Marijn Koolen. Overview of the INEX 2008 Ad Hoc Track. In *Advances in Focused Retrieval: 7th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2008 Dagstuhl Castle, Germany, December 2008, Revised and Selected Papers*, volume 5631 of *Lecture Notes in Computer Science*, pages 1–28, Berlin, Heidelberg, 2009. Springer-Verlag.

[53] Jaap Kamps, Maarten Marx, Maarten de Rijke, and Börkur Sigurbjörnsson. Articulating information needs in XML query languages. *ACM Trans. Inf. Syst.*, 24(4):407–436, 2006.

[54] Jaap Kamps, Jovan Pehcevski, Gabriella Kazai, Mounia Lalmas, and Stephen Robertson. INEX 2007 Evaluation Measures. In *Focused Access to XML Documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007 Dagstuhl Castle, Germany, December 2007, Revised and Selected Papers*, volume 4862 of *Lecture Notes in Computer Science*, pages 24–33, Berlin, Heidelberg, 2008. Springer-Verlag.

[55] G. Kazai and M. Lalmas. Notes on what to measure in inex. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, pages 22–38, Glasgow,UK., 2005.

[56] G. Kazai, M. Lalmas, and S. Malik. INEX guidelines for topic development. In *Proceedings of the 1st workshop of the initiative for the evaluation of XML retrieval (INEX)*, pages 178–181, 2002.

[57] G. Kazai, M. Lalmas, and S. Malik. INEX ’03 guidelines for topic development. In *Proceedings of the 2nd workshop of the initiative for the evaluation of XML retrieval (INEX)*, 2003.

[58] G. Kazai, M. Lalmas, and A. P. Vries. The overlap problem in content-oriented XML retrieval evaluation. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 72–79, Sheffield, UK, July 2004. ACM.

[59] Gabriella Kazai. Report of the INEX 2003 metrics working group. In *INEX 2003 Workshop Proceedings*, pages 184–190, 2003.

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

[61] Gabriellla Kazai and Mounia Lalmas. INEX 2005 evaluation metrics. In Norbert Fuhr, Mounia Lalmas, and Andrew Trotman, editors, *Advances in XML Retrieval and Evaluation: 4th Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2005)*, volume Lecture Notes in Computer Science vol. 3977, pages 16–29. Springer-Verlag, 2006.

[62] Jaana Kekäläinen and Kalervo Järvelin. Using graded relevance assessments in IR evaluation. *Journal of the American Society for Information Science and Technology*, 53(13):1120–1129, 2002.

[63] M. Lalmas. INEX 2005 retrieval task and result submission specification. Technical report, Queen Mary, University of London, 2005.

[64] M. Lalmas and S. Malik. INEX 2004 retrieval task and result submission specification. In *N. Fuhr, M. Lalmas, S. Malik, Z. Szlavik, (Eds) Advances in XML Information Retreival. Third Workshop of the INitiative for the Evaluation of XML Retrieval INEX 2004*, volume 3493, pages 237–240, Schloss Dagstuhl, Germany, 2005. LNCS.

[65] M. Lalmas and B. Piwowarski. INEX 2006 relevance assessment guide, 2006.

[66] Mounia Lalmas, Gabriella Kazai, Jaap Kamps, Jovan Pehcevski, Benjamin Piwowarski, and Stephen Robertson. INEX 2006 evaluation measures. In Norbert Fuhr, Mounia Lalmas, and Andrew Trotman, editors, *Comparative Evaluation of XML Information Retrieval Systems: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2006)*, volume 4518 of *Lecture Notes in Computer Science*, pages 20–34. Springer Verlag, Heidelberg, 2007.

[67] Christine Largeron, Christophe Moulin, and Mathias Géry. UJM at INEX 2009 XML Mining Track. In Geva et al. [39], pages 426–433.

[68] M. Lehtonen. When a few highly relevant answers are enough. In *Pre-Proceedings of the Fourth Annual Workshop of the Initiative for the Evaluation of XML retrieval(INEX)*, pages 215–216, 2005.

[69] Alon Levy. More on Data Management for XML. `http://www.cs.washington.edu/homes/alon/widom-response.html`, 1999.

