Parsing the Wiki Collection and Snippet Generation

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Sai Subramanyam Chittilla

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

Dr Donald Crouch

April 2013

Acknowledgements

I would like to acknowledge both Dr Donald Crouch and Dr Carolyn Crouch for providing me this wonderful opportunity to work in the field of IR and the motivation they gave me in delivering this thesis. I would also thank Mr Satayanarayana Murthy Ganpathibhotla, who was mainly instrumental in relishing my dream of doing masters in the field of IR. I would like to thank Dr Joseph Gallian for guiding me in the one of the best math course I have ever taken and for being a part of my thesis committee. I would also thank the UMD computer science department for motivating me. I would like to thank all of my friends and family for their support.

Abstract

Information Retrieval (IR) is a field which deals with retrieving useful information from large sets of data in response to a query. Much information in this digital age is stored in XML format, which associates a structure with a document. Though IR systems have been used for years to access documents, the field has greatly expanded with the emergence of the world wide web, which emphasizes the structure of the data. The amount of data makes the identification of various portion(s) of a document difficult; document structure helps in this task.

This thesis describes a retrieval task known as snippet retrieval. A snippet is the smallest meaningful body of text which can be used to establish the relevance of the document without actually looking at the document. The work on snippet retrieval is extended from past work in focused retrieval, wherein a ranked list of focused elements is retrieved in response to the user query [1]. The Vector Space Model [10] provides the framework for retrieval; we use Smart for basic retrieval functions. Our system for dynamic element retrieval, Flex, enables us to identify and rank the individual elements of each hypertext document with respect to the query. We include a discussion of focusing strategies and the use of focused elements as a basis for snippet generation. Results of our top-ranked 2011 and 2012 Snippet Retrieval track runs are included.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Overview

Information Retrieval (IR) is a field which deals with retrieving useful information from large sets of data in response to a query. Much information in this digital age is stored in XML [6] format, which associates a structure with a document. Though IR systems have been used for years to access documents, the field has greatly expanded with the emergence of the world wide web, which emphasizes the structure of the data. The amount of data makes the identification of various portion(s) of a document difficult; document structure helps in this task

This thesis describes a retrieval task known as snippet retrieval. A snippet is the smallest meaningful body of text which can be used to establish the relevance of the document without actually looking at the document [2]. The work on snippet retrieval is extended from past work in focused retrieval, wherein a ranked list of focused elements is retrieved in response to the user query [1]. The Vector Space Model [10] provides the framework for retrieval.

In the Vector Space Model, documents and queries are represented as vectors. Functions of term frequency and inverse document frequency are traditionally used to weight the document and query vectors. The Smart system [9], which implements the Vector Space Model is used in our research for basic retrieval. A measure such as inner product or cosine is used to calculate the similarity between the document and the query vectors.

1.2 INEX and Web Retrieval

INEX [3], the *Initiative for the Evaluation of XML retrieval*, is an organization which provides a platform for XML retrieval. INEX organizes competitions in various tracks, such as Ad-hoc retrieval, Question-answering, etc. INEX also provides the document collection and topics or queries to be used for that track. Our UMD research group participates in the INEX Snippet Retrieval track. Snippet retrieval derives from the earlier (Ad-hoc) track in Focused Retrieval, which deals with identifying all the focused (i.e, non-overlapping) elements within an XML document. Our current (2012) approach to snippet retrieval is a further refinement of the 2010 Focused track.

XML format, used to provide structure for web documents, enables the retrieval of individual elements of a document. The current INEX document collection is a portion of Wikipedia; all documents are in XML format. The collection has changed over the years but most of the documents maintain in general a common structure, wherein the text enclosed within the *article* tags represents the main element of the document. The Xpath (a string indicating a series of tags from the root to the tag of interest) represents both the name of the element and its location in the document. The Wikipedia collection used for the 2011 Snippet Retrieval task is the same collection released by INEX in 2009. Focused element retrieval was extended to generate snippets for the 2011 topics, and we continued this approach in our 2012 experiments.

Retrieval of documents from the web is not new. It has existed for some years, as evidenced by various commercial search engines. With INEX, the focus on the elements (rather than the entire document) and the XML structure allows both the recognition of an element and its retrieval using the XPath and tags.

The remaining chapters describe background and related work (Chapter 2), processing and parsing of the document collection along with the retrieval of the elements (Chapter 3), 2011 and 2012 snippet track expirements, results and analysis (Chapter 4), and a follow-up discussion and conclusions (Chapter 5). Much of this research

was performed within a team environment. The team consisted of the author and and a fellow graduate student, S. Nagalla, whose work is reported in [7].

2 Background

This chapter provides an overview of Salton's Vector Space Model. We then look at how documents are represented in XML format and the modifications that have occurred in the Wikipedia collection over the years. Finally, we introduce our runtime retrieval system, called Flex.

