

**Producing Improved Results for the INEX Focused
and Relevant in Context Tasks**

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA

BY

Sandeep Vadlamudi

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Donald B. Crouch

August, 2010

Acknowledgements

I would like to thank all the people who supported me during the course of my thesis.

I would like to thank Dr. Donald Crouch and Dr. Carolyn Crouch, for providing me this opportunity to work with them and their remarkable support, suggestions and precious feedback.

I would also like to thank my seniors Chaitanya Polumetla, Pavan Poluri, Dinesh Bhirud, and Varun Sudhakar for sharing their knowledge and invaluable support.

I would also like thank my co-workers Ramakrishna Cherukuri, Abhijeet Mahule, and Sridhar Uppala for being very friendly, helpful, and for their support throughout this work.

I would also like to thank Lori Lucia and Jim Luttinen for their invaluable support.

Abstract

Information retrieval (IR) is the science of retrieving information associated with a given query that is judged relevant by the user. With the use of XML, a mechanism was developed to identify the structure of a document, enabling the retrieval of elements from within documents. Now we can retrieve relevant elements at different levels of granularity. We use *flexible retrieval* to retrieve elements from a document [3].

The goal of this thesis is to improve the results of the INEX Focused and Relevant in Context (RIC) tasks. In the Focused task, we are required to produce a rank ordered list of non-overlapping elements, whereas in the RIC task, we are required to retrieve relevant focused elements from relevant articles. In this thesis, we discuss various methodologies that we have developed to improve our results for the Focused and RIC tasks. Experiments demonstrating the efficacy of our methods are detailed herein.

Table of Contents

List of Tables.....	iv
List of Figures.....	v
1. Introduction.....	1
2. Overview.....	3
2.1 INEX.....	3
2.2 2009 Retrieval Tasks.....	7
2.3 Evaluation Measures.....	9
2.4 Smart Retrieval Engine.....	11
3. Background.....	12
4. Experiments and Analysis.....	27
4.1 Focused Retrieval Methodology.....	27
4.2 Strategies for Overlap Removal.....	28
4.3 Rearrangement of Focused Output.....	31
4.4 RIC Task.....	32
4.5 Experiments.....	32
4.6 Analysis.....	37
5. Conclusion and Future Work.....	39
6. References.....	40

List of Tables

Table 1: Details of query sub fields [5].....	6
Table 2: iP[0.01] Focused Retrieval - Child Strategy 2009.....	33
Table 3: iP[0.01] Focused Retrieval - Section Strategy 2009.....	34
Table 4: iP[0.01] Focused Retrieval - Correlation Strategy 2009.....	34
Table 5: iP[0.01] RIC - Child Strategy 2009.....	35
Table 6: iP[0.01] RIC - Section Strategy 2009.....	36
Table 7: iP[0.01] RIC - Correlation Strategy 2009.....	36
Table 8: Top 10 Ranked Universities in the Focused Task [2].....	37
Table 9: Top 10 Ranked Universities in the RIC Task [2].....	38

List of Figures

Figure 1: Sample Document, Document ID: 6005.xml (2009).....	5
Figure 2: Sample Query, Query ID: 2009114 (2009) [5].....	6
Figure 3: Formula to Calculate MAiP[1].....	9
Figure 4: Formula to Calculate iP[x] [1].....	10
Figure 5: Step-by-step Procedure of Element Retrieval Process.....	13
Figure 6: Example Showing Nesting of Sub-sections in a Document.....	15
Figure 7: Sample XML Document.....	16
Figure 8: Article Parse of the Sample Document.....	16
Figure 9: Section Parse of the Sample Document.....	17
Figure 10: Sub-Section Parse of the Sample Document.....	17
Figure 11: Paragraph Parse of the Sample Document.....	17
Figure 12: <i>Para + mt</i> Parse of the Sample Document.....	18
Figure 13: An Example Doctree.....	20
Figure 14: Sample Output of Flex.....	22
Figure 15: Sample Output in TREC Format.....	23
Figure 16: Sample Document Containing Elements Between Article and Body....	24
Figure 17: Sample Patched Xpath.....	24
Figure 18: Patched Version of the Sample Output.....	25
Figure 19: Sample Output of the INEX Evaluation Tool.....	26
Figure 20: Sample Output of Child Strategy.....	28

Figure 21: Sample Output(1) of Section Strategy.....	29
Figure 22: Sample Output(2) of Section Strategy.....	30
Figure 23: Sample Output(3) of Section Strategy.....	30
Figure 24: Sample Output of Correlation Strategy.....	31

1. Introduction

Information retrieval (IR) is the science of retrieving information associated with a given query that is judged relevant by the user. Before the introduction of the World Wide Web (WWW), the focus of IR was on retrieving documents. But, after the introduction of the WWW and Extensible Markup Language (XML), a method that finds structures in an electronic document was developed. The extensible markup of a document provides the underlying structure of its elements like sections, paragraphs, etc., which facilitates retrieval more specific to the query.

Even though XML documents are supposed to contain structured elements, recent changes to XML have resulted in the addition of untagged text within XML documents. The initial design of our flexible retrieval system (Flex) was unable to handle the semi-structured documents, but it was later modified to handle them as well. Flexible retrieval is dynamic (elements are retrieved dynamically, at run time) [3]. We use the Vector Space Model [11] for dynamic element retrieval. In flexible retrieval, we represent a semi-structured document in the form of a tree, in which each paragraph is considered to be a leaf node.

INEX, the Initiative for the Evaluation of XML Retrieval [5], conducts a competition for the development and evaluation of XML based retrieval systems. Every year, INEX provides a document collection, a set of topics, and an assessment package to the participants. A goal of this thesis is to improve the results of the INEX Focused and Relevant in Context Tasks. In the Focused Task, a rank-ordered list of

non-overlapping elements has to be returned for each query. For this thesis, we have worked on the 2009 Wikipedia document collection provided by INEX. The major difference between 2008 document collection and the 2009 document collection is the inclusion of sub-sections in the semi-structured documents. This modification in the document collection required changes to our retrieval process in order to improve the results.