[70] David D. Lewis and Karen Sparck Jones. Natural language processing for information retrieval. *Commun. ACM*, 39(1):92–101, 1996.

[71] Andrea Carneiro Linhares and Patricia Velazquez. Using textual energy (enertex) at qaine-track 2010. In *INEX 2010 Workshop Pre-Proceedings*, pages 234–237, the Netherlands, 2010.

[72] Julie Beth Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.

[73] H.P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317, October 1957.

[74] R. Luk, A. Chan, T. Dillon, and H.V. Leong. A survey of Search Engines for XML documents. `http://www.haifa.il.ibm.com/sigir00-xml/final-papers/Luk/XMLSUR.htm`, 2000.

[75] R. W. P. Luk, H. V. Leong, T. S. Dillon, A. T. S. Chan, W.B. Croft, and J. Allan. A Survey in Inedexing and Searching XML Documents. *Journal of the American Society for Information Science and Technlogy*, 53(6):415–437, 2002.

[76] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[77] Martin-D Lacasse . FUDGIT version 2.41. http://hpux.connect.org.uk/hppd/hpux/Maths/Misc/fudgit-2.41/.

[78] Y. Mass and M. Mandelbrod. Retrieving the most relevant XML component. In *Proceedings of the Second Annual Workshop of the Initiative for the Evaluation of XML retrieval(INEX)*, pages 53–58, Dagstuhl, Germany, 2003.

[79] Y. Mass and M. Mandelbrod. Component Ranking and Automatic Query Refinement. In *INEX 2004, Lecture Notes in Computer Science*, volume 3493, pages 73–84. Springer-Verlag GmbH, 2005.

[80] April R. McQuire and Caroline M. Eastman. Ambiguity of negation in natural language queries to information retrieval systems. *Journal of the American Society for Information Science*, 49(8):686–692, April 1998.

[81] Mandar Mitra, Amit Singhal, and Chris Buckley. Improving automatic query expansion. In *SIGIR 98*, pages 206–214, Melbourne, Australia, 1998. ACM.

[82] Mandar Mitra, Amit Singhal, and Chris Buckley. Improving automatic query expansion. In *SIGIR 98*, pages 206–214, Melbourne, Australia, 1998. ACM.

[83] Alistair Moffat, William Webber, and Justin Zobel. Strategic system comparisons via targeted relevance judgments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’07, pages 375–382, New York, NY, USA, 2007. ACM.

[84] Alistair Moffat and Justin Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Inf. Syst.*, 27(1):1–27, 2008.

[85] P.G. Mutalik, A. Deshpande, and P.M. Nadkarni. Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the UMLS.(NegFinder). *Journal of the American Medical Informatics Association*, 8(6):598–609, 2001.

[86] Ziad S. Nakkouzi and Caroline M. Eastman. Query formulation for handling negation in information retrieval systems. *Journal of the American Society for Information Science*, 41(3):171–182, April 1990.

[87] Gonzalo Navarro and Ricardo Baeza-Yates. A language for queries on structure and contents of textual databases. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 93–101, New York, NY, USA, 1995. ACM.

[88] Gonzalo Navarro and Ricardo Baeza-Yates. Proximal nodes: a model to query document databases by content and structure. *ACM Trans. Inf. Syst.*, 15(4):400–435, 1997.

[89] Sukomal Pal and Mandar Mitra. Indian Statistical Institute at INEX 2007 Adhoc Track: VSM Approach. In Norbert Fuhr, Jaap Kamps, Mounia Lalmas, and Andrew Trotman, editors, *INEX*, volume 4862 of *Lecture Notes in Computer Science*, pages 122–128. Springer, 2007.

[90] Sukomal Pal and Mandar Mitra. *XML Retrieval: A Survey*, volume 8, pages 229–272. Nova Science Publishers, Inc., 2011.

[91] Sukomal Pal, Mandar Mitra, and Arnab Chakraborty. Stability of INEX 2007 Evaluation Measures. In Tetsuya Sakai and Mark Sanderson, editors, *Proc. of the Second International Workshop on EValuating Information Access (EVIA 2008), NTCIR 7*, pages 23–29, 2008. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings7/pdf/EVIA2008/06-EVIA2008-PalS.pdf.