2.1 The Vector Space Model

Salton's Vector Space Model [10] treats documents et.al., and queries as vectors. Smart [9], the search engine built at Cornell by Buckley, et.al., is entirely based on the Vector Space Model. The system consists of three main functions, namely, indexing, term-weighting and retrieval.

Scrubbed documents are those resulting from removal of all punctuation (and in this case, word-net tags), leaving only terminal and non-terminal node set tags within the text. (See 3.2 for details.) From this scrubbed text, the content-bearing words which remain are indexed to produce a set of vectors (e.g., article vectors, terminal node vectors, etc.). The vectors are weighted, using *Lnu-ltu* term weighting [1]. Retrieval takes place when the query vector is correlated with an element vector using a similarity measure (in this case, inner product). We use Smart for, indexing, term weighting, and retrieval of documents or elements from the document collection.

2.2 XML Representation of Documents

The document collection for this research is a 2,666,190 article dump from Wikipedia [2]. The files are in XML format. ("The Extensible Markup Language (XML) is a simple text-based format for representing structured information: documents, data, configuration, books, transactions, invoices, and much more." [11, p.1]) An XML file is usually contained within tags (e.g., $\langle article \rangle$). Every XML tag must have a corresponding closing tag (e.g., $\langle /article \rangle$). The general structure of the Wiki XML document starts of with an article tag, followed by tags containing word-net information, the hdr, and then the body tag, (which here is the sibling of hdr). All the text which comprises the document is contained within $\langle bdy \rangle$ and $\langle /bdy \rangle$ tags.

Until 2009, the Wikipedia collection had a structure with *body* followed by *section* as an intermediate tag, followed by *paragraph* and other terminal node tags as seen in the Figure 2.1. Over the years, Wikipedia documents have changed, and a major change in the document collection was released by INEX in 2009. A new tag was introduced, changing the standard structure of the document. In particular, the introduction of the subsection or $\langle ss \rangle$ tag produced a major change in the structure of the XML documents. Initially, subsections were thought to exist one level below sections, but a detailed parse (as shown in Figure 2.2, below) shows that subsections were added at many different levels. In fact, subsections were added as children of other subsections, increasing the complexity of the parse.

2.3 Flex

Flex is our retrieval system [5]. It generates an element from an XML document at run time. Given a doc-tree representing the structure of the parse tree and its set of terminal nodes, Flex generates the XML elements of the tree, dynamically, from the terminal nodes to the top (i.e, article) node. It correlates the query with each element vector to produce a rank-ordered list of elements from that document.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<article>
   <name id="2778410">Calvinistic Methodists</name>
   <conversionwarning>0</conversionwarning>
 ▼<body>
    <emph3>Calvinistic Methodists</emph3>
    are a body of
    <collectionlink xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" xlink:href="1339886.xml">Christian</collectionlink>
    s formingthe
    <collectionlink xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" xlink:href="1131689.xml">PresbyterianChurch of Wales</collectionlink>
    and claiming to be the onlydenomination of the
    <collectionlink xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" xlink:href="24403.xml">Presbyterian</collectionlink>
    orderin
    <collectionlink xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" xlink:href="69894.xml">Wales</collectionlink>
    which is of purelyWelsh origin.
   ▶<p>...</p>
   ▶<p>...</p>
   ▶<p>...</p>
   ▶<p>...</p>
   ▶<p>...</p>
   ▶<p>...</p>
   ▶<p>...</p>
   ▶<p>...</p>
   ▶<p>...</p>
   ▶<p>...</p>
   ▶<p>...</p>
   ▶<p>...</p>
   ▶<p>...</p>
   ▶<p>...</p>
   ▼<section>
      <title>See also</title>
     ▶<normallist>...</normallist>
    </section>
   ▼<section>
      <title>References</title>
     ▶<normallist>...</normallist>
    </section>
  </body>
</article>
```

Figure 2.1: An XML Document from the 2008 Wikipedia Collection

Given that Flex allows us to generate each element of the document tree from the bottom up, we can rank each element in terms of its correlation with the query. The weighting mechanism is *Lnu-ltu*. We then identify the focused elements from the list of ranked elements and use them as candidates for snippet generation.

2.4 Related Work

In 2009 our IR research group participated in the INEX Adhoc track, consisting of the Thorough and Focused tasks. The Thorough task is designed to retrieve all elements of interest to the query, and elements may overlap within the same document. The Focused task retrieves non-overlapping (or focused) elements that correlate with the query. Overlap occurs when both a child and a parent element are retrieved. Child, Section and Correlation are three strategies used for producing focused elements from the Thorough elements. In the Child strategy, the child is chosen over any of its parents. In the Section strategy, the highest correlated non-body element is

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<article>
  ▼<header>
     <title>Superman music</title>
   ▼<categories>
      Superman films Superman television series Superman music Film soundtracks Film scores
    </categories>
  </header>
 ▼<bdy>
   ▶<sec>...</sec>
   ▶<sec>...</sec>
   ▶<sec>...</sec>
   ▶<sec>...</sec>
   ▶<sec>...</sec>
   ▶<sec>...</sec>
   ▶<sec>...</sec>
   ▼<sec>
      <st>The soundtracks</st>
    ▼<ss1>
       <st>Superman</st>
     ▼<ss2>
        <st>Original release</st>
      ▶<p>...</p>
      ▼<ss3>
         <st>WB Records Track Listing</st>
        ▼<ss4>
           <st>Side 1</st>
          ▼<p>
             Theme from Superman Main Title 4 24 The Planet Krypton 4 45 Destruction of Krypton 5 58 The Trip to Earth 2 23 Growing Up 2 34
           </p>
          </ss4>
        ▼<ss4>
           <st>Side 2</st>
          ▶<p>...</p>
          </ss4>
        ▶<ss4>...</ss4>
        ▶<ss4>...</ss4>
        </ss3>
      </ss2>
     ▶<ss2>...</ss2>
     ▶<ss2>...</ss2>
     ▶<ss2>...</ss2>
     </ss1>
    ▶<ss1>...</ss1>
    ▶<ss1>...</ss1>
    ▶<ss1>...</ss1>
    ▶<ss1>...</ss1>
    ▶<ss1>...</ss1>
    ▼<ss1>
       <st>Superman Returns</st>
     ▶<p>...</p>
     </ss1>
   </sec>
  ▶<sec>...</sec>
  ▶<mt>...</mt>
  </bdy>
</article>
```

