

UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of master's thesis by

Dinesh H. Bhirud

and have found that it is complete and satisfactory in all respects,
and that any and all revision required by the final
examining committee have been made

Dr. Carolyn J. Crouch

Name of Faculty Adviser

Signature of Faculty Adviser

Date

GRADUATE SCHOOL

Focused Retrieval Using Upper Bound

Methodology

A thesis

submitted to the faculty of the graduate school

of the University of Minnesota

by

Dinesh H. Bhirud

In partial fulfillment of the requirements

for the degree of

Master of Science

August, 2009.

Department of Computer Science

University of Minnesota, Duluth

Duluth, MN 55812

USA

Acknowledgements

Many people have contributed towards the successful completion of this thesis. I would like to take this opportunity to thank them all.

I would like to thank Dr. Carolyn Crouch for letting me work on this thesis and providing advice and guidance throughout the period of two years. I would also like to thank Dr. Donald Crouch for his suggestions and feedback.

I would also take this opportunity to thank the faculty and staff of the Department of Computer Science at UMD, especially Lori Lucia, Linda Meek and Jim Luttinen for their continuous help and support.

I also thank Salil Bapat, Darshan Paranjape and Aditya Mone for sharing their experience and helping me out at various aspects of this thesis.

And last, but not the least I would like to thank my family and friends for all the unconditional love and support extended to me during the course of these two years. A special thanks to all my friends in Duluth for their help, support and guidance through the course of last two years. Thank you all for your blessings and good wishes.

Abstract

Information Retrieval is the science of retrieving useful information from large collections of documents. With the World Wide Web acting as an ever growing source of information, the task of locating the precise content that the user is looking for has become increasingly complicated. XML is one of the most popular forms of textual data representation on the web. To provide the user with relevant and precise information in response to a query one must consider the large amount of textual data represented as XML. Information retrieval from XML documents involves retrieving XML elements at various levels of granularity, instead of entire document retrieval as in case of traditional information retrieval.

The aim of this thesis is to explore various strategies for improving the performance of the INEX 2008 Ad Hoc Focused retrieval task. The Focused task requires the system to retrieve a rank ordered list of non-overlapping elements. In this thesis, we discuss FLEX, a dynamic and flexible element retrieval system, which is capable of retrieving XML elements at various levels of granularity at runtime. We have used 2 techniques for Focused. This thesis presents the experiments using the Upper Bound method, wherein we know only the upper bound of the number of elements retrieved. This thesis also explores the merits of combining the correlation score of an element with that of its enclosing article to decide the ranking of the element. This approach to ranking of elements has produced results better than the top ranked participant at INEX 2008.

TABLE OF CONTENTS

List OF Tablesv

LIST OF FIGURES.....vii

1. Introduction 1

2. Overview3

2.1 INEX 3

2.2 2008 Retrieval Tasks.....5

2.3 2008 Evaluation Measures6

2.4 Retrieval Engine9

3. Background to Focused Retrieval 11

3.1 Parsing and Indexing..... 11

3.2 Smart Retrieval 13

3.3 Flexible Element Retrieval (Flex)..... 15

3.4 Post-processing on Flex output..... 20

4. Experiments, Results and Analysis22

4.1 Basics (Tag Sets and Term Weighting Parameters) 22

4.2 Focused Retrieval Experiments 26

4.3 Focusing strategies..... 28

4.4 Experiments 29

4.5	Conclusions	51
5.	Future Work	54
	References.....	56

LIST OF TABLES

Table 1 : <i>tags-to-index</i> – Terminal node element tags – Tag set 1	23
Table 2 : <i>tags to keep</i> – Tags considered for retrieval – Tag set 1.....	23
Table 3 : <i>tags-to-index</i> – Terminal node element tags – Tag Set 2.....	24
Table 4 : <i>tags to keep</i> – Tags considered for retrieval – Tag set 2.....	25
Table 5 : Slope and Pivot values used for INEX 2008 experiments.....	26
Table 6 : iP[0.01] for 2007 collection – Section strategy – Tag set 1.....	30
Table 7 : iP[0.01] for 2007 collection – Section strategy – Tag set 2.....	31
Table 8 : iP[0.01] for 2008 collection – Section strategy – Tag set 1.....	31
Table 9 : iP[0.01] for 2008 collection – Section strategy – Tag set 2.....	32
Table 10 : iP[0.01] for 2007 collection – Child strategy – Tag set 1	33
Table 11 : iP[0.01] for 2007 collection – Child strategy – Tag set 2.....	33
Table 12 : iP[0.01] for 2008 collection – Child strategy – Tag set 1	34
Table 13 : iP[0.01] for 2008 collection – Child strategy – Tag set 2.....	34
Table 14 : iP[0.01] for 2007 collection – Correlation score strategy – Tag set 1.....	35
Table 15 : iP[0.01] for 2007 collection – Correlation score strategy – Tag set 2.....	36
Table 16 : iP[0.01] for 2008 collection – Correlation score strategy – Tag set 1.....	36
Table 17 : iP[0.01] for 2008 collection – Correlation score strategy – Tag set 2.....	37
Table 18 : iP[0.01] for 2007 collection – Section strategy rearranged – Tag set 1.....	38
Table 19 : iP[0.01] for 2007 collection – Section strategy rearranged – Tag set 2.....	39
Table 20 : iP[0.01] for 2008 collection – Section strategy rearranged – Tag set 1.....	39
Table 21 : iP[0.01] for 2008 collection – Section strategy rearranged – Tag set 2.....	40

Table 22 : iP[0.01] for 2007 collection – Child strategy rearranged – Tag set 1.....	40
Table 23 : iP[0.01] for 2007 collection – Child strategy rearranged – Tag set 2.....	41
Table 24 : iP[0.01] for 2008 collection – Child strategy rearranged – Tag set 1.....	41
Table 25 : iP[0.01] for 2008 collection – Child strategy rearranged – Tag set 2.....	42
Table 26 : iP[0.01] for 2007 collection – Correlation strategy rearranged – Tag set 1.....	43
Table 27 : iP[0.01] for 2007 collection – Correlation strategy rearranged – Tag set 2.....	43
Table 28 : iP[0.01] for 2008 collection – Correlation strategy rearranged – Tag set 1.....	44
Table 29 : iP[0.01] for 2008 collection – Correlation strategy rearranged – Tag set 2.....	44
Table 30 : iP[0.01] for 2008 collection – Section – Flex chosen documents	46
Table 31 : iP[0.01] for 2008 collection – Child – Flex chosen documents	46
Table 32 : iP[0.01] for 2008 collection – Correlation – Flex chosen documents	47
Table 33 : iP[0.01] for 2008 collection – Section –All-el slope and pivot values	48
Table 34 : iP[0.01] for 2008 collection – Child - All-el slope and pivot values.....	48
Table 35 : iP[0.01] for 2008 collection – Correlation – All-el slope and pivot values	49
Table 36 : iP[0.01] for 2008 collection – Section Rearranged - All-el slope and pivot....	49
Table 37 : iP[0.01] for 2008 collection – Child Rearranged - All-el slope and pivot.....	50
Table 38 : iP[0.01] for 2008 collection–Correlation Rearranged–All-el slope and pivot.	50
Table 39 : iP[0.01] for 2007 Focused, 2007 Evaluation, without XPath expansion.....	58
Table 40 : iP[0.01] for 2007 Focused retrieval, 2007 Evaluation, with XPath expansion	59
Table 41 : 2007 Focused retrieval – Child Strategy – Tag set 1.....	60

LIST OF FIGURES

Figure 1 : Formula for calculating $iP[x]$	8
Figure 2 : Formula for calculating AiP	8
Figure 3 : Formula for calculating $MAiP$	9
Figure 4 : A high level block diagram of the focused retrieval system used.....	11
Figure 5 : Formula for Lnu term weighting of documents	14
Figure 6 : Formula for ltu weighting of queries.....	15
Figure 7 : Example of a doctree.....	16
Figure 8 : Example of a seeded tree.....	17
Figure 9 : Production of seed subsets.....	19
Figure 10 : Comparison - non-rearranged Vs rearranged output - section strategy....	52
Figure 11 : Comparison - non-rearranged Vs rearranged output - child strategy.	52
Figure 12 : Comparison - non-rearranged Vs rearranged output - correlation strategy.	53

1. Introduction

Information Retrieval is the science of retrieving useful information from large collections of documents. With the World Wide Web acting as an ever growing source of information, the task of locating the precise content that the user is looking for has become increasingly complicated. Search engines like Google, Yahoo and many more are constantly working in the field of Information Retrieval to improve the experience of the user when he goes online to search for information.

XML is one of the most popular forms of textual data representation on the web. To provide the user with relevant and precise information in response to a query one must consider the large amount of textual data represented as XML. The focus of this research is to extend traditional information retrieval, which concentrates on retrieving entire documents, to the retrieval of textual elements represented as XML. The research endeavors to improve XML retrieval systems by reducing the granularity of search results from entire documents to individual elements such as paragraphs, sections, etc as well as enabling search within XML documents that do not strictly follow a DTD (Document Type Definition). We call these XML documents *semi-structured* documents.

This thesis describes the work performed within our research group at the University Of Minnesota Duluth. The group is a participant in the INEX (Initiative for the Evaluation of XML Retrieval) competition. This competition is sponsored by INEX to enhance the state of the art of XML retrieval. INEX provides all the participants a

XML document collection (in 2008 a subset of Wikipedia), a set of queries, and metrics for evaluating the results returned. The performance of the participant systems is compared based on these metrics. As the Wikipedia document collection does not adhere strictly to a DTD, this research concentrates on retrieval from semi-structured documents.

The University Of Minnesota Duluth is a participant in the Ad Hoc retrieval track for INEX 2008. This thesis deals with this track in general, and with the Focused Retrieval Task in particular. The tasks, along with the document collection, the evaluation measures adopted and the basics of the retrieval system are described in Chapter 2. Chapter 3 describes Flexible retrieval system, (i.e., efficient element retrieval). Chapter 4 details our experiments in Focused retrieval and presents and analyzes the results of these experiments. Conclusions and suggestions for future work are discussed in Chapter 5.

2. Overview

This chapter provides a description of the INEX 2008 Ad hoc track along with the evaluation measures used to assess the performance of the system. It also gives a brief description of the Vector Space Model, which lies at the core of the basic retrieval system i.e., Smart [14], that is used in this research.

2.1 INEX

INEX [6] is an initiative that facilitates development of effective information retrieval strategies for XML documents through tracks like the Ad-hoc, Book, Efficiency, Interactive etc. The main goal of INEX is to provide large test collections, uniform evaluation measures and a forum for its participants to compare their results and discuss their strategies [6]. Our IR research group at UMD participates in the Ad Hoc track. Our system uses the test collection (the 2008 Wikipedia XML collection) and queries provided by INEX and retrieves relevant XML elements as results. These results are then evaluated against the relevance assessments provided by INEX to generate scores based on the evaluation metrics defined by INEX. The retrieval systems of the participants are compared based on these scores.

Document Collection

Prior to 2006, INEX utilized the IEEE document collection, which strictly conformed to a DTD. Our research group had successfully produced competitive results for

retrieval on this collection, using our method of dynamic or flexible element retrieval [4].

In 2007, INEX moved to the Wikipedia XML document collection. We call the documents semi-structured because whereas they do contain some degree of uniform structure as XML documents, they do not strictly conform to a DTD. Also, the documents include text that falls within none of the tags, i.e., untagged text.

The Wikipedia document collection currently use is approximately 5.8GB in size and contains 659, 338 documents. The average number of XML nodes per article is 161 and the average depth of an element is 6.72 [14]. Our participation in INEX 2006, 2007 and 2008 is based on experiments performed on this document collection.

Starting in 2009, INEX has provided a new Wikipedia text collection, which is almost 10 times the size of the previous collection at 50.7GB. It contains 2,666,190 articles [5].

Topic Collection

The topics are created by the participants of INEX. Each topic has mandatory node called *title*. This node contains a short Content-Only (CO) query, which we use for retrieval. This query does not take into account any structural information nor does it provide any hint into the structure of the text that might be relevant to it. Each topic can have an optional field called *castitle*, which provides a Content-And-Structure (CAS) query.

The 2008 topic set contains 135 topics have been submitted with topic numbers ranging from 544 to 678 [6].

Relevance Assessments

Relevance assessments are produced by manually assessing a subset of the document collection for all the topics. Every participating group is given a set of topics to assess; generally these are the topics that the group submitted. The relevance assessments produced from this exercise form the basis for evaluating the results generated by the systems of each of the participating groups.