[92] Sukomal Pal, Mandar Mitra, Debasis Ganguly, Samaresh Maiti, Ayan Bandyopadhyay, Aparajita Sen, and Sukanya Mitra. Indian Statistical Institute at INEX 2008 Adhoc Track. In Geva et al. [38], pages 79–86.

[93] Sukomal Pal, Mandar Mitra, and Jaap Kamps. Evaluation effort, reliability and reusability in XML retrieval. *JASIST*, 62(2):375–394, 2011.

[94] Sukomal Pal, Mandar Mitra, and Samaresh Maiti. Estimating Pool-depth on Per Query Basis. In Tetsuya Sakai, Mark Sanderson, and William Webber, editors, *Proc. of the Third International Workshop on EValuating Information Access (EVIA 2010), NTCIR 8*, pages 2–6, 2010. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings8/EVIA/02-EVIA2010-SukomalP.pdf.

[95] Sukomal Pal, Mandar Mitra, and Prasenjit Majumder. Indian Statistical Institute at INEX 2006 Adhoc Track: A Preliminary VSM Approach. In *INEX 2006 Workshop Pre-Proceedings*, pages 118–120, Dagstuhl, Germany, 2006.

[96] Sauparna Palchowdhury, Sukomal Pal, and Mandar Mitra. Using negative information in search. In *Second International Conference on Emerging Applications of Information Technology*, pages 53–56, Kolkata, 2011.

[97] J. Pehcevski and J.A. Thom. HiXEval: Highlighting XML Retrieval Evaluation. In *Pre-Proceedings of the Fourth Annual Workshop of the Initiative for the Evaluation of XML retrieval(INEX)*, pages 11–24, 2005.

[98] B. Piwowarski. EPRUM metrics and INEX 2005: Draft. In *Pre-Proceedings of the Fourth Annual Workshop of the Initiative for the Evaluation of XML retrieval (INEX)*, pages 1–10, 2005.

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

[100] Benjamin Piwowarski, Andrew Trotman, and Mounia Lalmas. Sound and complete relevance assessment for XML retrieval. *ACM Transactions on Information Systems*, 27(1):1–37, 2008.

[101] Anna Rafferty and Christopher Manning. Stanford classifier. http://nlp.stanford.edu/software/classifier.shtml.

[102] V. V. Raghavan, G. S. Jung, and P. Bollmann. A Critical Investigation of Recall and Precision a Measures of Retrieval System Performance. *ACM Transactions on Information Systems*, 7(3):205–229, July 1989.

[103] S. E. Robertson and S. Walker. Some simple effective approximations to the 2–poisson model for probabilistic weighted retrieval. In W.B. Croft and C.J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241. Springer-Verlag, July 1994.

[104] Stephen E. Robertson and Karen Sparck Jones. Relevance weighting of search terms. *Journal of American Society of Information Science*, 27:129–146, May-June 1976.

[105] Stephen E. Robertson, Evangelos Kanoulas, and Emine Yilmaz. Extending average precision to graded relevance judgments. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 603–610, New York, NY, USA, 2010. ACM.

[106] J. Rocchio(Jr.). Relevance feedback in information retrieval. In G.Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter 14, pages 313–323. Prentice Hall, Englewood Cliffs, New Jersey,USA, 1971.

[107] Lior Rokach, Roni Romano, and Oded Maimon. Negation recognition in medical narrative reports. *Inf. Retr.*, 11(6):499–538, 2008.

[108] T. Sakai. New performance metrics based on multigrade relevance: Their application to question answering. In *NTCIR Workshop 4 Meeting Working Notes*, June 2004.

[109] Tetsuya Sakai. Alternatives to bpref. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 71–78, New York, NY, USA, 2007. ACM.

[110] Tetsuya Sakai. Evaluating Information Retrieval Metrics Based on Bootstrap Hypothesis Tests. *IPSJ Digital Courier*, 3:625–642, 2007.

[111] Tetsuya Sakai. Comparing metrics across TREC and NTCIR:: the robustness to pool depth bias. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 691–692, New York, NY, USA, 2008. ACM.