Figure 2.2: An XML document from the 2009 Wikipedia collection

chosen irrespective of its location in the tree. In the Correlation strategy, the highest correlated element among all the elements along a path is chosen irrespective of its position in the tree.

In 2010 the Adhoc track shifted its focus to the length and correlating content within the element rather than the element itself, with two new tasks called Relevance in Context (RiC) and Restricted Relevance in Context (RRiC). Both of these tasks utilized focused elements as the base. Experiments were performed and reported in [8]. Results, evaluated in terms of MAgP (mean average generalized precision), are shown in Table 2.1 and Table 2.2. These results show that the MAgP values of our

Table 2.1: 2010 Ric Task Results with T2I Evaluation

Participant	MAgP	Rank
ENSM-SE	0.1977	1
University of Minnesota Duluth(Section Strategy)*	0.1877	-
University of Minnesota Duluth(Child Strategy)*	0.1833	-
University of Minnesota Duluth(Child Strategy)*	0.1733	-
Peking University	0.1615	2
LIA - University of Avignon	0.1588	3
Queensland University of Technology	0.1521	4
University of Otago	0.1436	5
Radboud University Nijmegen	0.1377	6
Renmin University of China	0.1372	7
RMIT University	0.1335	8
Doshisha University	0.1014	9
University of Amsterdam	0.0695	10

Table 2.2: 2010 RRic Task Results with T2I Evaluation

Participant	MAgP	Rank
University of Minnesota Duluth(Child Strategy)*	0.1782	-
University of Minnesota Duluth(Section Strategy)*	0.1762	-
University of Minnesota Duluth(Correlation Strategy)*	0.1632	-
Peking University	0.1580	1
LIA - University of Avignon	0.1541	2
Queensland University of Technology	0.1508	3
University of Otago	0.1436	4
Radboud University Nijmegen	0.1375	5
University of Waterloo	0.0650	6
Doshisha University	0.0600	7
University of Amsterdam	0.0576	8
Indian Statistical Institute	0.0485	9

runs rank above those of the top-ten ranked runs submitted. Clearly, our focused elements are competitive with respect to results submitted by other participants. This realization caused us to consider the value of focused elements with respect to other tasks, and in particular, to the Snippet Retrieval task proposed in 2011.

In 2011, the Adhoc track shifted its focus to Snippet Retrieval, where the snippet is defined as a text segment extracted from a document that enables a user to assess

the document's relevance to a query without seeing the document itself. The length of the snippet (set at 300 characters in 2011) was cut to 160 characters in 2012. The snippets generated by the participants were analyzed by crowd sourcing, wherein all participants were sent snippets generated by other participants and asked to validate correctness by classifying the snippet as relevant or not. Our participation in the 2011 Snippet Track is based largely on our 2010 Focused task results. We reasoned that a snippet generated from the highest-ranked focused element in a document would serve as a good representation of the larger whole in deciding the relevance of the document, i.e., it would serve as (at least the basis of) a good snippet. The details of the algorithm used for snippet generation and experimental results are discussed in Chapter 4.