For INEX 2008, the GPXRai assessment system has been provided. This tool allows us to mark text as relevant to a particular query and to specify the the Best Entry Point for each document [6]. To enable the evaluation of arbitrary passages retrieved, INEX converts this marked text into File Offset and Length format before it is used for evaluation of the results.

2.2 2008 Retrieval Tasks

The 2008 Ad Hoc retrieval track includes the following tasks:

Focused Task

For this task “a ranked list of non-overlapping elements must be returned” [6]. The challenge in this task is to find non-overlapping elements. In a XML document, if an element is found relevant, then naturally all its parents will be found relevant to some extent. The main goal of this task is to find out which of all these elements in the tree

is the most relevant to the query and return only one of them. We call this process *overlap removal*, and our technique for achieving this is explained in more detail in Chapter 3.

Relevant in Context Task

For this task “non-overlapping results grouped by document” must be returned [6]. Users may consider documents the most natural retrieval unit. However, the relevant content may be spread across various elements in the documents, and the entire document may not be relevant. Hence, the goal of this task to provide the user with a rank ordered list of focused elements within each document sorted on relevant documents i.e., in document order. For details, see [12]

Best in Context Task

For this task “a single starting point for each article” must be returned[6]. This task must provide the user with a ranked list of articles along with the best entry point for each article. For details, see [19].

2.3 2008 Evaluation Measures

For evaluating the Ad hoc track, the organizers of INEX have made the assumption that “the amount of relevant information retrieved is measured in terms of the length of relevant text retrieved” [7]. This assumption has been continued in the INEX 2008 evaluation. Hence, the evaluation methods in INEX 2008 support document elements represented in the File Offset and Length (FOL) format, as opposed to the XPath

used earlier. Although, XPath continues to be the primary method of identifying document parts (elements), they are converted into FOL format before evaluation.

Focused Task Evaluation

To assess the quality of the ranked order list of elements produced by Focused retrieval task, interpolated precision at different recall levels is used as the measure.

They are extended to form the measure, interpolated precision at various recall levels [6]. INEX is most interested in the elements returned at high ranks and hence interpolated precision at 1% recall is chosen as the official measure of this task.

Interpolated precision at recall level x [7] is calculated using the formula shown in Figure 1.

However, for assessing the overall performance of focused retrieval over the entire list, INEX evaluation also provides average interpolated precision measure [7], which is calculated by averaging interpolated precisions at 101 standard recall levels. The formula for average interpolated precision (AiP) is shown in Figure 2.

Performance across a set of topics is measured by calculating the mean of the AiP values obtained for each individual topic, resulting in a mean average interpolated precision (MAiP) [7]. The formula for calculating MAiP is shown in Figure 3.

$$iP[x] = \begin{cases} \max_{1 \leq r \leq |L_q|} (P[r] \wedge R[r]) & \text{if } x \leq R[|L_q|] \\ 0 & \text{otherwise} \end{cases}$$

where L_q is the ranked list of elements

and $|L_q|$ is the length of this ranked list, which is 1500 for INEX competition

$P[r]$ is the precision at rank r

$R[r]$ is the recall at rank r

$R[|L_q|]$ is the recall over all the documents retrieved

Figure 1 : Formula for calculating $iP[x]$

$$AiP = \frac{1}{101} \sum_{x=0.0,0.01,\dots,1.0} iP[x]$$

where $iP[x]$ is interpolated precision at recall x .

Figure 2 : Formula for calculating AiP

Assuming there are n topics

$$MAiP = \frac{1}{n} \sum_t AiP(t)$$

Figure 3 : Formula for calculating MAiP

2.4 Retrieval Engine

Our approach to XML retrieval is based on the Vector Space Model [15]. Our approach to element retrieval uses the Smart retrieval engine [14] to retrieve XML elements.

Vector Space Model

The Vector Space Model [15] is the basic model used for retrieval in this research. In this model, both the document and the query are represented as n-dimensional vectors, whose components are all the terms (word types) occurring in them. A similarity measure such as cosine is used to compute the correlation between the vectors [15].

Smart Retrieval Engine

Smart is an experimental retrieval system based on the Vector Space Model [3]. It has an automatic indexing component that constructs a vector representation of each

document and the query. Retrieval produces a rank ordered list of documents that seem to correlate with the query based on a specified measure.

We use *Lnu-ltu* term weighted document and query vectors which require the use of two collection specific parameters called slope and pivot [18]. The weighting attempts to ensure that length disparities between element vectors do not bias results unfairly in favor of the longer elements. The pivot is calculated as the average number of unique terms in a document (computed across the entire collection), and the value of slope is found experimentally.

Smart includes an evaluation component which evaluates the results generated in terms of precision and recall metric. (However, we use the evaluation tool provided by INEX, which gives results in terms of the official INEX evaluation measures.) We use Smart as the basic retrieval engine during this research. But, for element retrieval we use our method of dynamic element or flexible element retrieval called Flex [8]. A detailed explanation of Flex and our techniques for XML element retrieval is given in Chapter 3.

3. Background to Focused Retrieval

This chapter describes the procedures and algorithms that retrieve focused elements and arrange them in orders suitable to the different tasks. A detailed discussion of Flex, the component that achieves dynamic or flexible element retrieval, is also included. The particular techniques and details of focused retrieval, along with the experiments performed are presented in Chapter 4. This chapter describes all the aspect and components utilized in the production of focused elements.

The overview of the approach is seen in Figure 4.

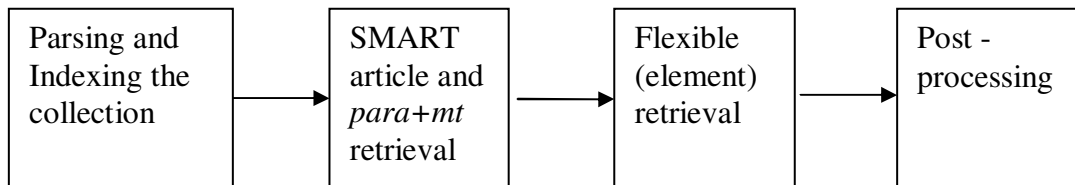


Figure 4 : A High Level Block Diagram of the Focused Retrieval System

3.1 Parsing and Indexing

This section describes the pre-retrieval tasks (i.e., tasks that prepare the collection for retrieval).

Parsing

Parsing removes the unneeded tags from the XML documents while retaining the text within them. Tags that are considered important for retrieval purposes (i.e., structural tags) are retained.

Parsing takes 2 inputs : the XML document collection itself, and a configuration file which specifies which tags to be retain.

Parsing produces both an article and a *para+mt* parse. The *article parse* creates a single element per article [2]. To create this parse, the text enclosed within document tags is retained and all the other tags occurring within the article tag are removed. The *para+mt parse* identifies all the terminal or leaf node elements of an XML document. It retains the text contained in these elements, discarding all other tags occurring within them. This parse also identifies the untagged text in the XML document and marks it with a *mt* (magic text) tag. Untagged text may occur at any level in the document tree (para, section, body). All such text occurring at the same level of the tree is combined into one *mt* element (which becomes a child of the element within which it is enclosed). See [1, 10] for details. However, as it is not a valid tag in the collection, an *mt* tags is never returned as result. But they must be parsed and indexed so that the text contained in them can be included as part of their parent element when the parent is constructed.

Another process taking place at this stage includes the building of *doctrees* or document schemas which are infact preorder traversal of the documents themselves. The tags included in the schemas are found in the configuration file under *tags to keep*. These *doctrees* are later used for flexible element retrieval.

To summarize the parsing stage, we identify and isolate each retrievable unit/element from the collection, so that they can be treated as individual elements (or documents) for retrieval purposes.

Indexing

Taking the elements contained in the parse as input, indexing creates element and the query vectors. We use the Smart [14, 3] retrieval system for this purpose. Smart generates *nnn vectors* or term frequency vectors where each vector represents an element

Thus, in the indexing stage we generate *article* and *paras+mt* and *query* indices. The article index is used for article retrieval. The *paras+mt* index is used for terminal node retrieval which forms the basis of the flexible retrieval step. (This step uses the same configuration file as that of the parsing stage to identify leaf node elements.)

The parsed collection now exists in vector form.

3.2 Smart Retrieval

Smart is also used for retrieval. To avoid a bias towards elements of bigger size (i.e., longer vectors), we use the *Lnu-ltu* weighting scheme [8]. Document vectors are *Lnu* - weighted and query vectors are *ltu* - weighted. The formulas for *Lnu* and *ltu* weights are given in Figure 5 and Figure 6. The *Lnu-ltu* weighting scheme uses the

parameters slope and pivot. Details of the Lnu-ltu weighting scheme can be found in [18, 17]

$$\frac{\frac{1 + \log(tf)}{1 + \log(\text{average } tf \text{ in text})}}{(1 - slope) + slope \times \frac{\# \text{ unique terms in text}}{pivot}}$$

Where *tf* is the term frequency obtained from the nnn element vectors.

average tf in text is the average of the *tf* of all the terms in this element

#unique terms in text is the number of distinct terms in this element

slope and *pivot* are the empirically determined parameters

Figure 5 : Formula for Lnu term weighting of documents

The similarity score between a document and a query is found by taking the inner product of the *Lnu* weighted element vector with the *ltu* weighted query vector, and a ranked list is produced based on this score.

The final output of Smart retrieval is a rank ordered list of articles and a rank ordered list of leaf node (*para+mt*) elements for each query. (Note that we retrieve a large number of leaf nodes (125,000) to ensure that our list of elements is complete, i.e., that it contains every element in the collection that has a non-zero correlation with respect to the query.)

$$\frac{(1 + \log(tf)) \times \left(\log \frac{N+1}{df} \right)}{(1 - slope) + slope \times \frac{\#unique\ terms\ in\ text}{pivot}}$$

Where tf is the term frequency which is obtained from the nnn vectors.

N is the total number of elements in the collection

df is the document frequency of the term

$\#unique\ terms\ in\ text$ is the number of terms in the element

$slope$ and $pivot$ are the empirically determined parameters

Figure 6 : Formula for Itu Weighting of Queries

3.3 Flexible Element Retrieval (Flex)

This section describes Flex, the component which implements flexible retrieval.

Flex dynamically generates the document trees *of interest* along with correlation scores for all the elements with respect to the query.

Seeding of doctrees.

This is the first step in flexible retrieval. In order to build a parent (or a higher level element) vector, doctrees need to be seeded or populated with the text or content of the lead node elements.

Seeding takes the following inputs:

- 1) doctrees or the document schemas of all documents. (See Figure 7 for an example of a doctree),
- 2) the rank-ordered list of *para+mt* retrieved by Smart,
- 3) the docid – docpath mapping file from the *para+mt* index, which maps the Smart identifier of each element to its XPath.

/article[1]	2	0			
/article[1]/name[1]	0	1			
/article[1]/body[1]	7	0			
/article[1]/body[1]/mt[1]		0	1		
/article[1]/body[1]/emph3[1]		0	1		
/article[1]/body[1]/section[1]	4	1			
/article[1]/body[1]/section[1]/title[1]			0	1	
/article[1]/body[1]/section[1]/mt[1]		0	1		
/article[1]/body[1]/section[1]/emph3[1]			0	1	
/article[1]/body[1]/section[1]/p[1]		0	0		
/article[1]/body[1]/section[2]	3	1			
/article[1]/body[1]/section[2]/title[1]			0	1	
/article[1]/body[1]/section[2]/mt[1]		0	1		
/article[1]/body[1]/section[2]/p[1]		0	0		
/article[1]/body[1]/section[3]	2	1			
/article[1]/body[1]/section[3]/title[1]			0	1	
/article[1]/body[1]/section[3]/normallist[1]			0	0	
/article[1]/body[1]/section[4]	2	1			
/article[1]/body[1]/section[4]/title[1]			0	1	
/article[1]/body[1]/section[4]/normallist[1]			0	0	
/article[1]/body[1]/section[5]	2	0			
/article[1]/body[1]/section[5]/title[1]			0	1	
/article[1]/body[1]/section[5]/normallist[1]			0	0	

Figure 7 : Example of a Doctree

The output of seeding is a set of doctrees per query, with the correlation scores for their terminal node elements populated from the *para+mt* Smart retrieval. Also, each terminal node in the doctrees is populated with its Smart id, i.e., a pointer to its vector. An example of a seeded doctree is shown in Figure 8.

```
1392796/article[1] 2 0
1392796/article[1]/name[1] 0 1 4372021 9.5861
1392796/article[1]/body[1] 7 0
1392796/article[1]/body[1]/mt[1] 0 1 4372022 0
1392796/article[1]/body[1]/emph3[1] 0 1 4372023 9.5020
1392796/article[1]/body[1]/section[1] 4 1
1392796/article[1]/body[1]/section[1]/title[1] 0 1 -1 0
1392796/article[1]/body[1]/section[1]/mt[1] 0 1 4372024 0
1392796/article[1]/body[1]/section[1]/emph3[1] 0 1 4372025 9.5020
1392796/article[1]/body[1]/section[1]/p[1] 0 0 4372026 11.1950
1392796/article[1]/body[1]/section[2] 3 1
1392796/article[1]/body[1]/section[2]/title[1] 0 1 -1 0
1392796/article[1]/body[1]/section[2]/mt[1] 0 1 4372027 0
1392796/article[1]/body[1]/section[2]/p[1] 0 0 4372028 8.9523
1392796/article[1]/body[1]/section[3] 2 1
1392796/article[1]/body[1]/section[3]/title[1] 0 1 -1 0
1392796/article[1]/body[1]/section[3]/normallist[1] 0 0 4372029 0
1392796/article[1]/body[1]/section[4] 2 1
1392796/article[1]/body[1]/section[4]/title[1] 0 1 -1 0
1392796/article[1]/body[1]/section[4]/normallist[1] 0 0 4372030 0
1392796/article[1]/body[1]/section[5] 2 0
1392796/article[1]/body[1]/section[5]/title[1] 0 1 -1 0
1392796/article[1]/body[1]/section[5]/normallist[1] 0 0 4407079 10.0645
```

Figure 8 : Example of a Seeded Tree.