2. Overview

This chapter gives an overview of INEX (its document collection, query set, and tasks) and the Smart search engine.

2.1 INEX

INEX, the Initiative for the Evaluation of XML Retrieval [5], is an organization that conducts competitions for the development and evaluation of effective XML information retrieval techniques. It has different tracks (e.g., Ad Hoc, Book, Efficiency, Data-Centric, Interactive, Link-the-Wiki, XML-Mining, and Question Answering). The University of Minnesota, Duluth participates only in the Ad Hoc track, which contains three tasks: Focused, Relevant in Context (RIC), and Best in Context (BIC). In 2009, our work was limited to the Focused and Relevance in Context tasks. In the Focused task, we have to produce non-overlapping (focused) elements that are relevant to the query, whereas in the Relevance in Context task, we initially produce relevant articles and then extract relevant elements from those articles. INEX provides the document collection, set of topics, and assessment package to evaluate the results.

Document Collection

The 2009 Wikipedia document collection is approximately 50 GB in size, and it contains around two million XML documents distributed over 1000 directories. Each document in the 2009 collection (i.e., XML document) contains primarily text enclosed within tags. The collection is semi-structured as some documents contain untagged text. The collection has over 30,000 unique tags in it. The 2010 document collection is identical to the 2009 collection (i.e., the same collection is used for both years). A sample document from the 2009 collection is shown in Figure 1.

Topics

Each INEX participant group is responsible for submitting a set of queries, which are used by INEX to generate the final query set. The queries are in Content Only + Structure (CO + S) format with the fields as shown in Table 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE article SYSTEM "../article.dtd">
<article xmlns:xlink="http://www.w3.org/1999/xlink">
<header>

<title>Communications in Comoros</title>

<id>6005</id>

<revision>
<id>231932955</id>
<timestamp>2008-08-14T16:53:09Z</timestamp>
<contributor>
<username>Blofeld of SPECTRE</username>
<id>1616157</id>
</contributor>
</revision>
<categories>
<category>Communications in Comoros</category>
</categories>
</header>

<bdy>

<p>

<b>Communications in <country wordnetid="108544813"
confidence="0.9508927676800064">
<link xlink:type="simple" xlink:href="../403/5403.xml">
Comoros</link></country>
</b>

</p>
...
</article>
```

Figure 1: Sample Document, Document ID: 6005.xml (2009)

Table 1: Details of query sub fields [5]

Field	Description
Title	Content Only (CO) queries are given here
Castitle	Content and Structure (CAS) queries are given here
Description	A brief description of information need written in natural language
Narrative	A detailed description of information need and the description of what makes an element relevant or not

There are a total of 115 queries for the 2009 collection. A sample query is shown in Figure 2.