[112] Tetsuya Sakai. Comparing metrics across TREC and NTCIR: the robustness to system bias. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 581–590, New York, NY, USA, 2008. ACM.

[113] Tetsuya Sakai and Noriko Kando. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval*, 11(5):447–470, 2008.

[114] Tetsuya Sakai and Ruihua Song. Evaluating diversified search results using per-intent graded relevance. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1043–1052, New York, NY, USA, 2011. ACM.

[115] G. Salton. A Blueprint for Automatic Indexing. *ACM SIGIR Forum*, 16(2):22–38, Fall 1981.

[116] G. Salton, A. Wong, and C.S. Yang. A Vector Space Model for Information Retrieval. *Communications of the ACM*, 18(11):613–620, November 1975.

[117] Gerard Salton. *Automatic text processing—the transformation, analysis and retrieval of information by computer*. Addison-Wesley Publishing Co., Reading, MA, 1989.

[118] Gerard Salton. A blueprint for automatic indexing. *ACM SIGIR Forum*, 2(16):22–38, Fall 1981.

[119] Gerard Salton and Chris Buckley. Term-weighting approaches in automatic text retrieval. 24(5):513–523, 1988.

[120] Gerard Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw Hill Book Co., New York, 1983.

[121] Gerard Salton, C. S. Yang, and C. T. Yu. A Theory of Term Importance in Automatic Text Analysis. 26(1):33–44, January–February 1975.

[122] Mark Sanderson and Ian Soboroff. Problems with kendall's tau. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 839–840, New York, NY, USA, 2007. ACM.

[123] Mark Sanderson and Justin Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *SIGIR '05: Proceedings of the 28th annual international ACM*

SIGIR conference on Research and development in information retrieval, pages 162–169, New York, NY, USA, 2005. ACM.

[124] R. Schenkel, A. Theobald, and G. Weikum. Semantic Similarity Search on Semistructured Data with the XXL Search Engine. *Information Retrieval*, 8:521–545, 2005.

[125] Torsten Schlieder and Holger Meuss. Querying and ranking xml documents. *J. Am. Soc. Inf. Sci. Technol.*, 53(6):489–503, 2002.

[126] B. Sigurbjörnsson, J. Kamps, and M. Rijke. An element based approach to XML Retrieval. In *Proceedings of the 2nd workshop of the initiative for the evaluation of XML retrieval (INEX)*, pages 19–26, 2003.

[127] B. Sigurbjörnsson and A. Trotman. Queries: INEX 2003 working group report. In *Proceedings of the 2nd workshop of the initiative for the evaluation of XML retrieval (INEX)*, pages 175–178, 2003.

[128] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Research and Development in Information Retrieval, ACM SIGIR '96*, pages 21–29, 1996.

[129] A. Singhal, C. Buckley, and M. Mitra. Pivoted Document Length Normalization. In H. Frei, D.K. Harman, P. Schauble, and R. Wilkinson, editors, *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29. ACM Press, 1996.

[130] Amit Singhal. *Term Weighting Revisited*. PhD thesis, Cornell University, 1996.

[131] Ian Soboroff, Charles Nicholas, and Patrick Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 66–73, 2001.

[132] K. Sparck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, March 1972.

[133] K. Sparck-Jones. Index Term Weighting. *Information Storage and Retrieval*, 9(11):619–633, November 1973.

[134] K. Sparck Jones and C. van Rijsbergen. Report on the Need for and Provision of an “Ideal” Information Retrieval Test Collection. British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.

[135] Hiroki Tanioka. A method of preferential unification of plural retrieved elements for xml retrieval task. In Fuhr et al. [35], pages 45–56.

[136] X. Tannier. From natural language to NEXI, an interface for INEX 2005 queries. In *Pre-Proceedings of the 4th workshop of the initiative for the evaluation of XML retrieval (INEX)*, pages 289–303, 2005.

[137] A. Trotman, N. Pharo, and D. Jenkinson. Can we at least agree on something? In *SIGIR ’07: Proceedings of the SIGIR 2007 Workshop on Focused Retrieval*, pages 49–56, 2007. http://www.cs.otago.ac.nz/sigirfocus/paper_1.pdf.