In Figure 8, the highlighted text is the Smart id and the correlation score of the terminal nodes.

As mentioned in Section 3.2, the ranked ordered list of *para+mt* is long enough to accommodate all the elements having non-zero correlations. In the past, we used this list to populate the terminal node elements of the doctrees. We now use the docid-docpath mapping file to populate the contents of the terminal node elements. Flex calculates the correlation scores for all the terminal node elements, and hence the score from the *para+mt* retrieval is no longer needed.

Producing the Seed Subsets

In INEX 2008, we combined article retrieval (by Smart) with element retrieval (by Flex) to produce the seed subsets used for focused retrieval. Seed subsets are sets of seeded doctrees i.e., doctrees with each terminal node populated with its content

representing the subset of documents retrieved by the query. The seed subsets are generated by filtering (i.e., selecting) of seeded doctrees based on the article retrieval. This subset contains complete document schemas with all their terminal nodes populated. This subset forms the input to flexible retrieval. Figure 9 shows the process of generation of doctrees and the seed subsets.

Flexible retrieval (Flex)

Flex [8] is the component of the retrieval system that generates elements dynamically. Given all the vectors of their children Flex can generate vectors for the parent nodes, even if they are not indexed.

Flex takes following inputs:

For each query

- 1) seed subset

A set of doctrees with the leaf elements populated

- 2) *para+mt* element vectors

Flex needs these because the seed subsets contain only the pointers to the element vectors. The vectors themselves are contained in the list of *paras+mt* elements.

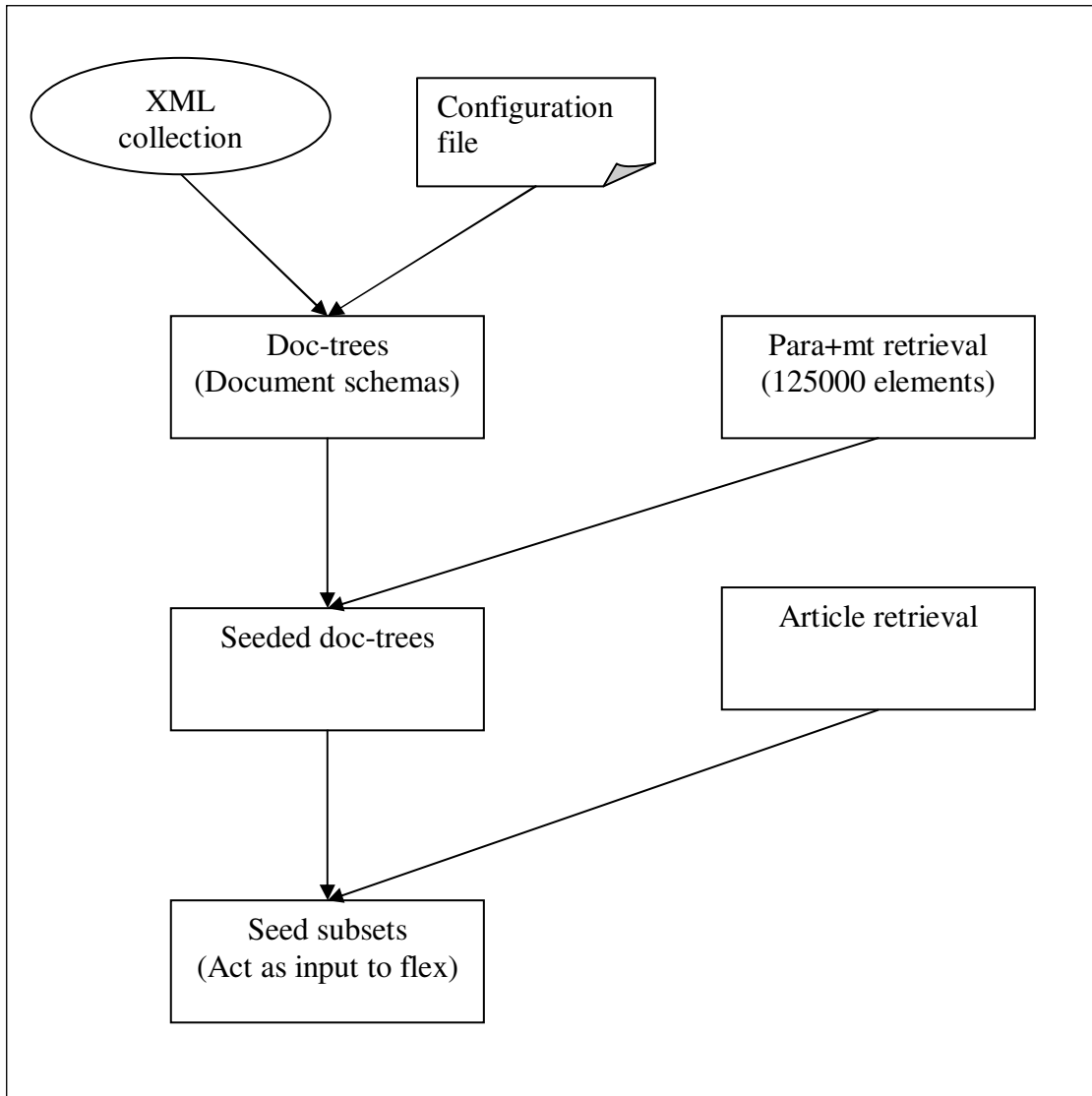


Figure 9 : Production of Seed Subsets

- 3) query index
- 4) slope and pivot values

Flex needs these to properly weight the element vectors produced.

Using the terminal node vectors as input, Flex recursively builds the vectors for their parents according to the document schema. Slope and pivot values are used for *Lnu* weighting. The process terminates when Flex has generated complete document trees which are fully populated (i.e., with a correlation score for every element in it.) Flex then sorts the elements to produce a rank ordered list.

3.4 Post-processing on Flex output

Once a ranked order list of elements is generated, the following steps are performed:

Removal of *mt* Nodes.

Recall that, *mt* (magic text) nodes have been introduced so that the the untagged text contained within an element is not lost [1, 10]. However, since *mt* is not a legal tag which can be evaluated, all the XPath's containing *mt* must be removed from the rank ordered list generated by Flex.

Conversion to Task Specific Outputs

The Ad Hoc retrieval track has three tasks. The rank ordered list generated in the previous step must be filtered and rearranged to suit the requirements of the Focused task. The Focused task expects a ranked list *non-overlapping* of elements. Two elements are said to be overlapping if a portion of their text is shared by them. We remove this overlap by using three different strategies (i.e., the *child*, *section* and *correlation strategies*.) These strategies along with their results are discussed in Chapter 4.

Evaluation

This is the final step in the retrieval process, wherein the focused output is evaluated using the evaluation tool provided by INEX.

- 1) The lists generated in Section 3.2 are converted into XML format, so that they are in a format compatible to the evaluation tool.
- 2) The XPathS of the returned elements are converted into the File Offset and Length (FOL) format using a tool provided by INEX.
- 3) The FOLs generated in the previous step are then given to the evaluation tool, which calculates the scores using the corresponding metric for each task. This tool uses *qrels* generated from the manual relevance assessments as the basis for evaluating the results submitted to it.

4. Experiments, Results and Analysis

As a part of this thesis, we have performed experiments to determine strategies and mechanisms to improve the results of the Focused retrieval task. Experiments are done on the 2007 and 2008 document and query collections. This chapter details these experiments and provides the results produced, along with an analysis of these results.

4.1 Basics (Tag Sets and Term Weighting Parameters)

This section describes the background for the experiments done for improving the focused retrieval task results. It includes a discussion set of the tags and slope and pivot values, which affect all the subsequent retrieval experiments.

Tags Sets

We divide the tags that are considered important for retrieval into two groups namely:

- 1) *tags-to-index* – These tags form the set of all terminal nodes. i.e., the elements with these tags do not have any child nodes. Collectively, all these tags are referred to as *paragraphs*; they are found in the *para+mt* parse. Magic text is also considered a terminal node. Hence, an indexing of these tags is called the *para+mt* index.
- 2) *tags-to-keep* – These are all the tags that we consider important for retrieval purposes, and hence these tags identify only the elements that are returned as a result of flexible retrieval. *tags-to-index* are always a subset of *tags-to-keep*.

The tags sets used in our early 2008 experiments are listed in Tables 1 and 2. We call this tag set *Tag Set 1*.

Table 1 : *tags-to-index* – Terminal Node Element Tags – Tag set 1

Tag Name	Tag Representation
Paragraph	<p> ... </p>
Name	<name> ... </name>
Figure	<figure> ... </figure>
Emphasis	<emph3> ... </emph3>
Magic Text	<mt> ... </mt>

Table 2 : tags to keep – Tags considered For Retrieval – Tag set 1

Tag Name	Tag Representation
Article	<article> ... </article>
Section	<section> ... </section>
Body	<body>... </body>
Paragraph	<p> ... </p>
Normal-list	<normallist> ... </normallist>
Ordered-list	 ...
Unordered-list	 ...
Number-list	<numberlist> ... </numberlist>
Definition-list	<definitionlist> ... </definitionlist>
Figure	<figure> ...</figure>
Name	<name> ... </name>
Title	<title> ... </title>
Magic Text	<mt> ... </mt>

In the *tags-to-index* shown in Table 1, it is observed that all the list elements (like *normal-list*, *ordered list*, etc.), along with the table element are not considered as terminal node elements. The side effect of this is that the list elements and the table element would be assigned a non-zero correlation score if and only if they contain one of the tags in *tags-to-index* as a child. That is, such a node would be retrieved only if it had a child node which was identified as terminal node element. Otherwise, text contained within the *list tags* and the *table tag* is folded into its parent tag. To avoid this condition, a new Tag set is identified. Tables 3 and 4 shows this new tag set.

Table 3 : *tags-to-index* – Terminal Node Element Tags – Tag Set 2

Tag Name	Tag Representation
Paragraph	<p> ... </p>
Name	<name> ... </name>
Figure	<figure> ... </figure>
Emphasis	<emph3> ... </emph3>
Magic Text	<mt> ... </mt>
Normal-list	<normallist> ... </normallist>
Ordered-list	 ...
Unordered-list	 ...
Number-list	<numberlist> ... </numberlist>
Definition-list	<definitionlist> ... </definitionlist>
Table	<table> ... </table>

Table 4 : *tags to keep* – Tags Considered For Retrieval – Tag set 2

Tag Name	Tag Representation
Article	<article> ... </article>
Section	<section> ... </section>
Body	<body>... </body>
Paragraph	<p> ... </p>
Emphasis	<emph3> ... </emph3>
Normal-list	<normallist> ... </normallist>
Ordered-list	 ...
Unordered-list	 ...
Number-list	<numberlist> ... </numberlist>
Definition-list	<definitionlist> ... </definitionlist>
Figure	<figure> ...</figure>
Name	<name> ... </name>
Title	<title> ... </title>
Magic Text	<mt> ... </mt>
Table	<table> ... </table>

Experiments are performed using both these tag sets and results presented for both in the later sections.