```

<topic id="2009114" ct_no="310">
<title>self-portrait</title>
<castitle>//painter//figure[about(../caption, self-portrait)]</castitle>
<phrasetitle>"self portrait"</phrasetitle>
<description>Find self-portraits of painters.</description>
<narrative>
I am studying how painters visually depict themselves in their work. Relevant
document components are images of works of art, in combination with sufficient
explanation (i.e., a reference to the artist and the fact that the artist him/herself is
depicted in the work of art). Also textual descriptions of these works, if sufficiently
detailed, can be relevant. Document components discussing the portrayal of artists in
general are not relevant, as are artists that figure in painters of other artists.
</narrative>
</topic>

```

Figure 2: Sample Query, Query ID: 2009114 (2009) [5]

Relevance Assessments

Every year, INEX provides relevance assessments (which helps in evaluating the results of the participants). INEX also provides a tool, GPXRai [5], which helps to produce the relevance assessments. The GPXRai tool allows the user to highlight text that is relevant to a particular query. INEX creates an assessment pool by merging all the assessments from all the participants. INEX converts this highlighted text into File Offset and Length (FOL) format to evaluate the results [5].

2.2 2009 Retrieval Tasks

The following tasks are included in the 2009 Ad Hoc track:

Through Task

In the Through task, we return a ranked list of elements that are relevant to the given query. It may contain overlapped elements. The primary goal of the Thorough task is to return all the relevant elements within the document.

Focused Task

We must return a ranked list of non-overlapping elements for this task [5]. In an XML document, if a child element (like paragraph) is found to be relevant to a particular query, then its parent elements (like sub-section or section or body) are relevant to that query to some extent. The primary goal of the focused task is to identify all these elements and return only the most relevant element. This process is called *overlap removal*, which is explained in detail in Chapter 3.

Relevant in Context Task

In this task, we also return non-overlapping elements, but the elements must be grouped by document [5]. This task requires the retrieval of focused elements from articles that correlate highly with the query [8]. For this task, we initially identify the articles that highly correlate with the query and then Flex is used to retrieve elements from those articles.

Best in Context Task

In this task, we first identify the articles that highly correlate with the query and then the Best Entry Point (BEP) for each of those articles is determined. The BEP is the best point in the article for the user to begin reading, in order to find

information relevant to the query. We only have one BEP per document, and most of the time, it is found at beginning of the document. In 2009, we centered our investigations on the Focused and RIC tasks.

2.3 Evaluation Measures

Different metrics are used to evaluate the different tasks [5].

Thorough Task

For the evaluation of the results in the Thorough Task, the mean average interpolated precision (MAiP) is used. See [5] for details. The formula used to calculate MAiP is shown in Figure 3.

If there are n topics:

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

where, AiP (average interpolated precision) is calculated as:

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

Here, $iP[x]$ is interpolated precision at recall x (see Figure 4).

Figure 3: Formula to Calculate MAiP [1]

Focused Task

For the evaluation of the results in the Focused Task, a metric called interpolated precision at 1% recall (i.e., $iP[0.01]$) is used. *Precision* is defined as fraction of retrieved text that is highlighted, and *recall* is defined as the fraction of highlighted text that is retrieved [5]. See [5] for more details. The formula to calculate $iP[x]$ is shown in Figure 4.

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

where L_q is the rank ordered list of elements,

$|L_q|$ is the length of the ranked list,

$P[r]$ is the precision and $R[r]$ is the recall, at rank r , and