[138] A. Trotman and B. Sigurbjörnsson. Narrowed Extended XPath i (NEXI). In *Proceedings of the 3rd workshop of the initiative for the evaluation of XML retrieval (INEX)*, pages 16–40, 2004.

[139] A. Trotman and B. Sigurbjörnsson. NEXI, Now and Next. In *Proceedings of the 3rd workshop of the initiative for the evaluation of XML retrieval (INEX)*, pages 41–53, 2004.

[140] Delphine Verbyst and Philippe Mulhem. Using Collectionlinks and Documents as Context for INEX 2008. In Geva et al. [38], pages 87–96.

[141] Ellen M. Voorhees and Chris Buckley. The effect of topic set size on retrieval experiment error. In *SIGIR ’02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 316–323, New York, NY, USA, 2002. ACM.

[142] A. Vries, G. Kazai, and M. Lalmas. Tolerance to irrelevance: A user-effort oriented evaluation of retrieval systems without predefined retrieval unit. In *Proceedings of RIAO*

(Recherche d'Information Assistée par Ordinateur (Computer Assisted Information Retrieval)), 2004.

[143] A. P. Vries, G. Kazai, and M. Lalmas. Evaluation metrics 2004. In *Pre-Proceedings of the third workshop of the initiative for the evaluation of XML retrieval (INEX)*, pages 249–250, 2004.

[144] W3C. XPath-XML Path Language(XPath) Version 1.0. http://www.w3.org/TR/xpath.

[145] Songlin Wang, Feng Liang, and Jianwu Yang. PKU at INEX 2010 XML Mining Track. In *INEX 2010 Workshop Pre-Proceedings*, pages 321–334, the Netherlands, 2010.

[146] David H. D. Warren and Fernando C. N. Pereira. An efficient easily adaptable system for interpreting natural language queries. *Comput. Linguist.*, 8(3-4):110–122, 1982.

[147] William Webber, Alistair Moffat, and Justin Zobel. Statistical power in retrieval experimentation. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 571–580, New York, NY, USA, 2008. ACM.

[148] William Webber, Alistair Moffat, and Justin Zobel. The Effect of Pooling and Evaluation Depth on Metric Stability. In Tetsuya Sakai, Mark Sanderson, and William Webber, editors, *Proc. of the Third International Workshop on EValuating Information Access (EVIA 2010), NTCIR 8*, pages 7–15, 2010. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings8/EVIA/03-EVIA2010-WebberW.pdf.

[149] Felix Weigel, Klaus U. Schulz, and Holger Meuss. Ranked Retrieval of Structured Documents with the S-Term Vector Space Model.". In *INEX'04*, pages 238–252, 2004.

[150] Jennifer Widom. Data Management for XML. `http://www-db.stanford.edu/$\sim$widom/xml-whitepaper.html`, 1999.

[151] S. K. M. Wong and Y. Y. Yao. On modeling information retrieval with probabilistic inference. *ACM Trans. Inf. Syst.*, 13(1):38–68, 1995.

[152] A. Woodley and S. Geva. NLPX at INEX 2005. In *Pre-proceedings of the 4th workshop of the initiative for the evaluation of XML retrieval (INEX)*, pages 274–288, 2005.

[153] A. Woodley and S. Geva. XCG Overlap at INEX 2004. In *Pre-proceedings of the 4th workshop of the initiative for the evaluation of XML retrieval (INEX)*, pages 25–39, 2005.

[154] A. Woodley and S. Geva. XCG Overlap at INEX 2006. In *Proceedings of the 5th workshop of the initiative for the Evaluation of XML retrieval (INEX)*, pages 302–311, 2007.

[155] Emine Yilmaz and Javed A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 102–111, New York, NY, USA, 2006. ACM.

[156] Emine Yilmaz, Javed A. Aslam, and Stephen Robertson. A new rank correlation coefficient for information retrieval. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 587–594, New York, NY, USA, 2008. ACM.

[157] C.T. Yu, C. Buckley, K. Lam, and G. Salton. A generalized term dependence model in information retrieval. *Information Technology: Research and Development*, 2(4):129–154, October 1983.

[158] Justin Zobel. How reliable are the results of large-scale information retrieval experiments? In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314, New York, NY, USA, 1998. ACM.