Slope and Pivot Values

Since the document collection for the INEX 2008 tasks is the same as that of INEX 2007 competition [6], slope and pivot values need not be changed. Table 5 shows the slope and pivot values used throughout our experiments[2].

Table 5 : Slope and Pivot values used for INEX 2008 experiments

Retrieval Type	Slope	Pivot
Article	0.04	120
Paragraph	0.12	15
Section	0.12	38
All-element	0.12	38

4.2 Focused Retrieval Experiments

This section describes the experiments performed to improve the focused retrieval task results. Both article and element retrieval are used to produce the final output.

These steps are followed after the parsing and indexing tasks are done:

For each query

- 1) Retrieve 125,000 terminal node element using Smart retrieval (to guarantee the generation of complete trees).
- 2) Seed the doctrees with the elements from this list.
- 3) Perform an article retrieval using Smart. Retrieve n articles.

- 4) Select a seeded doctree for each of the n articles retrieved. This will create a subset of the seeded doctrees.
- 5) Use this subset of seeded doctrees as input to Flex. Using Flex, populate the doctrees. From the elements generated choose the top m elements.
- 6) Remove all the elements with mt tag from this list.
- 7) Focus the remaining list of elements elements (i.e., remove the overlapping elements to generate focused results).
- 8) Convert the list into XML format so that it can be read by the INEX evaluation tool.
- 9) Convert this XML file to File Offset and Length (*FOL*) format.
- 10) Evaluate this FOL file to get a score for the experiment.

The list of m elements retrieved from Flex is reduced by removal of mt tag elements and also by removal of overlapping elements. Hence, the final list generated would contain at most m elements. It is not possible to predict how many elements will be returned for a given value of m . Hence, we call this approach the *Upper Bound* approach. We have performed experiments in which the number of elements returned (m) is known exactly, which is called the *Exact* approach. Details of this method can be found in [13]. All the experiments and results presented in this thesis use the *Upper Bound* approach.

4.3 Focusing strategies

Focusing is the task of removal of overlap from the rank ordered list of elements obtained from Flex. Two elements are said to be overlapping if they have common text.

Following is an example of overlapping elements.

12112/article[1]/body[1]/section[1] ----- (1)

12112/article[1]/body[1]/section[1]/p[1] ----- (2)

As can be seen, the element (2) is a child of element (1), which means that the text contained in element (2) will also be a part of element (1). The focusing strategy decides which one of these two should be chosen as a relevant element. The output of the focusing operation is a list of non-overlapping elements (with overlap removed using one of the strategies discussed below), ranked in order of their correlation score.

Following is a discussion of the three overlap removal strategies that we use.

Section Strategy

In section strategy, given two overlapping XPath's, preference is given to the XPath having a higher correlation score, provided that XPath is not that of a body tag itself. This strategy does not allow entire body or article tags to be selected in lieu of its children.

It is observed that most of the elements returned by this strategy are *sections* and hence we have named this strategy the section strategy.

Child Strategy

This is the same strategy referred to the terminal node strategy in [11]. In this strategy for overlap removal, given two overlapping XPath, the element which is deeper in the document tree is given preference over the other elements i.e., the child node is always given higher preference than its parent node even if the parents' correlation score is higher than that of the child. Hence, this strategy is named the child strategy.

Correlation Strategy

The correlation strategy gives preference to the highest correlating element along a path, i.e., this strategy selects any element which has a higher correlation score in lieu of all other elements along the same path. Hence, most of the elements returned by this strategy are the entire article body elements.

4.4 Experiments

This section describes all the experiments performed using the *Upper Bound* methodology discussed earlier. (For a detailed discussion of the experiments done using the *Exact* method refer to [13]). The evaluations done for all the experiments presented here are performed using the evaluation tool released by INEX in 2008.

The following steps are performed before starting experiments 1, 2 and 3:

- 1) Perform flexible retrieval of m elements from n articles using Flex.
- 2) Remove mt tag elements from this list

Experiment 1: Focused retrieval using Upper Bound and *section* strategy

Results are generated in this experiment by removing overlap from the rank ordered list of elements using the *section* strategy. This experiment was performed on both the INEX 2007 and 2008 collections, using tag sets 1 and 2. Results of this experiments are shown in Tables 6, 7, 8 and 9.

Table 6 : $iP[0.01]$ for 2007 collection – Section strategy – Tag set 1

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.4924	0.4731	0.4726	0.4731	0.4816	0.4806	0.4806	0.4806	0.4806	0.4806	0.4806
50	0.4996	0.4728	0.4709	0.4735	0.4844	0.4810	0.4764	0.4764	0.4764	0.4764	0.4764
100	0.4804	0.4647	0.4564	0.4593	0.4801	0.4779	0.4754	0.4711	0.4711	0.4711	0.4711
150	0.4547	0.4642	0.4551	0.4559	0.4603	0.4628	0.4615	0.4564	0.4520	0.4520	0.4520
200	0.4550	0.4643	0.4673	0.4562	0.4621	0.4625	0.4622	0.4561	0.4560	0.4515	0.4515
250	0.4814	0.4576	0.4669	0.4564	0.4788	0.4814	0.4780	0.4734	0.4706	0.4661	0.4661
500	0.4780	0.4540	0.4649	0.4620	0.4828	0.4776	0.4759	0.4763	0.4734	0.4713	0.4686

Table 7 : iP[0.01] for 2007 collection – Section strategy – Tag set 2

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.4730	0.4686	0.4652	0.4711	0.4708	0.4691	0.4688	0.4688	0.4688	0.4688	0.4688
50	0.4483	0.4525	0.4507	0.4478	0.4514	0.4559	0.4498	0.4498	0.4498	0.4498	0.4498
100	0.4436	0.4501	0.4527	0.4523	0.4526	0.4484	0.4523	0.4503	0.4458	0.4458	0.4458
150	0.4455	0.4479	0.4550	0.4555	0.4548	0.4478	0.4481	0.4519	0.4497	0.4452	0.4452
200	0.4439	0.4435	0.4524	0.4569	0.4560	0.4494	0.4481	0.4499	0.4477	0.4477	0.4432
250	0.4438	0.4439	0.4524	0.4572	0.4555	0.4570	0.4493	0.4466	0.4496	0.4466	0.4420
500	0.4415	0.4405	0.4441	0.4525	0.4533	0.4532	0.4435	0.4461	0.4456	0.4440	0.4450

Table 8 : iP[0.01] for 2008 collection – Section strategy – Tag set 1

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6589	0.6457	0.6451	0.6440	0.6458	0.6455	0.6455	0.6455	0.6455	0.6455	0.6455
50	0.6698	0.6471	0.6580	0.6486	0.6533	0.6521	0.6521	0.6521	0.6521	0.6521	0.6521
100	0.6709	0.6572	0.6443	0.6371	0.6384	0.6450	0.6446	0.6443	0.6443	0.6443	0.6443
150	0.6702	0.6572	0.6487	0.6412	0.6332	0.6372	0.6411	0.6409	0.6410	0.6410	0.6410
200	0.6679	0.6613	0.6518	0.6445	0.6339	0.6313	0.6348	0.6329	0.6327	0.6327	0.6327
250	0.6697	0.6748	0.6555	0.6488	0.6511	0.6357	0.6353	0.6356	0.6355	0.6355	0.6355
500	0.6662	0.6689	0.6544	0.6480	0.6517	0.6269	0.6336	0.6334	0.6317	0.6337	0.6333

Table 9 : $iP[0.01]$ for 2008 collection – Section strategy – Tag set 2

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6494	0.6394	0.6405	0.6417	0.6441	0.6426	0.6426	0.6426	0.6426	0.6426	0.6426
50	0.6463	0.6319	0.6242	0.6260	0.6223	0.6256	0.6233	0.6233	0.6233	0.6233	0.6233
100	0.6444	0.6401	0.6247	0.6196	0.6145	0.6165	0.6171	0.6170	0.6170	0.6170	0.6170
150	0.6429	0.6470	0.6330	0.6198	0.6159	0.6137	0.6150	0.6131	0.6130	0.6131	0.6131
200	0.6434	0.6456	0.6340	0.6200	0.6188	0.6115	0.6131	0.6135	0.6110	0.6108	0.6108
250	0.6512	0.6488	0.6380	0.6323	0.6220	0.6130	0.6134	0.6150	0.6130	0.6127	0.6127
500	0.6479	0.6471	0.6468	0.6346	0.6300	0.6155	0.6126	0.6104	0.6131	0.6107	0.6104

Observations and Analysis regarding the section strategy: For both the collections, the best $iP[0.01]$ score obtained by tag set 1 is better than the best $iP[0.01]$ score obtained by tag set 2. Across the tables, i.e., for a wide range of articles and elements retrieved the scores for tag set 1 are higher than those for tag set 2, for the section strategy. This observation can be attributed to the fact that tag set 2 contains list and table tags as terminal node tags, which are generally quite small in size. Hence, in cases when the terminal node has higher correlation score than the enclosing section, chance of selecting a larger amount of correlating text is better with the use of tag set 1. Also, the best scores occur when a fairly small number of elements is retrieved. The best score of 0.6748 for this strategy using *Tag Set 1* ranks us as Fourth out of 61 participant runs for the Focused retrieval task at INEX 2008.

Experiment 2: Focused retrieval using *child* strategy

Results are generated in this experiment by removing overlap from the rank ordered list of elements using the *child* strategy. This experiment was performed on both the INEX 2007 and 2008 collections, using tag set 1 and tag set 2. Results generated for this experiment are shown in Tables 10, 11, 12 and 13.

Table 10 : iP[0.01] for 2007 collection – Child strategy – Tag set 1

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.4972	0.4734	0.5089	0.5144	0.5383	0.5386	0.5381	0.5381	0.5381	0.5381	0.5381
50	0.4751	0.4736	0.4747	0.4943	0.5338	0.5367	0.5343	0.5343	0.5343	0.5343	0.5343
100	0.4510	0.4429	0.4315	0.4538	0.4962	0.5242	0.5273	0.5238	0.5238	0.5238	0.5238
150	0.4398	0.4435	0.4351	0.4369	0.4557	0.5016	0.5066	0.5002	0.4964	0.4964	0.4964
200	0.4379	0.4445	0.4477	0.4361	0.4480	0.4829	0.4894	0.4928	0.4932	0.4893	0.4893
250	0.4697	0.4376	0.4513	0.4361	0.4603	0.4920	0.5031	0.5148	0.5136	0.5098	0.5098
500	0.4664	0.4244	0.4508	0.4444	0.4589	0.4746	0.4990	0.4900	0.4995	0.5052	0.5058

Table 11 : iP[0.01] for 2007 collection – Child strategy – Tag set 2

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.4784	0.4693	0.4664	0.4768	0.4763	0.4840	0.4846	0.4846	0.4846	0.4846	0.4846
50	0.4410	0.4612	0.4410	0.4412	0.4467	0.4685	0.4716	0.4718	0.4718	0.4718	0.4718
100	0.4256	0.4497	0.4565	0.4576	0.4569	0.4525	0.4633	0.4638	0.4588	0.4588	0.4588
150	0.4237	0.4407	0.4594	0.4536	0.4662	0.4495	0.4457	0.4589	0.4615	0.4569	0.4569
200	0.4172	0.4359	0.4489	0.4547	0.4601	0.4409	0.4392	0.4447	0.4545	0.4541	0.4523
250	0.4161	0.4350	0.4496	0.4616	0.4543	0.4446	0.4412	0.4397	0.4506	0.4534	0.4495
500	0.4119	0.4297	0.4412	0.4514	0.4571	0.4458	0.4456	0.4380	0.4351	0.4429	0.4455

Table 12 : $iP[0.01]$ for 2008 collection – Child strategy – Tag set 1

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6252	0.6213	0.6155	0.6151	0.6266	0.6150	0.6150	0.6150	0.6150	0.6150	0.6150
50	0.6107	0.5961	0.6083	0.5800	0.6015	0.6155	0.6057	0.6057	0.6057	0.6057	0.6057
100	0.6162	0.5822	0.5698	0.5755	0.5778	0.5890	0.6035	0.6003	0.6003	0.6003	0.6003
150	0.6160	0.5975	0.5757	0.5630	0.5643	0.5751	0.5903	0.5907	0.5906	0.5906	0.5906
200	0.6107	0.5958	0.5742	0.5734	0.5661	0.5699	0.5840	0.5754	0.5819	0.5820	0.5820
250	0.6119	0.6065	0.5955	0.5846	0.5782	0.5733	0.5779	0.5904	0.5784	0.5849	0.5786
500	0.6082	0.5993	0.6060	0.5870	0.5941	0.5516	0.5762	0.5690	0.5759	0.5679	0.5717