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

Figure 4: Formula to Calculate $iP[x]$ [1]

Relevant in Context Task

In this task, the evaluation is done based on the overall performance estimate [5]. The metric used for this task is called mean average generalized precision (MAgP). See [5] for details.

2.4 Smart Retrieval Engine

The retrieval engine we use is Smart 13.0 [10]. Smart uses the Vector Space Model [11]. In this model, the documents and queries are denoted as weighted term vectors. In the vector space, the correlation of the document with the query can be found using the distance between the query vector and the document vector [11]. Smart produces the fundamental retrieval functionalities like the indexing of documents, weighting of document and query vectors, and retrieval of rank-ordered elements (documents) [7].

3. Background

This chapter gives an overview of our methods for flexible retrieval, and its use in the Focused and Relevance in Context tasks. We begin with the processing of the document collection and then retrieve elements that are highly correlated with each query. A step-by-step description of the process is shown in Figure 5.

Scrubbing the 2009 document collection

The 2009 document collection contains around 30,000 unique tags. Most of those tags are unwanted; such tags are removed from each XML document. While this process is going on, we ensure that each XML tag remaining in the document has a matching closing XML tag. The resultant collection is suitable to be given as input for parsing.

Parsing the collection

In the parsing stage, we produce five kinds of parses: article parse, paragraph parse, *para + mt* parse, section parse, and sub-section parse.

- 1) The article parse outputs only the text enclosed within article tags.

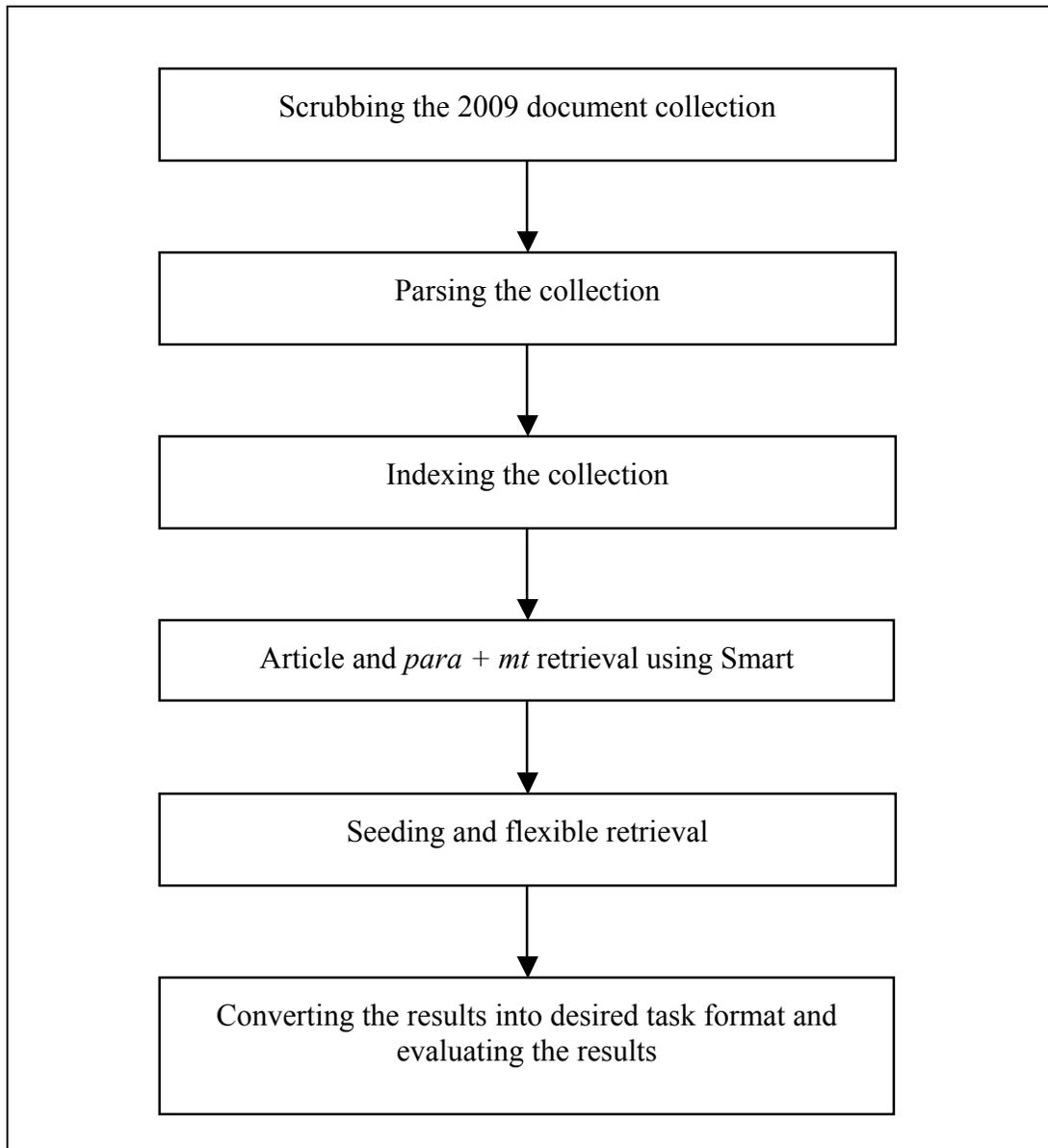


Figure 5: Step-by-step Procedure of Element Retrieval Process

2) The term *paragraph* represents a set of terminal nodes. This set of terminal nodes may be selected based upon the collection and the set of tags of interest. For example, for the 2009 document collection, the set of terminal nodes are p

(paragraph), entry, list, table, etc. The paragraph parse returns elements containing the text enclosed in each terminal node.

3) The untagged text in the document is referred to as “magic text” (*mt*). We do not ignore magic text because Flex must generate the entire document tree. The *para + mt* parse is similar to a paragraph parse, but it also returns the magic text as a separate element (which is enclosed in `<mt>` tags).

4) The section parse returns as an element the text that is enclosed in each section (as identified by `<sec>` tag).

5) The major difference between the 2008 and 2009 collections is the inclusion of sub-sections in the 2009 document collection. The maximum depth of a sub-section in the document collection is four, i.e., subsections may only be nested four levels deep. The tags used to recognize these levels are: `<ss1>`, `<ss2>`, `<ss3>`, and `<ss4>`. If the subsection is present at a depth of one, we should use the tag `<ss1>`; if the subsection is present at a depth of two, we should use `<ss2>`, and so forth. Figure 6 shows an example illustrating the nesting of sub-sections. For example, in Figure 6, if we have a second sub-section 1 (ss1) inside the section, then it is represented as `ss1[2]` (which represents it as the second subsection inside its parent element).

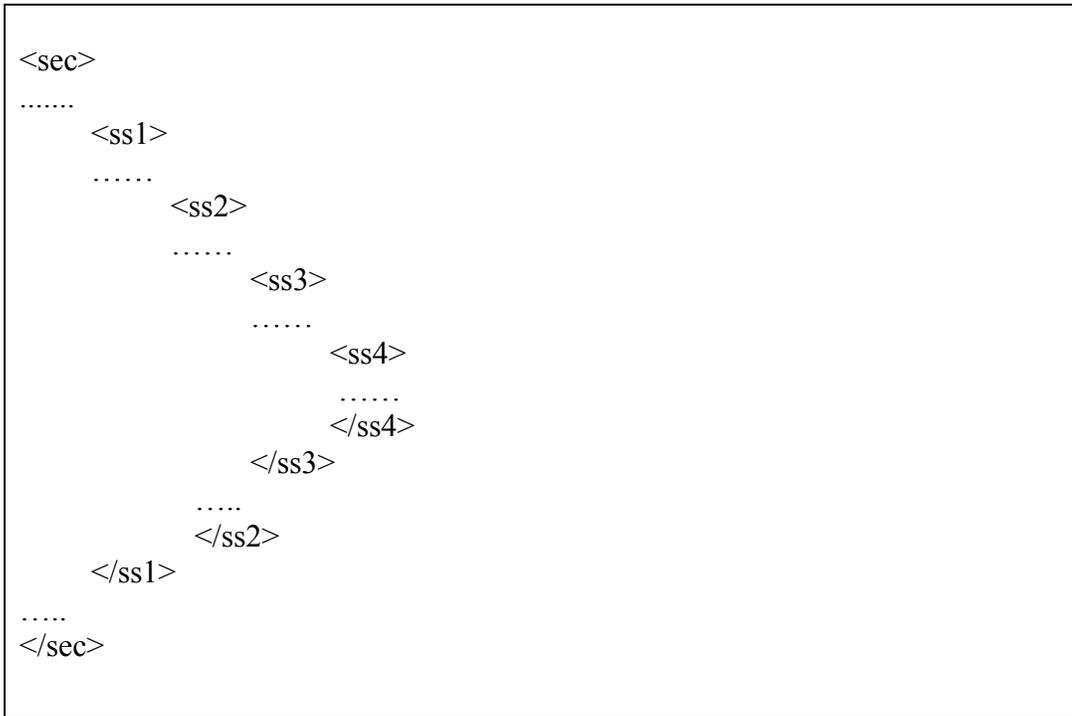


Figure 6: Example Showing Nesting of Sub-sections in a Document

The sub-section parse returns elements containing text enclosed in each sub-section tag. Figure 7 shows a sample document containing sections, sub-sections, paragraphs, and magic text. The article parse for the sample document shown in Figure 7 is shown in Figure 8. The corresponding *section*, *sub-section*, *paragraph*, and *para + mt* parses are shown in Figures 9-12.

The section parse for the sample document would produce the element shown in Figure 9. The sub-section parse would produce the elements as shown in Figure 10. The paragraph parse would produce the elements as shown in Figure 11. The *para + mt* parse would produce the elements as shown in Figure 12.