Table 13 : $iP[0.01]$ for 2008 collection – Child strategy – Tag set 2

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6255	0.6170	0.5984	0.6090	0.6265	0.6223	0.6156	0.6156	0.6156	0.6156	0.6156
50	0.6299	0.6051	0.5982	0.6082	0.6013	0.6102	0.5919	0.5933	0.5933	0.5933	0.5933
100	0.6239	0.6133	0.5967	0.5719	0.5704	0.5784	0.5969	0.5833	0.5835	0.5835	0.5835
150	0.6133	0.6212	0.5875	0.5715	0.5590	0.5689	0.5920	0.5887	0.5713	0.5731	0.5731
200	0.6081	0.6099	0.6000	0.5772	0.5566	0.5759	0.5764	0.5935	0.5924	0.5726	0.5741
250	0.6233	0.6105	0.6012	0.5769	0.5606	0.5692	0.5679	0.5913	0.5845	0.5725	0.5745
500	0.6172	0.6100	0.6157	0.5840	0.5559	0.5557	0.5600	0.5564	0.5628	0.5687	0.5713

Observations and Analysis regarding the child strategy: For the *child* strategy of overlap removal, it is seen that the $iP[0.01]$ scores using tag set 1 are consistently higher than corresponding scores using tag set 2 for the INEX 2007 collection. For the 2008 collection, even though the best $iP[0.01]$ score using tag set 2 is marginally better than the best score using tag set 1, for a wide range of number of articles and elements retrieved, tag set 1 is seen to have better scores. This is because tag set 2

includes tags like list tags and table tag, which are expected to be smaller in size as terminal node tags. In the child strategy, these small child nodes would be getting preference in lieu of the bigger tags enclosing them, thus a smaller chunk of relevant text is identified as relevant. The best score of 0.6299 for this strategy using tag set 2, ranks us 21 out of 61 participants runs in the Focused retrieval task in INEX 2008 competition.

Experiment 3: Focused retrieval using *correlation* score strategy

Results are generated in this experiment by removing overlap from the rank ordered list of elements using the *correlation score* strategy. This experiment was performed on both the INEX 2007 and 2008 collections, using both the tag set 1 and tag set 2. Results generated by this experiment are shown in Tables 14, 15, 16 and 17.

Table 14 : iP[0.01] for 2007 collection – Correlation score strategy – Tag set 1

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.5029	0.4896	0.4898	0.4897	0.5041	0.5041	0.5041	0.5041	0.5041	0.5041	0.5041
50	0.4985	0.4813	0.4820	0.4820	0.5002	0.5003	0.5004	0.5004	0.5004	0.5004	0.5004
100	0.4885	0.4697	0.4701	0.4705	0.4910	0.4912	0.4913	0.4914	0.4914	0.4914	0.4914
150	0.4706	0.4712	0.4715	0.4715	0.4740	0.4740	0.4742	0.4718	0.4718	0.4718	0.4718
200	0.4738	0.4738	0.4740	0.4743	0.4765	0.4765	0.4765	0.4744	0.4744	0.4744	0.4744
250	0.4932	0.4739	0.4742	0.4742	0.4955	0.4955	0.4956	0.4956	0.4956	0.4956	0.4956
500	0.4998	0.4807	0.4810	0.4811	0.5030	0.5030	0.5030	0.5030	0.5030	0.5030	0.5030

Table 15 : iP[0.01] for 2007 collection – Correlation score strategy – Tag set 2

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.4710	0.4743	0.4747	0.4748	0.4751	0.4758	0.4758	0.4758	0.4758	0.4758	0.4758
50	0.4567	0.4594	0.4597	0.4602	0.4602	0.4602	0.4603	0.4603	0.4603	0.4603	0.4603
100	0.4555	0.4582	0.4587	0.4588	0.4590	0.4593	0.4593	0.4595	0.4595	0.4595	0.4595
150	0.4549	0.4573	0.4574	0.4577	0.4577	0.4578	0.4580	0.4580	0.4580	0.4580	0.4580
200	0.4534	0.4567	0.4569	0.4569	0.4571	0.4572	0.4573	0.4573	0.4573	0.4573	0.4573
250	0.4532	0.4565	0.4567	0.4567	0.4569	0.4570	0.4570	0.4570	0.4570	0.4570	0.4570
500	0.4508	0.4543	0.4547	0.4548	0.4548	0.4549	0.4549	0.4549	0.4549	0.4549	0.4549

Table 16 : iP[0.01] for 2008 collection – Correlation score strategy – Tag set 1

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6563	0.6599	0.6610	0.6620	0.6620	0.6621	0.6621	0.6621	0.6621	0.6621	0.6621
50	0.6576	0.6598	0.6603	0.6605	0.6607	0.6611	0.6611	0.6611	0.6611	0.6611	0.6611
100	0.6568	0.6592	0.6592	0.6592	0.6595	0.6595	0.6599	0.6599	0.6599	0.6599	0.6599
150	0.6563	0.6582	0.6583	0.6585	0.6585	0.6586	0.6592	0.6593	0.6593	0.6593	0.6593
200	0.6539	0.6554	0.6554	0.6556	0.6558	0.6558	0.6565	0.6565	0.6565	0.6565	0.6565
250	0.6553	0.6564	0.6565	0.6568	0.6568	0.6570	0.6571	0.6576	0.6576	0.6576	0.6576
500	0.6533	0.6544	0.6546	0.6549	0.6550	0.6551	0.6554	0.6554	0.6554	0.6560	0.6561

Observations and Analysis for correlation score strategy: In the 2008 collection, the correlation strategy of overlap removal has benefited by the use of tag set 2. This is because tag set 2 includes a more exhaustive list of terminal node elements, which in turn allows for a more accurate calculation of scores of the higher level elements. The

best score of 0.6898 using this strategy and tag set 2 ranks us First for the Focused retrieval task in the INEX 2008 competition.

Table 17 : iP[0.01] for 2008 collection – Correlation score strategy – Tag set 2

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6829	0.6855	0.6867	0.6880	0.6886	0.6898	0.6898	0.6898	0.6898	0.6898	0.6898
50	0.6650	0.6709	0.6715	0.6718	0.6718	0.6721	0.6726	0.6726	0.6726	0.6726	0.6726
100	0.6678	0.6692	0.6692	0.6692	0.6692	0.6692	0.6692	0.6692	0.6692	0.6692	0.6692
150	0.6668	0.6682	0.6682	0.6682	0.6682	0.6682	0.6682	0.6682	0.6682	0.6682	0.6682
200	0.6670	0.6684	0.6684	0.6684	0.6684	0.6684	0.6685	0.6685	0.6685	0.6685	0.6685
250	0.6681	0.6692	0.6692	0.6692	0.6692	0.6692	0.6692	0.6693	0.6693	0.6693	0.6693
500	0.6674	0.6685	0.6685	0.6685	0.6685	0.6685	0.6685	0.6685	0.6685	0.6685	0.6686

General Procedure for Experiments 4, 5 and 6.

Following is the general procedure for performing experiments 4, 5 and 6.

- 1) Perform article retrieval of n articles.
- 2) Perform flexible retrieval of m elements from n articles using Flex.
- 3) Remove all mt tag elements from this list
- 4) Remove overlap from the rank ordered list of elements generated in step 3 using the specified strategy.
- 5) Output the elements from the list in sorted order, grouped by article, i.e., sorted in order of the article list.

Observations and analysis for all 3 experiments are discussed after all the experiments are presented.

Experiment 4: Rearranging the *section* strategy focused output according to article scores

This experiment is performed by rearranging the list of focused elements produced by using the section strategy. This experiment was done on both the INEX 2007 and 2008 collections, using both tag set 1 and tag set 2. Results generated by this experiment can be found in Tables 18, 19, 20 and 21.

Table 18 : iP[0.01] for 2007 collection – Section strategy rearranged – Tag set 1

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.4991	0.4851	0.4920	0.4929	0.4976	0.4964	0.4965	0.4965	0.4965	0.4965	0.4965
50	0.5005	0.4899	0.4927	0.4869	0.5012	0.4959	0.4977	0.4977	0.4977	0.4977	0.4977
100	0.4995	0.4818	0.4918	0.4844	0.5040	0.4945	0.4983	0.4978	0.4979	0.4979	0.4979
150	0.4781	0.4841	0.4853	0.4887	0.4872	0.4943	0.4913	0.4884	0.4891	0.4892	0.4892
200	0.4817	0.4825	0.4855	0.4867	0.4891	0.4934	0.4858	0.4896	0.4884	0.4892	0.4892
250	0.5038	0.4816	0.4863	0.4849	0.5061	0.5039	0.4971	0.5016	0.4978	0.4982	0.4982
500	0.5050	0.4759	0.4841	0.4872	0.5087	0.5058	0.5011	0.4960	0.5009	0.4999	0.4981

Table 19 : iP[0.01] for 2007 collection – Section strategy rearranged – Tag set 2

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.4652	0.4759	0.4847	0.4818	0.4810	0.4798	0.4804	0.4804	0.4804	0.4804	0.4804
50	0.4597	0.4653	0.4702	0.4799	0.4810	0.4791	0.4772	0.4773	0.4773	0.4773	0.4773
100	0.4630	0.4664	0.4659	0.4723	0.4725	0.4786	0.4785	0.4774	0.4773	0.4776	0.4776
150	0.4612	0.4613	0.4687	0.4677	0.4724	0.4717	0.4782	0.4786	0.4775	0.4776	0.4776
200	0.4600	0.4618	0.4679	0.4667	0.4720	0.4725	0.4770	0.4771	0.4786	0.4773	0.4777
250	0.4625	0.4621	0.4665	0.4659	0.4704	0.4727	0.4779	0.4776	0.4790	0.4776	0.4775
500	0.4569	0.4616	0.4672	0.4664	0.4684	0.4718	0.4713	0.4776	0.4774	0.4787	0.4786

Table 20 : iP[0.01] for 2008 collection – Section strategy rearranged – Tag set 1

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.7028	0.6965	0.6978	0.7131	0.7091	0.7093	0.7093	0.7093	0.7093	0.7093	0.7093
50	0.7089	0.6933	0.7053	0.7008	0.6973	0.7095	0.7090	0.7090	0.7090	0.7090	0.7090
100	0.6971	0.7059	0.7033	0.7019	0.7147	0.6948	0.7102	0.7105	0.7104	0.7104	0.7104
150	0.6989	0.7061	0.6968	0.7076	0.7089	0.6942	0.7122	0.7113	0.7116	0.7114	0.7114
200	0.6923	0.7056	0.6993	0.7064	0.7103	0.7017	0.7093	0.7081	0.7114	0.7112	0.7112
250	0.6932	0.7032	0.7003	0.7044	0.7042	0.7033	0.6961	0.7119	0.7112	0.7114	0.7113
500	0.6935	0.6878	0.7086	0.6982	0.7073	0.7076	0.6987	0.6965	0.7112	0.7123	0.7117

Table 21 : iP[0.01] for 2008 collection – Section strategy rearranged – Tag set 2

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.7172	0.7092	0.7056	0.7188	0.7171	0.7188	0.7195	0.7195	0.7195	0.7195	0.7195
50	0.6953	0.7150	0.7151	0.7166	0.7072	0.7203	0.7200	0.7204	0.7204	0.7204	0.7204
100	0.6956	0.7143	0.7111	0.7123	0.7176	0.7094	0.7215	0.7207	0.7209	0.7211	0.7211
150	0.6949	0.7157	0.7089	0.7104	0.7122	0.7201	0.7194	0.7216	0.7209	0.7214	0.7214
200	0.6835	0.7104	0.7093	0.7114	0.7137	0.7217	0.7205	0.7205	0.7221	0.7212	0.7217
250	0.6833	0.7089	0.7098	0.7100	0.7122	0.7213	0.7096	0.7203	0.7207	0.7210	0.7214
500	0.6851	0.7097	0.7108	0.7078	0.7101	0.7178	0.7213	0.7101	0.7199	0.7204	0.7223

Experiment 5: Rearranging the *child* strategy focused output according to article scores

The procedure for this experiment is similar to that of experiment 4, except that the focused output generated by the child strategy of overlap removal is rearranged according to the article rankings. This experiment was also used the 2007 and 2008 collections and both the tag sets.

Table 22 : iP[0.01] for 2007 collection – Child strategy rearranged – Tag set 1

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.4975	0.5039	0.5101	0.5159	0.5233	0.5159	0.5149	0.5149	0.5149	0.5149	0.5149
50	0.4924	0.5001	0.5101	0.5026	0.5127	0.5217	0.5165	0.5165	0.5165	0.5165	0.5165
100	0.4908	0.4857	0.5057	0.5072	0.5218	0.5154	0.5189	0.5170	0.5169	0.5169	0.5169
150	0.4734	0.4972	0.4961	0.5073	0.5091	0.5099	0.5197	0.5147	0.5103	0.5103	0.5103
200	0.4770	0.4909	0.4867	0.5033	0.5093	0.5059	0.5081	0.5163	0.5106	0.5103	0.5103
250	0.4974	0.4920	0.4906	0.4968	0.5193	0.5151	0.5171	0.5270	0.5215	0.5173	0.5170
500	0.4961	0.4800	0.4878	0.4971	0.5191	0.5212	0.5123	0.5133	0.5189	0.5241	0.5218

Table 23 : iP[0.01] for 2007 collection – Child strategy rearranged – Tag set 2

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.4636	0.4881	0.5013	0.4843	0.4872	0.4881	0.4884	0.4884	0.4884	0.4884	0.4884
50	0.4687	0.4768	0.4811	0.4994	0.4887	0.4873	0.4856	0.4855	0.4855	0.4855	0.4855
100	0.4709	0.4716	0.4794	0.4912	0.4957	0.4881	0.4891	0.4856	0.4855	0.4857	0.4857
150	0.4666	0.4617	0.4741	0.4842	0.4955	0.4765	0.4845	0.4886	0.4857	0.4858	0.4858
200	0.4648	0.4612	0.4674	0.4819	0.4938	0.4800	0.4789	0.4886	0.4874	0.4856	0.4858
250	0.4658	0.4621	0.4667	0.4699	0.4916	0.4847	0.4860	0.4843	0.4895	0.4858	0.4857
500	0.4571	0.4619	0.4771	0.4662	0.4819	0.4943	0.4786	0.4849	0.4825	0.4883	0.4860

Table 24 : iP[0.01] for 2008 collection – Child strategy rearranged – Tag set 1

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.7060	0.6927	0.6931	0.7179	0.7142	0.7158	0.7158	0.7158	0.7158	0.7158	0.7158
50	0.7128	0.6932	0.7002	0.6869	0.6894	0.7234	0.7160	0.7160	0.7160	0.7160	0.7160
100	0.7029	0.7030	0.7040	0.6971	0.7113	0.6961	0.7160	0.7179	0.7176	0.7176	0.7176
150	0.7032	0.7051	0.7000	0.6992	0.6982	0.6859	0.7177	0.7174	0.7183	0.7181	0.7181
200	0.6966	0.7056	0.7046	0.7017	0.6998	0.6914	0.7204	0.7126	0.7183	0.7182	0.7183
250	0.6958	0.7080	0.7054	0.7073	0.6985	0.6993	0.7002	0.7250	0.7151	0.7185	0.7184
500	0.6961	0.6933	0.7084	0.7016	0.7044	0.7153	0.6953	0.7010	0.7167	0.7162	0.7186

Table 25 : iP[0.01] for 2008 collection – Child strategy rearranged – Tag set 2

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.7087	0.7016	0.6977	0.7124	0.7183	0.7231	0.7179	0.7179	0.7179	0.7179	0.7179
50	0.6967	0.7093	0.7021	0.7051	0.7046	0.7211	0.7189	0.7192	0.7192	0.7192	0.7192
100	0.6936	0.7110	0.7104	0.7088	0.7056	0.7001	0.7221	0.7194	0.7199	0.7198	0.7198
150	0.6920	0.7148	0.7047	0.7065	0.7043	0.7145	0.7204	0.7237	0.7198	0.7201	0.7201
200	0.6753	0.7087	0.7068	0.7079	0.7099	0.7262	0.7167	0.7211	0.7252	0.7199	0.7203
250	0.6757	0.7073	0.7086	0.7057	0.7084	0.7207	0.7068	0.7157	0.7217	0.7199	0.7204
500	0.6744	0.7083	0.7097	0.7066	0.7047	0.7206	0.7110	0.7069	0.7147	0.7208	0.7264

Experiment 6: Rearranging the correlation strategy focused output according to article scores

The procedure for this experiment is similar to that of experiment 4, except that the focused output generated by the *correlation score* strategy of overlap removal is rearranged according to the article rankings.

This experiment was also done on the 2007 and 2008 collections using both of the tag sets.

Table 26 : iP[0.01] for 2007 collection – Correlation strategy rearranged – Tag set 1

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.4890	0.4827	0.4905	0.4898	0.4864	0.4855	0.4855	0.4855	0.4855	0.4855	0.4855
50	0.4908	0.4823	0.4874	0.4819	0.4899	0.4866	0.4866	0.4866	0.4866	0.4866	0.4866
100	0.4919	0.4787	0.4836	0.4808	0.4888	0.4882	0.4864	0.4863	0.4863	0.4863	0.4863
150	0.4808	0.4793	0.4840	0.4834	0.4822	0.4884	0.4855	0.4852	0.4852	0.4853	0.4853
200	0.4832	0.4791	0.4827	0.4809	0.4823	0.4885	0.4866	0.4849	0.4851	0.4853	0.4853
250	0.4964	0.4809	0.4831	0.4802	0.4915	0.4907	0.4888	0.4866	0.4866	0.4864	0.4863
500	0.5004	0.4771	0.4788	0.4828	0.4947	0.4916	0.4886	0.4887	0.4879	0.4869	0.4867

Table 27 : iP[0.01] for 2007 collection – Correlation strategy rearranged – Tag set 2

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.4724	0.4772	0.4855	0.4821	0.4798	0.4793	0.4796	0.4796	0.4796	0.4796	0.4796
50	0.4715	0.4720	0.4732	0.4817	0.4816	0.4775	0.4759	0.4759	0.4759	0.4759	0.4759
100	0.4766	0.4723	0.4724	0.4742	0.4751	0.4788	0.4772	0.4759	0.4758	0.4761	0.4761
150	0.4747	0.4756	0.4730	0.4736	0.4743	0.4720	0.4774	0.4773	0.4760	0.4761	0.4761
200	0.4734	0.4765	0.4728	0.4730	0.4735	0.4735	0.4768	0.4761	0.4771	0.4759	0.4761
250	0.4754	0.4743	0.4745	0.4727	0.4742	0.4736	0.4773	0.4769	0.4775	0.4760	0.4760
500	0.4680	0.4750	0.4771	0.4741	0.4734	0.4735	0.4712	0.4779	0.4772	0.4777	0.4772

Observations and Analysis for Experiments 4, 5 and 6: Comparing the results for experiments 4, 5 and 6 with their corresponding non-rearranged versions (i.e., experiments 1,2 and 3), we observed that reranking the elements based on the correlation score of their articles gives better results. This observation has led us to the conclusion that along with the correlation score of the elements themselves, the

correlation score of the articles enclosing these elements plays an important role in ranking the elements. The best score of 0.7264, obtained by rearranging the focused results obtained by child strategy give us a ranking of 1 (by a large margin) in the focused retrieval task of the INEX 2008 competition.

Table 28 : iP[0.01] for 2008 collection – Correlation strategy rearranged – Tag set 1

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6933	0.7017	0.7013	0.7040	0.6995	0.7000	0.7000	0.7000	0.7000	0.7000	0.7000
50	0.6979	0.6909	0.7000	0.7019	0.7003	0.7009	0.7004	0.7004	0.7004	0.7004	0.7004
100	0.6930	0.6984	0.6975	0.6974	0.7030	0.7017	0.7010	0.7010	0.7010	0.7010	0.7010
150	0.6902	0.6980	0.6951	0.6994	0.6983	0.7004	0.7059	0.7018	0.7019	0.7018	0.7018
200	0.6841	0.6968	0.6945	0.6987	0.6987	0.7031	0.7038	0.7023	0.7018	0.7015	0.7015
250	0.6845	0.6963	0.6944	0.6995	0.6985	0.7045	0.7028	0.7057	0.7020	0.7018	0.7016
500	0.6854	0.6899	0.7020	0.6959	0.6995	0.6985	0.7031	0.7032	0.7061	0.7063	0.7019

Table 29 : iP[0.01] for 2008 collection – Correlation strategy rearranged – Tag set 2

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.7160	0.7110	0.7169	0.7164	0.7150	0.7159	0.7166	0.7166	0.7166	0.7166	0.7166
50	0.7097	0.7115	0.7161	0.7170	0.7203	0.7175	0.7171	0.7175	0.7175	0.7175	0.7175
100	0.7085	0.7133	0.7113	0.7124	0.7184	0.7199	0.7189	0.7177	0.7178	0.7180	0.7180
150	0.7079	0.7146	0.7109	0.7116	0.7125	0.7194	0.7175	0.7190	0.7179	0.7182	0.7182
200	0.7021	0.7113	0.7087	0.7123	0.7133	0.7211	0.7181	0.7182	0.7191	0.7180	0.7184
250	0.7022	0.7115	0.7086	0.7135	0.7126	0.7219	0.7196	0.7179	0.7182	0.7179	0.7181
500	0.7038	0.7112	0.7092	0.7110	0.7130	0.7183	0.7207	0.7202	0.7178	0.7180	0.7195

Experiment 7: Using Flex chosen documents for element retrieval

This experiment aims to evaluate the use of article retrieval to identify the documents that are most likely to contain highly correlating elements. Experiments 1 - 6 use article retrieval to identify important documents. This experiment uses Flex alone to identify these documents.

Following is the procedure followed for this experiment:

For each query

- 1) Retrieve 125,000 terminal node element using Smart retrieval to guarantee complete trees.
- 2) Seed the doctrees with the elements from this list.
- 3) Using the top m_l elements from the list of elements, identify documents which contain these elements. Choose m_l such that $m_l > n$, where n is the number of documents considered as important.
- 4) Select the seeded doctrees of the identified documents. This will create a seed subset which acts as input to Flex.
- 5) Using Flex, retrieve m elements from the documents selected above. Note that the number of documents used in this retrieval will be an approximation of n (i.e., the number of documents to be considered as important).
- 6) Remove all elements mt -tagged elements from the list generated in step 5.

- 7) Remove the overlapping elements from this list (using one of the 3 focussing strategies) to generate focused results.

This experiment was performed only on the 2008 document and query collection using tag set 1. Results of these experiments are shown in tables 30,31 and 32.

Table 30 : iP[0.01] for 2008 collection – Section – Flex chosen documents

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6356	0.6315	0.6290	0.6297	0.6300	0.6302	0.6302	0.6302	0.6302	0.6302	0.6302
50	0.6739	0.6551	0.6523	0.6466	0.6534	0.6521	0.6521	0.6521	0.6521	0.6521	0.6521
100	0.6657	0.6497	0.6504	0.6428	0.6399	0.6398	0.6379	0.6379	0.6379	0.6379	0.6379
150	0.6708	0.6572	0.6533	0.6495	0.6404	0.6328	0.6332	0.6335	0.6335	0.6335	0.6335
200	0.6704	0.6742	0.6517	0.6482	0.6407	0.6361	0.6328	0.6327	0.6329	0.6329	0.6329
250	0.6686	0.6722	0.6500	0.6454	0.6443	0.6342	0.6308	0.6305	0.6303	0.6305	0.6305
500	0.6675	0.6703	0.6503	0.6496	0.6435	0.6290	0.6319	0.6286	0.6272	0.6273	0.6275

Table 31 : iP[0.01] for 2008 collection – Child – Flex chosen documents

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.5918	0.5960	0.5935	0.5859	0.5873	0.5771	0.5771	0.5771	0.5771	0.5771	0.5771
50	0.6384	0.6238	0.5855	0.5794	0.5895	0.5679	0.5716	0.5716	0.5716	0.5716	0.5716
100	0.6147	0.6030	0.5975	0.5985	0.5931	0.5895	0.5752	0.5684	0.5684	0.5684	0.5684
150	0.6164	0.6068	0.6135	0.6001	0.6006	0.5928	0.5646	0.5730	0.5743	0.5665	0.5665
200	0.6148	0.6165	0.6086	0.6041	0.5912	0.5900	0.5807	0.5675	0.5714	0.5657	0.5657
250	0.6124	0.6091	0.6035	0.5938	0.5975	0.5796	0.5685	0.5627	0.5673	0.5723	0.5642
500	0.6111	0.6029	0.6031	0.5905	0.5890	0.5806	0.5715	0.5658	0.5702	0.5619	0.5683

Table 32 : iP[0.01] for 2008 collection – Correlation – Flex chosen documents

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6285	0.6299	0.6334	0.6345	0.6345	0.6348	0.6348	0.6348	0.6348	0.6348	0.6348
50	0.6590	0.6596	0.6600	0.6605	0.6630	0.6633	0.6633	0.6633	0.6633	0.6633	0.6633
100	0.6501	0.6521	0.6527	0.6527	0.6528	0.6570	0.6570	0.6570	0.6570	0.6570	0.6570
150	0.6568	0.6580	0.6587	0.6590	0.6592	0.6592	0.6620	0.6620	0.6620	0.6620	0.6620
200	0.6565	0.6579	0.6585	0.6590	0.6594	0.6594	0.6616	0.6616	0.6616	0.6616	0.6616
250	0.6545	0.6555	0.6564	0.6571	0.6575	0.6575	0.6576	0.6588	0.6589	0.6589	0.6589
500	0.6524	0.6535	0.6544	0.6550	0.6556	0.6558	0.6558	0.6558	0.6559	0.6559	0.6559

Observations and Analysis: Comparing Tables 6, 10, and 14 with 30, 31 and 32 it is observed that by letting Flex choose the articles/documents of *interest*, the results generated are comparable to those generated by using article retrieval. The best score of 0.6742 produced by this approach ranks us as fourth out of the 61 participant runs for the focused retrieval task. These results indicate that Flex does an equally good job of identifying articles which are most likely to have highly correlating elements.