```
<article>
<bdy>
Sachin Tendulkar
<sec>
<ss1>
<p>
Sachin Tendulkar started his career in 1989.
</p>
<ss2>
<p>
He is born in Mumbai, India.
</p>
</ss2>
</ss1>
</sec>
</bdy>
</article>
```

Figure 7: Sample XML Document

```
Sachin Tendulkar Sachin Tendulkar started his career in 1989. He is born in
Mumbai, India.
```

Figure 8: Article Parse of the Sample Document

```
<sec>
Sachin Tendulkar started his career in 1989. He is born in Mumbai, India.
</sec>
```

Figure 9: Section Parse of the Sample Document

```
<ss1>
Sachin Tendulkar started his career in 1989. He is born in Mumbai, India.
</ss1>
<ss2>
He is born in Mumbai, India.
</ss2>
```

Figure 10: Sub-Section Parse of the Sample Document

```
<p>
Sachin Tendulkar started his career in 1989.
</p>
<p>
He is born in Mumbai, India.
</p>
```

Figure 11: Paragraph Parse of the Sample Document

```
<p>
Sachin Tendulkar started his career in 1989.
</p>

<p>
He is born in Mumbai, India.
</p>

<mt>
Sachin Tendulkar
</mt>
```

Figure 12: *Para + mt* Parse of the Sample Document

Indexing the collection

Smart produces indexes of the parses produced in the previous step. Indexing is performed for each of these parses: *article*, *section*, *sub-section*, *paragraph*, and *para + mt*. The input for each indexing is the corresponding parse; for example, article parse is given as input for article indexing. Each indexing produces term frequency vectors (nnn-weighted vectors), the inverted file (inv.words), the dictionary (which contains unique words), and the textloc file.

Article and *Para + mt* Retrieval

We use Smart for the article and *para + mt* retrievals, and all the vectors are weighed using *Lnu-ltu* [12] term weighting. In the retrieval process, Smart produces a ranked list of elements (i.e., articles, paragraphs, etc.). For each set of vectors, we have to determine the slope and pivot values before we continue with the retrieval process. See [12] for more details on *Lnu-ltu* weighting.

Seeding and Flexible Retrieval

Seeding

Before the seeding takes place, we have to produce the doctrees for each document. A doctree is nothing more than a preorder traversal of a document tree. A doctree contains the Xpaths for each element in the document. An example of a doctree is shown in Figure 13.

The first column in Figure 13 represents the Xpath to the element; the second represents the number of children of that element. If the third column is zero, then the element is not a leaf node, and if it is one, the element is a leaf node.

/article[1]	1	0
/article[1]/bdy[1]	5	0
/article[1]/sec[1]	1	0
/article[1]/sec[1]/ss[1]	3	0
/article[1]/sec[1]/ss[1]/p[1]	0	1
/article[1]/sec[1]/ss[1]/p[2]	0	1
/article[1]/sec[1]/ss[1]/p[3]	0	1

Figure 13: An Example Doctree

Each element in every Xpath has a number inside square braces (e.g., article[1]); this number represents the number of occurrences of that particular element inside its parent element. For example, in the doctree shown in Figure 13, sub-section 1 (ss1) is the parent element of three paragraphs. So, they are represented as p[1], p[2], and p[3], where p[1] indicates that it is the first paragraph in sub-section 1, p[2] indicates the second paragraph in sub-section 1, and so forth.

After the doctrees are generated, we must make sure that each *para + mt* element from every top-ranked document is retrieved. To do this, we perform a very large *para + mt* retrieval, and the output is fed to the seeding script. The doctrees are also given as input to the seeding script. Once provided its input, the seeding script populates all the terminal nodes in a tree (which means that it provides the content for each terminal node). This ensures that Flex generates complete trees.

In the case of Focused retrieval, we combine the *para + mt* retrieval with article retrieval. That is, we select the trees corresponding to each document that is retrieved in article retrieval. This set of trees is called the seed subset [9] (These are the trees of interest).

Flexible Retrieval

We use Flex for the flexible retrieval of elements. This step is done after seeding, and the input for Flex is the seed subset generated in the previous step. In the seeding step, all the terminal nodes are populated, and these terminal nodes are used by Flex to generate the parent nodes. In this process, Flex calculates the scores (correlation) for each terminal node with the query, and then it builds the parent element and calculates the score for the parent with each query. When Flex has generated the score for the root node, this process terminates.

After building every doctree, Flex generates a rank ordered list of elements for each query (during this process, Flex removes the *mt* elements that are inserted during parsing). The Config file for Flex includes the paths to the indexes, slope and pivot values, and the number of output elements. Figure 14 shows a sample output of Flex.

1	21010	/article[1]/bdy[1]/sec[1]	29.7164
1	158543	/article[1]/bdy[1]/sec[4]/p[1]	29.4613
1	264768	/article[1]/bdy[1]/p[2]	29.1159
1	17947	/article[1]/bdy[1]/sec[3]/ss1[1]	29.8736
1	68546	/article[1]/bdy[1]/sec[2]	28.4632
1	82618	/article[1]/bdy[1]/sec[1]/p[4]	27.9731
1	542713	/article[1]/body[1]/p[1]	27.1262

Figure 14: Sample Output of Flex

Conversion of Flex Output to Desired Task Format

As discussed in Chapter 2, we participate in both the Focused and Relevant in Context tasks. After Flex generates its output, it must be converted into the desired task format (this is explained in detail in Chapter 4). Once we get the output from the desired task, it must be converted into TREC format [5] for evaluation (because the INEX evaluation tool does not recognize our output format). Figure 15 shows the sample output after the conversion takes place.

The first column in Figure 15 represents the query number, the second is a dummy column, the third is the document ID, the fourth is the ranking, the fifth represents reverse ranking, the sixth is the output document ID, and the last column represents the Xpath to the element.

```
2009001 Q0 19679192 1 4000 UMD_FOCUSED /article[1]/bdy[1]/sec[4]/p[1]
2009001 Q0 19653466 2 3999 UMD_FOCUSED /article[1]/bdy[1]/sec[4]/p[3]
2009001 Q0 14665819 3 3998 UMD_FOCUSED /article[1]/bdy[1]/sec[1]
2009001 Q0 21201 4 3997 UMD_FOCUSED /article[1]/bdy[1]/sec[4]
2009001 Q0 2829505 5 3996 UMD_FOCUSED /article[1]/bdy[1]/sec[4]/ss1[1]
2009001 Q0 1528075 6 3995 UMD_FOCUSED /article[1]/bdy[1]/sec[1]
2009001 Q0 50718 7 3994 UMD_FOCUSED /article[1]/bdy[1]/sec[6]/p[1]
2009001 Q0 21598 8 3993 UMD_FOCUSED /article[1]/bdy[1]/sec[7]
2009001 Q0 36845 9 3992 UMD_FOCUSED /article[1]/bdy[1]/sec[7]/p[2]
2009001 Q0 15121040 10 3991 UMD_FOCUSED /article[1]/bdy[1]/sec[2]/p[1]
```

Figure 15: Sample Output in TREC Format

Patching and Evaluating the Results

Before the results are given to the INEX evaluation tool, the Xpaths must be patched. This is because the root node of the tree that is generated by Flex is *article* and its child is always *body*. However, some articles have intermediate tags present between *article* and *body*, which must be present in the Xpath for proper evaluation to take place. These elements are not shown in the Flex output because we removed all these tags while cleaning the collection. Since the Xpaths without these tags are not recognizable by the evaluation tool, the Xpaths must be patched.

Figure 16 shows an example that contains tags between article and body. For the document shown in Figure 16, the Xpath is as shown in Figure 17. The patched version of the sample output shown in Figure 15 is shown in Figure 18.