Experiment 8: Use of All-element slope and pivot value by Flex

It was found that Flex used different slope and pivot values at the different levels of the document trees. In this experiment, we have modified Flex to calculate the *Lnu* and *ltu* weights of the element and query vectors using the all-element values of slope and pivot. This experiment is done on the 2008 query collection using tag set 2.

Results for this experiment are shown in Tables 33, 34, 35, 36, 37 and 38.

Table 33 : iP[0.01] for 2008 collection – Section – Using All-el slope and pivot values

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6506	0.6548	0.6541	0.6539	0.6543	0.6550	0.6551	0.6551	0.6551	0.6551	0.6551
50	0.6460	0.6467	0.6456	0.6430	0.6415	0.6435	0.6427	0.6427	0.6427	0.6427	0.6427
100	0.6469	0.6471	0.6412	0.6398	0.6411	0.6387	0.6389	0.6389	0.6389	0.6389	0.6389
150	0.6427	0.6430	0.6403	0.6371	0.6357	0.6340	0.6351	0.6342	0.6343	0.6344	0.6344
200	0.6394	0.6379	0.6360	0.6326	0.6316	0.6314	0.6288	0.6306	0.6295	0.6295	0.6295
250	0.6378	0.6368	0.6367	0.6338	0.6300	0.6299	0.6267	0.6285	0.6276	0.6275	0.6276
500	0.6453	0.6423	0.6347	0.6317	0.6317	0.6275	0.6252	0.6248	0.6256	0.6267	0.6251

Table 34 : iP[0.01] for 2008 collection – Child - Using All-el slope and pivot values

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6214	0.6213	0.5974	0.5999	0.6037	0.6041	0.6043	0.6043	0.6043	0.6043	0.6043
50	0.6238	0.6268	0.6218	0.6038	0.5989	0.6062	0.6043	0.6054	0.6054	0.6054	0.6054
100	0.6284	0.6249	0.6215	0.6157	0.6184	0.6006	0.6027	0.6016	0.6025	0.6025	0.6025
150	0.6268	0.6233	0.6185	0.6115	0.6096	0.5928	0.5959	0.5959	0.5958	0.5956	0.5956
200	0.6220	0.6173	0.6101	0.6028	0.6040	0.5914	0.5870	0.5895	0.5886	0.5881	0.5884
250	0.6196	0.6159	0.6115	0.6026	0.5997	0.5888	0.5829	0.5855	0.5837	0.5851	0.5851
500	0.6283	0.6150	0.6106	0.6018	0.6009	0.5870	0.5818	0.5799	0.5803	0.5828	0.5802

Table 35 : iP[0.01] for 2008 collection – Correlation – Using All-el slope and pivot values

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6728	0.6784	0.6784	0.6784	0.6786	0.6799	0.6799	0.6799	0.6799	0.6799	0.6799
50	0.6634	0.6691	0.6711	0.6711	0.6711	0.6711	0.6716	0.6716	0.6716	0.6716	0.6716
100	0.6593	0.6637	0.6642	0.6648	0.6651	0.6651	0.6651	0.6651	0.6651	0.6651	0.6651
150	0.6555	0.6602	0.6614	0.6616	0.6617	0.6620	0.6620	0.6620	0.6620	0.6620	0.6620
200	0.6523	0.6573	0.6580	0.6580	0.6580	0.6580	0.6580	0.6581	0.6581	0.6581	0.6581
250	0.6514	0.6567	0.6574	0.6574	0.6574	0.6574	0.6574	0.6574	0.6575	0.6575	0.6575
500	0.6494	0.6554	0.656	0.656	0.656	0.656	0.6561	0.6561	0.6561	0.6561	0.6561

Table 36 : iP[0.01] for 2008 collection – Section Rearranged - Using All-el slope and pivot values

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.6971	0.7147	0.7041	0.7076	0.7191	0.7198	0.7211	0.7211	0.7211	0.7211	0.7211
50	0.6844	0.7131	0.7197	0.7039	0.7065	0.7200	0.7211	0.7222	0.7222	0.7222	0.7222
100	0.6923	0.7096	0.7171	0.7191	0.7190	0.7078	0.7210	0.7217	0.7229	0.7229	0.7229
150	0.6977	0.7092	0.7162	0.7192	0.7171	0.7111	0.7199	0.7228	0.7215	0.7233	0.7233
200	0.6918	0.7057	0.7084	0.7145	0.7185	0.7190	0.7086	0.7210	0.7223	0.7224	0.7235
250	0.6896	0.7029	0.7103	0.7153	0.7177	0.7207	0.7080	0.7197	0.7218	0.7221	0.7229
500	0.6817	0.7034	0.7108	0.7108	0.7128	0.7185	0.7100	0.7088	0.7221	0.7203	0.7233

Table 37 : iP[0.01] for 2008 collection – Child Rearranged - Using All-el slope and pivot values

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.7063	0.7100	0.7038	0.7068	0.7254	0.7258	0.7205	0.7205	0.7205	0.7205	0.7205
50	0.6925	0.7203	0.7188	0.6984	0.7044	0.7283	0.7212	0.7218	0.7218	0.7218	0.7218
100	0.7039	0.7169	0.7213	0.7169	0.7206	0.7074	0.7268	0.7210	0.7224	0.7224	0.7224
150	0.7052	0.7190	0.7224	0.7159	0.7165	0.7090	0.7218	0.7290	0.7274	0.7227	0.7227
200	0.7022	0.7128	0.7159	0.7173	0.7184	0.7133	0.7090	0.7274	0.7293	0.7222	0.7230
250	0.6982	0.7112	0.7180	0.7170	0.7154	0.7139	0.7082	0.7221	0.7278	0.7214	0.7227
500	0.6897	0.7104	0.7194	0.7136	0.7139	0.7179	0.7077	0.7087	0.7236	0.7268	0.7307

Table 38 : iP[0.01] for 2008 collection – Correlation Rearranged – Using All-el slope and pivot values

# of Docs (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.7064	0.7217	0.7197	0.7210	0.7195	0.7191	0.7203	0.7203	0.7203	0.7203	0.7203
50	0.6949	0.7169	0.7229	0.7214	0.7227	0.7199	0.7202	0.7213	0.7213	0.7213	0.7213
100	0.6997	0.7141	0.7205	0.7225	0.7228	0.7223	0.7211	0.7208	0.7219	0.7219	0.7219
150	0.7043	0.7111	0.7198	0.7236	0.7209	0.7265	0.7207	0.7223	0.7205	0.7222	0.7222
200	0.6987	0.7083	0.7149	0.7180	0.7222	0.7235	0.7214	0.7213	0.7215	0.7213	0.7223
250	0.6966	0.7085	0.7164	0.7188	0.7219	0.7254	0.7217	0.7206	0.7217	0.7211	0.7217
500	0.6895	0.7087	0.7151	0.7169	0.7168	0.7241	0.7246	0.7226	0.7228	0.7207	0.7226

Observations and Analysis: If we compare results of this experiment with previous ones (i.e., compare Tables 9,13,17,21,25 and 29 with Tables 33,34,35,36,37 and 38) we can see that the results are similar and follow the same trends with respect the number of elements and number of articles retrieved. However, using the all-element

value of slope and pivot at all levels of the tree has improved our best result for rearrangement of child strategy to 0.7307 from 0.7264.

4.5 Conclusions

Experiments 1, 2 and 3 present the different overlap removal strategies that we used. While the comparison of the results for these strategies appear inconclusive with respect to the 2007 collection, the section strategy gives better results for the 2008 collection using tag set 1 than the child and correlation strategy. This is because child strategy always gives preference to the child nodes at the lowest level in the document tree (thereby usually selecting very small sized elements) and the correlation strategy gives preference to the nodes having the highest correlation score (thereby usually selecting elements with large size). On the other hand, section strategy takes a middle path, i.e., gives preference to elements with high correlation scores that lie between the top level nodes and the bottom level nodes in a document tree.

Experiments 4, 5 and 6 clearly show that the rearrangement of focused results based on article rankings give better results. In fact the best score generated by rearrangement ($iP[0.01] = 0.7264$) beats the score of the participant at rank 1 ($iP[0.01] = 0.6896$) by a large margin. When the number of articles (n) ranges from 25 to 500 and the number of elements (m) ranges from 50 to 4000, Figures 10, 11 and 12 compare the average of the $iP[0.01]$ scores for both the rearranged and non-rearranged focused output.

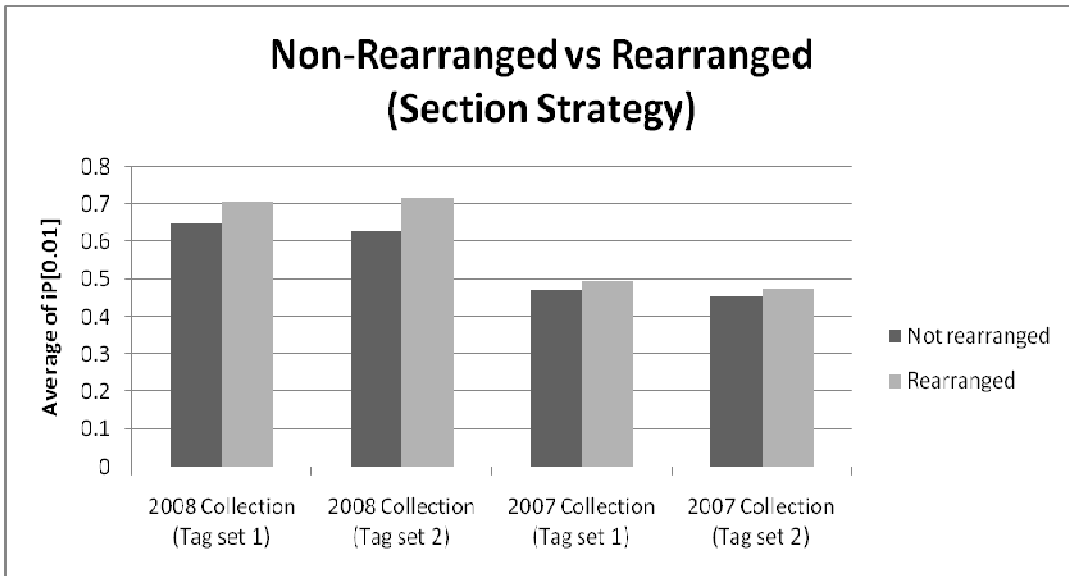


Figure 10 : Comparison of non-rearranged and rearranged output for section strategy.