```
<article>
<movie>
<person>
<title>List of Kuruba films</title>
<bdy>
....
</bdy>
</article>
```

Figure 16: Sample Document Containing Elements Between Article and Body

```
Xpath - /article[1]/movie[1]/person[1]/bdy[1]/.....
```

Figure 17: Sample Patched Xpath

```
2009001 Q0 19679192 1 4000 UMD_FOCUSED /article[1]/laureate[1]/bdy[1]/sec[4]/p[1]
2009001 Q0 19653466 2 3999 UMD_FOCUSED /article[1]/laureate[1]/bdy[1]/sec[4]/p[3]
2009001 Q0 14665819 3 3998 UMD_FOCUSED /article[1]/bdy[1]/sec[1]
2009001 Q0 21201 4 3997 UMD_FOCUSED /article[1]/organization[1]/award[1]/bdy[1] /sec[4]
2009001 Q0 2829505 5 3996 UMD_FOCUSED /article[1]/bdy[1]/sec[4]/ss1[1]
2009001 Q0 1528075 6 3995 UMD_FOCUSED /article[1]/bdy[1]/sec[1]
2009001 Q0 50718 7 3994 UMD_FOCUSED /article[1]/symbol[1]/signal[1]/bdy[1]/sec[6]/p[1]
2009001 Q0 21598 8 3993 UMD_FOCUSED /article[1]/bdy[1]/sec[7]
2009001 Q0 36845 9 3992 UMD_FOCUSED /article[1]/entity[1]/bdy[1]/sec[7]/p[2]
2009001 Q0 15121040 10 3991 UMD_FOCUSED /article[1]/bdy[1]/sec[2]/p[1]
```

Figure 18: Patched Version of the Sample Output

Once the patching is done, the output is converted into FOL (File-Offset-Length) [5] format using the software given by INEX. The output, in FOL format, is then fed to the evaluation tool. The evaluation tool then returns the result, evaluated using the desired metric in accordance with the task. Figure 19 shows a sample output of the evaluation tool.

```
<eval run-id="UMD_FOCUSED" file="UMD_FOCUSED_1.txt">
num_q      all      68
num_ret    all      75985
num_rel    all      4858
num_rel_ret all      3665
ret_size   all      108728168
rel_size   all      18838137
rel_ret_size all      5467308
iP[0.00]  all      0.6796786975452944
iP[0.01]  all      0.6332981519389856
iP[0.05]  all      0.5005782764908357
iP[0.10]  all      0.409541110469809
MAiP      all      0.1840738573598488
```

Figure 19: Sample Output of the INEX Evaluation Tool

4. Experiments and Analysis

The experiments performed on the 2009 document collection are discussed in detail in this chapter.

4.1 Focused Retrieval Methodology

Two methodologies have been developed for Focused Retrieval: *Upper Bound Strategy* and *Exact Strategy* [1, 9]. For each query, Flex gives us m rank-ordered elements from n documents, where m takes on the values 50, 100, 150, 200...1500 and n ranges from 50, 100, 200...500.

Once Flex produces its output, the *mts* are removed. In the *Upper Bound Strategy*, m represents the upper bound on the number of elements that are produced (see [1] for more detail). The *Exact strategy* is designed to retrieve exactly m focused elements. To accomplish this, we retrieve a large number of elements (as compared to m) to ensure that the output has at least m elements in it (after overlapping elements are removed) [9]. See [9] for details on *Exact Strategy*. For our experiments, we prefer to use the *Exact Strategy* instead of the *Upper Bound Strategy*, based on experiments done on the 2008 document collection [4].

4.2 Strategies for Overlap Removal

To convert the output of Flex into focused format, we need to eliminate overlapping elements from Flex output. There are three strategies for this: child, section, and correlation strategy [9].

Child Strategy

In the child strategy, the child element is given preference compared to the parent element. For example, given two overlapping elements, where one element is the child of the other element; then we choose the child element rather than the parent element (even if the child element has a lower correlation score than the parent element). Figure 20 shows a sample output of the child strategy.

```
1 18456 /article[1]/bdy[1]/sec[1] 42.13
.
.
1 18456 /article[1]/bdy[1]/sec[1]/p[2] 34.46
Focused output is:
1 18456 /article[1]/bdy[1]/sec[1]/p[2]
```

Figure 20: Sample Output of Child Strategy

Section Strategy

In the section strategy, we prefer the highest correlating element as long as it is not a body element. The parent is preferred to the child as long as the parent has a higher correlation score, and it is not a body element [9]. The child element is preferred if it has a higher correlation score. Some sample outputs are shown in Figures 21, 22, and 23. In Figure 21, section (sec[1]) is preferred to paragraph (p[2]) because section is the parent element and has a higher correlation score. In Figure 22, paragraph is preferred to section because paragraph is the child element and it has the higher correlation score. In Figure 23, we choose the child element even though it has the lower correlation score because the parent is a body.

```
1 18456 /article[1]/bdy[1]/sec[1] 46.53
.
.
1 18456 /article[1]/bdy[1]/sec[1]/p[2] 34.46
Focused output is:
1 18456 /article[1]/bdy[1]/sec[1]
```

Figure 21: Sample Output(1) of Section Strategy

```
1 18456 /article[1]/bdy[1]/sec[1] 48.53
.
.
1 18456 /article[1]/bdy[1]/sec[1]/p[2] 49.46
Focused output is:
1 18456 /article[1]/bdy[1]/sec[1]/p[2]
```

Figure 22: Sample Output(2) of Section Strategy

```
1 18456 /article[1]/bdy[1] 46.53
.
.
1 18456 /article[1]/bdy[1]/sec[1]/p[2] 34.46
Focused output is:
1 18456 /article[1]/bdy[1]/sec[1]/p[2]
```

Figure 23: Sample Output(3) of Section Strategy

Correlation Strategy

In this strategy, the element having the highest correction score is preferred. Figure 24 shows a sample output of the correlation strategy. Here, section is preferred to paragraph because it has the higher correlation score.

```
1 18456 /article[1]/bdy[1]/sec[1] 48.53
.
.
1 18456 /article[1]/bdy[1]/sec[1]/p[2] 44.46
Focused output is:
1 18456 /article[1]/bdy[1]/sec[1]
```

Figure 24: Sample Output of Correlation Strategy

4.3 Rearrangement of Focused Output

Experiments on the 2008 document collection determined that we get better results when the output is rearranged in document order. See [1, 9] for details. Using the *Exact Strategy*, two kinds of rearrangement are possible: *rearrangement after chopping* (RAC) and *rearrangement before chopping* (RBC).

In the *rearrangement after chopping* strategy, the top m elements of the focused output are chosen to be rearranged. In the *rearrangement before chopping* strategy, the huge set of Focused elements are rearranged initially, and then the top m elements are chosen [9]. For our experiments, we use the RBC strategy.

4.4 RIC Task

The goal of this task is to first find relevant articles and then to find relevant elements in those articles. See [8] for details. In our experiments, we use the same identical process that we described earlier for Focused retrieval. We also use the same strategies (*child*, *section*, and *correlation*) for removing overlapping elements and RBC for rearrangement.

4.5 Experiments

All our experiments are done on the 2009 document collection and evaluated with the 2009 evaluation package. For Tables 2-7, the column headings represent the number of elements retrieved and the row headings represent the number of documents that are used for retrieval. Experiments 1-3 represent the experiments done for Focused Retrieval, and Experiments 4-6 represent the experiments done for Relevant in Context retrieval.

Focused Task Experiments

Experiment 1 – Focused Task

We used child strategy to remove the overlapping elements in this experiment.

Table 2 shows the results of this Focused retrieval experiment.

Table 2: iP[0.01] Focused Retrieval - Child Strategy 2009

	50	100	150	200	250	500	1000	1500
25	0.6351	0.6377	0.6428	0.6428	0.6428	0.6428	0.6428	0.6428
50	0.6352	0.6379	0.6430	0.6452	0.6464	0.6464	0.6464	0.6464
100	0.6351	0.6379	0.6430	0.6454	0.6464	0.6472	0.6472	0.6472
150	0.6351	0.6379	0.6431	0.6454	0.6464	0.6476	0.6478	0.6478
200	0.6353	0.6378	0.6431	0.6454	0.6467	0.6476	0.6479	0.6481
250	0.6462	0.6379	0.6430	0.6453	0.6467	0.6476	0.6479	0.6482
500	0.6352	0.6379	0.6430	0.6453	0.6465	0.6475	0.6480	0.6482

Experiment 2 – Focused Task

We used section strategy to remove the overlapping elements in this experiment. Table 3 shows the results of this Focused retrieval experiment.

Table 3: iP[0.01] Focused Retrieval - Section Strategy 2009

	50	100	150	200	250	500	1000	1500
25	0.6462	0.6489	0.6539	0.6539	0.6539	0.6539	0.6539	0.6539
50	0.6465	0.6493	0.6539	0.6546	0.6557	0.6557	0.6557	0.6557
100	0.6465	0.6492	0.6537	0.6544	0.6556	0.6573	0.6573	0.6573
150	0.6463	0.6494	0.6539	0.6545	0.6558	0.6582	0.6585	0.6585
200	0.6459	0.6493	0.6538	0.6546	0.6556	0.6581	0.6589	0.6593
250	0.6464	0.6493	0.6538	0.6546	0.6556	0.6581	0.6587	0.6594
500	0.6465	0.6490	0.6539	0.6546	0.6557	0.6580	0.6588	0.6594

Experiment 3 – Focused Task

Correlation strategy was used to remove the overlapping elements in this experiment. Table 4 shows the results of this Focused retrieval experiment.

Table 4: iP[0.01] Focused Retrieval - Correlation Strategy 2009

	50	100	150	200	250	500	1000	1500
25	0.6374	0.6392	0.6418	0.6431	0.6431	0.6431	0.6431	0.6431
50	0.6375	0.6394	0.6421	0.6442	0.6442	0.6442	0.6442	0.6442
100	0.6375	0.6394	0.6423	0.6445	0.6459	0.6459	0.6459	0.6459
150	0.6374	0.6396	0.6423	0.6445	0.646	0.6469	0.6469	0.6469
200	0.6374	0.6396	0.6423	0.6445	0.6461	0.647	0.6484	0.6484
250	0.6474	0.6396	0.6423	0.6445	0.6461	0.647	0.6487	0.6488
500	0.6475	0.6395	0.6422	0.6445	0.6459	0.6471	0.6487	0.6488

RIC Experiments

Experiment 4 - RIC

We used child strategy to remove the overlapping elements in this experiment.

Table 5 shows the results of this Relevance in Context experiment.

Table 5: MAgP RIC - Child Strategy 2009

	50	100	150	200	250	500	1000	1500
25	0.1094	0.1126	0.1146	0.1146	0.1146	0.1146	0.1146	0.1146
50	0.1123	0.1178	0.1195	0.1221	0.1221	0.1221	0.1221	0.1221
100	0.1131	0.1295	0.1328	0.1395	0.1461	0.1469	0.1469	0.1469
150	0.1131	0.1386	0.1471	0.1476	0.1521	0.1504	0.1513	0.1513
200	0.1131	0.1503	0.1496	0.1501	0.1543	0.1514	0.1552	0.1558
250	0.1131	0.1625	0.1542	0.1565	0.1579	0.1605	0.1633	0.1651
500	0.1131	0.1657	0.1602	0.1616	0.1630	0.1654	0.1675	0.1689

Experiment 5 - RIC

We used section strategy to remove the overlapping elements in this experiment. Table 6 shows the results of this Relevance in Context experiment.

Table 6: MAgP RIC - Section Strategy 2009

	50	100	150	200	250	500	1000	1500
25	0.1076	0.1104	0.1129	0.1129	0.1129	0.1129	0.1129	0.1129
50	0.1098	0.1138	0.1152	0.1179	0.1179	0.1179	0.1179	0.1179
100	0.1102	0.1224	0.1268	0.1292	0.1331	0.1358	0.1358	0.1358
150	1103	0.1341	0.1392	0.1427	0.1488	0.1496	0.1501	0.1501
200	0.1103	0.1462	0.1444	0.1494	0.1505	0.1509	0.1523	0.1529
250	0.1103	0.1573	0.1521	0.1542	0.1536	0.1552	0.1571	0.1589
500	0.1103	0.1619	0.1593	0.1587	0.1594	0.1606	0.1627	0.1636

Experiment 6 - RIC

We used correlation strategy to remove the overlapping elements in this experiment. Table 7 shows the results of this Relevance in Context experiment.

Table 7: MAgP RIC - Correlation Strategy 2009

	50	100	150	200	250	500	1000	1500
25	0.1124	0.1154	0.1186	0.1154	0.1154	0.1154	0.1154	0.1154
50	0.1165	0.1201	0.1245	0.1256	0.1256	0.1256	0.1256	0.1256