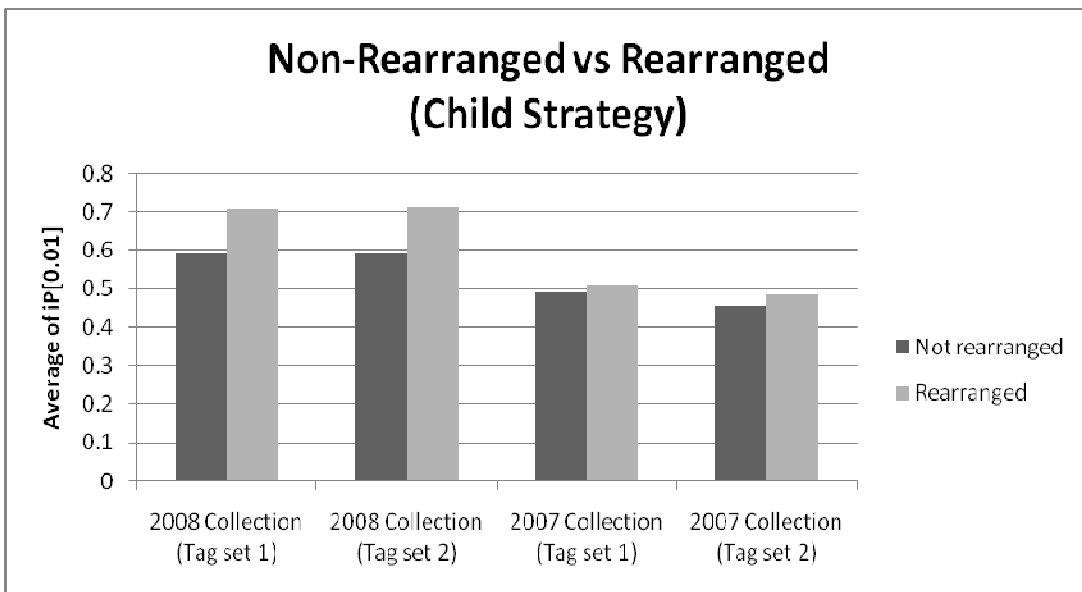


Figure 11 : Comparison of non-rearranged and rearranged output for child strategy.

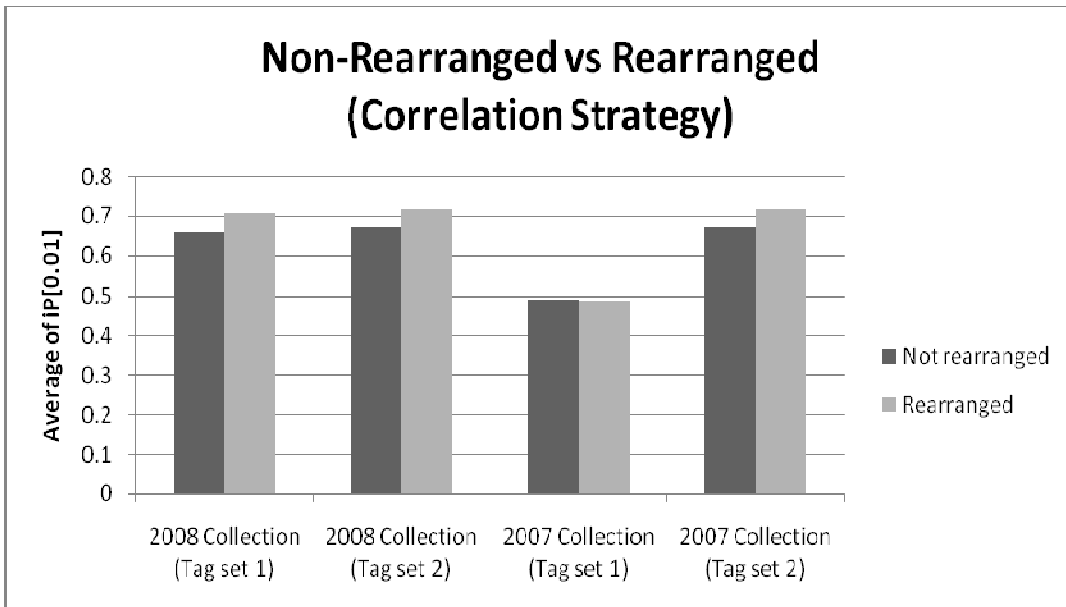


Figure 12 : Comparison of non-rearranged and rearranged output for correlation strategy.

Results of experiment 7 confirm the assumption in [8] that Flex can identify the documents that are most likely to contain elements with high correlation score. Hence, we can combine the conclusions of experiment 4, 5, 6 and 7 to say that while correlation scores of articles can provide us with important information for deciding the rank of elements, we do not need article retrieval for identifying the documents of *interest*.

5. Future Work

In this section, we make suggestions for future work and further improvement in results of the Ad Hoc Focused retrieval task.

Our experiments with rearranging the focused output based on the rankings of the articles have provided us with initial evidence that point to the utility of article ranking in deciding ranks for elements. We currently simply group the elements by article and present these groups according to article rank. This work can be extended by designing a weighting function for calculating the correlation score of an element based on its own score and that of the article enclosing it. The weights assigned to the article score and to the element score can be determined experimentally or can be calculated based on a parameter like the ratio of their size. Another alternative for this rearrangement could be the addition of a bias (based on the correlation score of the article) to the correlation score of an element.

A more thorough analysis of the relevance assessments can be done to identify a good tag set for retrieval. Analyzing the INEX 2008 qrels could make the task of analyzing the relevance assessments simpler.

Our observation that FLEX indeed does a good job at selecting good documents, leads us to believe that article retrieval in the Focused context is unnecessary. This can go a long way in improving the system's overall operational efficiency and eliminate the need for article indexing.

The values for slope and pivot need to be experimentally confirmed on the 2009 collection.

INEX is already encouraging work on retrieval of passages as against to retrieval of only individual elements. Much work needs to be done in extending the Focused retrieval and Relevant in Context retrieval to passage retrieval. Combining individual element scores into passage score also provides a challenging problem.

References

- [1] Bakshi, V. Flexible Retrieval for the Semi-Structured Documents. MS Thesis, Department of Computer Science, University of Minnesota Duluth, 2006. http://www.d.umn.edu/cs/thesis/vishal_bakshi_ms.pdf
- [2] Bapat, S. Improving Results for Focused and Relevant-In-Context Tasks. MS Thesis, Department of Computer Science, University of Minnesota Duluth, 2008. http://www.d.umn.edu/cs/thesis/salil_bapat_ms.pdf
- [3] Buckley, C. Implementation of the Smart Information Retrieval System. Technical Report TR85-686, Cornell University, 1985.
- [4] Crouch, C. Dynamic Element Retrieval in a Structured Environment. *ACM TOIS*, 24(4), 437-454, 2006.
- [5] Denoyer, L., Gillinari, P. The Wikipedia XML Corpus. *SIGIR Forum*, 40(1), 64-69, 2006.
- [6] Geva, S., Kamps, J., Trotman, A. INEX 2008 Workshop Pre-proceedings. <http://www.inex.otago.ac.nz/data/publications.asp>
- [7] Kamps, J., Pehcevski, J., Kazai, G., Lalmas, M., Robertson, S. INEX 2007 Evaluation Measures. *Focused Access to XML Documents*, Springer, LCNS 4862, 70-79, 2008.
- [8] Khanna, S. Design and Implementation of a Flexible Retrieval System. MS Thesis, Department of Computer Science, University of Minnesota Duluth, 2005. http://www.d.umn.edu/cs/thesis/sudip_khanna_ms.pdf
- [9] Manning, C., Praghavan, P., Shutze, H. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [10] Mone, A. Dynamic Element Retrieval for Semi-Structured Documents. MS Thesis, Department of Computer Science, University of Minnesota Duluth, 2007. http://www.d.umn.edu/cs/thesis/aditya_mone_ms.pdf
- [11] Paranjape, D. Improving Focused Retrieval. MS Thesis, Department of Computer Science, University of Minnesota Duluth, 2008. http://www.d.umn.edu/cs/thesis/darshan_paranjape_ms.pdf

- [12] Polumetla, C. Improving the Results of the Relevant in Context Task. MS Thesis, Department of Computer Science, University of Minnesota Duluth, 2009.
- [13] Poluri, P. Focused Retrieval using the Exact Methodology. MS Thesis, Department of Computer Science, University of Minnesota, Duluth 2009.
- [14] Salton, G., ed. *The Smart Retrieval System – Experiments in Automatic Document Processing*. Prentice-Hall, 1971.
- [15] Salton, G., Wong, A., Yang, C.A. A Vector Space Model for Automatic Indexing, *Comm. ACM*, 18(11), 613-620, 1975.
- [16] Schenkel, R., Suchanek, F., Kasneci, G. YAWN: A semantically annotated Wikipedia XML Corpus, 2007.
<http://www.mpi-inf.mpg.de/~kasneci/download/BTW2007.pdf>
- [17] Singhal, A. AT&T at TREC-6, *The Sixth Text REtrieval Conf (TREC-6)*, 215 – 225, 1998.
- [18] Singhal, A., Buckley, C., Mitra, M. Pivoted Document Length Normalization. *Proc. of the 19th Annual International ACM SIGIR Conference*, 19-21, 1996.
- [19] Sudhakar, V. Improving Results for the Best in Context Task. MS Project, Department of Computer Science, University of Minnesota Duluth, 2009.

Appendix A

XPath expansion

Until INEX 2006, the evaluation of results was based on the XPaths returned and hence we needed to correct the XPaths before returning as results. Refer [2] for more details on this.

However, the INEX 2007 and INEX 2008 evaluation package calculates the correlation scores based on the text marked and text returned, and hence the XPath expansion was not required. This observation is confirmed by running focused retrieval task of the 2007 competition and evaluating the results, using the 2007 evaluation tool, with and without expanding the XPaths.

The following tables present the results of this experiment. The rows represent the number of articles used for retrieval and the columns represent the number of elements retrieved.

Table 39 : $iP[0.01]$ for 2007 Focused retrieval, 2007 Evaluation, without XPath expansion

# of articles (N)	# of elements (M)		
	1000	1500	2000
25	0.5381	0.5381	0.5381
50	0.5343	0.5343	0.5343
100	0.5273	0.5238	0.5238
250	0.5031	0.5148	0.5136
500	0.4990	0.4900	0.4995

Table 40 : iP[0.01] for 2007 Focused retrieval, 2007 Evaluation, with XPath expansion

# of articles (N)	# of elements (M)		
	1000	1500	2000
25	0.5381	0.5381	0.5381
50	0.5343	0.5343	0.5343
100	0.5273	0.5238	0.5238
250	0.5031	0.5148	0.5136
500	0.4990	0.4900	0.4995

These results confirm that the evaluation in INEX 2007 does not take into account the correctness of the XPaths. It only considers the text that has been returned as the result, and the unimportant tags (generally the formatting tags) can be safely ignored. In all the experiments performed, the expansion of XPaths step has been dropped.

Removal of minus words from the queries

Some queries from the INEX 2007 query collection are of the form “term1 term2 .. – termN1 termN2 ... “, where the – sign indicates that the elements correlating to the terms following it should not be considered relevant. For example, a query like “apple – computers” means that the user is looking for information related to “apple” but *not* related to “computers”. To tackle these kinds of queries, the INEX 2007 query collection is modified by removing all the terms that followed a – sign in them. This modified query collection is then indexed and then used for retrieval. Refer to [2] for results produced without this modification of queries.

[2] reports on the results of an experiment which is in fact identical to that reported in Table 35 with one exception, that of the minus words being removed.

Table 41 : 2007 Focused retrieval – Child Strategy – Tag set 1.

# of Documents (N)	# of elements (M)										
	50	100	150	200	250	500	1000	1500	2000	3000	4000
25	0.4972	0.4734	0.5089	0.5144	0.5383	0.5386	0.5381	0.5381	0.5381	0.5381	0.5381
50	0.4751	0.4736	0.4747	0.4943	0.5338	0.5367	0.5343	0.5343	0.5343	0.5343	0.5343
100	0.4510	0.4429	0.4315	0.4538	0.4962	0.5242	0.5273	0.5238	0.5238	0.5238	0.5238
150	0.4398	0.4435	0.4351	0.4369	0.4557	0.5016	0.5066	0.5002	0.4964	0.4964	0.4964
200	0.4379	0.4445	0.4477	0.4361	0.4480	0.4829	0.4894	0.4928	0.4932	0.4893	0.4893
250	0.4697	0.4376	0.4513	0.4361	0.4603	0.4920	0.5031	0.5148	0.5136	0.5098	0.5098
500	0.4664	0.4244	0.4508	0.4444	0.4589	0.4746	0.4990	0.4900	0.4995	0.5052	0.5058

The only difference between the focused retrieval experiment in [15] and this experiment, is the removal of negative words. It can be seen that the best score previously achieved (0.5266), was improved to 0.5386 by removing the negative words from the queries. In both the cases, the best result was found at 25 articles and 500 elements. This result has confirmed the utility of removal of negative words from the query. Hence, for all the further experiments involved with the 2007 and 2008 query collections, we have removed the negative words from the query.