100	0.1177	0.1386	0.1404	0.1445	0.151	0.1472	0.1472	0.1472
150	0.1178	0.1492	0.1511	0.1562	0.1595	0.1522	0.1531	0.1531
200	0.1179	0.1577	0.1548	0.1588	0.1627	0.1614	0.1643	0.1654
250	0.1179	0.1656	0.1594	0.1621	0.1665	0.1651	0.1695	0.1696
500	0.1179	0.1699	0.1613	0.1645	0.1692	0.1687	0.1722	0.1731

4.6 Analysis

For Focused Retrieval, section strategy produced better results when compared to other strategies. But, for Relevance in Context, correlation strategy produced better results. For Focused Retrieval, our top result (0.6594) was not statistically significant when compared with the results produced by the top ten universities listed by INEX for the Focused Retrieval task [6]. But for the Relevance in Context task, our top result (0.1731) was statistically significant compared to the universities ranked 6-10 (listed by INEX for the Relevance in Context task [6]) but not statistically significant compared to the results produced by the universities that are ranked 1-4. See Tables 8 and 9, and [2] for details.

Table 8: Top 10 Ranked Universities in the Focused Task [2]

Participant ID	iP[0.01]	Rank	Institute
P72 (Section Strategy)*	0.6594	-	UMD**
P72 (Correlation Strategy)*	0.6488	-	UMD**
P72 (Child Strategy)*	0.6482	-	UMD**
P78	0.6333	1	University of Waterloo
P68	0.6141	2	Univ. Pierre et Marie Curie
P10	0.6134	3	Max-Planck-Institute
P60	0.6060	4	Saint Etienne University
P6	0.5997	5	Univ. of Amsterdam
P5	0.5592	6	Queensland Univ. of Tech.
P16	0.5903	7	Univ. of Applied Science
P48	0.5853	8	LIG
P22	0.5844	9	ENSM - SE
P25	0.4973	10	Renmin Univ. of China

* Our new results

** UMD - University of Minnesota, Duluth

Table 9: Top 10 Ranked Universities in the RIC Task [2]

Participant ID	MAgP	Rank	Institute
P5	0.1885	1	Queensland Univ. of Tech.
P4	0.1847	2	Univ. of Otago
P6	0.1773	3	Univ. of Amsterdam
P48	0.1760	4	LIG
P72 (Correlation Strategy)*	0.1731	-	UMD**
P36	0.1720	5	Univ. of Tampere
P72 (Child Strategy)*	0.1689	-	UMD**
P72 (Section Strategy)*	0.1636	-	UMD**
P346	0.1188	6	Univ. of Twente
P60	0.1075	7	Saint Etienne University
P167	0.1045	8	School of Ele. Engg. & CS
P25	0.1028	9	Renmin Univ. of China
P72	0.0424	10	UMD

* Our new results

** UMD - University of Minnesota, Duluth

5. Conclusion and Future Work

In 2009, best results for Focused retrieval are produced using the *section strategy*. The other two strategies, *child* and *correlation*, also produce good results. The highest score produced using the *section strategy* is **0.6594**, and it is produced by taking 1500 elements from 500 documents. Even though this result is not statistically significant compared to the top ten universities listed by INEX for the Focused task [6], we are able to reach first place with our score. This shows that our retrieval methods are good; combining the article retrieval with element retrieval yields good results for the 2009 document collection. The RIC results are also good; we are able to rank at 5 in the ranked list of Universities provided by INEX for the RIC task.

As we are combining article retrieval with element retrieval, the entire retrieval process depends on the retrieved articles. A better article retrieval could produce more relevant documents and in turn more relevant elements. So the future work includes improving our article retrieval to get more relevant articles (which gives us more scope to retrieve elements with higher correlations). In flexible retrieval, we are doing a very large *para + mt* retrieval to generate each terminal node of all the documents in question. We can improve our method by first retrieving articles and then for each article retrieve all the terminal nodes for that article. This is a more efficient approach.

References

- [1] Bhirud, D. Focused Retrieval Using Upperbound Methodology, MS Thesis, Department of Computer Science, University of Minnesota Duluth, 2009. <http://www.d.umn.edu/cs/thesis/bhirud.pdf>
- [2] Cherukuri, R. Significance Testing for INEX 2008-2009 Ad Hoc Track, MS Thesis, Department of Computer Science, University of Minnesota Duluth, 2009.
- [3] Crouch, C. Dynamic element retrieval in structured environment. *ACM TOIS*, 24(4): 437-454, 2006.
- [4] Crouch, C., Crouch, D., Bhirud, D., Poluri, P., Polumetla, C., Sudhakar, V. A Methodology for Producing Improved Focused Elements. *INEX 2009 Proceedings* (to appear).
- [5] Geva, S., Kamps, J., Trotman, A. INEX 2009 Workshop Pre-proceedings. <http://www.inex.otago.ac.nz/>
- [6] <http://www.inex.otago.ac.nz/tracks/adhoc/results.asp>
- [7] Paranjape, D. Improving Focused Retrieval. MS Thesis. University of Minnesota Duluth, 2008. <http://www.d.umn.edu/cs/thesis/paranjape.pdf>
- [8] Polumetla, C. Improving the Results for the Relevant In Context Task, MS Thesis, Department of Computer Science, University of Minnesota Duluth, 2009. <http://www.d.umn.edu/cs/thesis/polumetla.pdf>
- [9] Poluri, P. Focused Retrieval using Exact Methodology. MS Thesis, University of Minnesota Duluth, 2009. <http://www.d.umn.edu/cs/thesis/poluri.pdf>
- [10] Salton, G. The SMART Retrieval System – Experiments in Automatic Documents Retrieval. *Prentice-Hall, Eaglewood Cliffs, NJ*, 1971.
- [11] Salton, G., Wong A., and Yang C. A Vector Space Model for Information Retrieval. *Journal of American Society for Information Science*, 18(11):613-620, 1975.
- [12] Singhal, A., Buckley, C., Mitra M. Pivoted Document Length Normalization, *Proceedings of 19th Annual International Conference on Research and Development in Information Retrieval, Zurich*. 19-21, 1996.