3 Implementation

This chapter describes the scrubbing of the Wikipedia 2009 collection [2], the creation of the *para+mt* and the level parses and the indexing. We then examine how element retrieval may be used to create a specific subset of documents.

3.1 Scrubbing

Scrubbing is the first step involved in transforming the XML document into a document we can use. The process involves scanning the document, removing non-alphanumeric characters, and preserving the remaining content in its original form. Word-net information in general provides semantic information related to the document. Our retrieval process is based on the document itself in terms of its content. We remove tags which are present between *article* and the *body* tags because they increase the number of levels to be parsed. The tags are added later to annotate the document.

3.1.1 Removing Tags with Wordnet Attributes

Each document is scanned in a top-down manner using the Libxml2 parser. In order to determine whether a tag is meant to be a wordnet or not, attributes of each tag are examined and if there is a *Wordnet* attribute contained in it, then it is deemed to be a wordnet tag(as sown in Figure 3.1). (In reconstructing the new xml file, the wordnet tag is not output.) Before we start scrubbing XML documents, we need to identify a terminal node tag set, which represents each document at its lowest level. As a result, during the scrubbing process any tag at the lowest level

```
−<p>
  In the English version of
 −<it>
  −<message wordnetid="106598915" confidence="0.8">
   −<series wordnetid="108457976" confidence="0.8">
    −<arrangement wordnetid="107938773" confidence="0.8">
     −<narrative wordnetid="107221094" confidence="0.8">
      −<firm wordnetid="108059870" confidence="0.8">
       −<ordering wordnetid="108456993" confidence="0.8">
        −<enterprise wordnetid="108056231" confidence="0.8">
         −<group wordnetid="100031264" confidence="0.8">
          −<business wordnetid="108061042" confidence="0.8">
           −<publisher wordnetid="108062623" confidence="0.8">
              <link xlink:type="simple" xlink:href="../986/1952986.xml"> Geronimo Stilton</link>
            </publisher>
          </business>
         </group>
        </enterprise>
       </ordering>
      </firm>
     </narrative>
    </arrangement>
   </series>
  </message>
 </it>
 series, the main protagonist has a friend named "Hercule Poirat".
</p>
```


Figure 3.1: Document with Wordnet Tags

of an XML document which is not a part of the terminal node set is removed along with its content. (Referring to the cases like references and external links not about untagged text within the parent, removing the element as a whole.)

3.1.2 Creation of mt Tags

In the process of creating a scrubbed XML document, we create an *mt* tag as the child of the *bdy*. The purpose of creating an *mt* tag is to guarantee that we capture all the text of the document. There are cases wherein the content of a non-terminal node is left untagged (as show in Figure 3.2) due to removal of the word-net tags or because the content is untagged in the first place. All instances of untagged text which belong to the same parent are tagged within *mt* tags as shown in Figure 3.3.

```
−<bdy>
  This article refers to the contracts in Australia. For a list of other country's broadcasting rights, see
 −<artifact wordnetid="100021939" confidence="0.8">
  −<instrumentality wordnetid="103575240" confidence="0.8">
   −<medium wordnetid="106254669" confidence="0.8">
     <link xlink:type="simple" xlink:href="../840/2031840.xml"> Sports television broadcast contracts</link>
    </medium>
   </instrumentality>
  </artifact>
```

Figure 3.2: Portios of an XML Document Containing Untagged Text

```
MT tag
------

/article[1]/bdy[1]/mt[1]/
This article refers to the contracts in Australia For a list of other country s broadcasting rights see Sports television broadcast contracts
```

Figure 3.3: An mt Tag and the Text Associated with it

3.1.3 Removal of Sections with References

During the process of evaluating the snippets that were generated for the 2011 INEX tasks, we found that many top-ranked elements were sections which contained references related information in them. The reference sections were selected due to repeated occurrences of the query terms within them. The content present in the reference section contained little useful information (and most of the content is referred to other documents). So in generating the scrubbed XML document, we removed the sections with *st* (section title) tags containing *references*, *see also* and *external links*.

The resulting XML document formed from these three steps (delineated above) is called the scrubbed XML document (as shown in Figure 3.4). Scrubbing the entire collection takes approximately eight hours.

3.2 Parsing

The XML documents are parsed using a DOM (Document Object Model) hierarchy, where the document is represented as a tree with each tag representing a node

```
−<article>
 −<header>
    <title> Gates of Cairo </title>
   +<categories></categories>
   </header>
 −<bdy>
   +<table></table>
    <p> Cairo s gates include </p>
   +<p></p>
   −<p>
      Bb is Arabic for door or entrance from bawwaba to divide into chapters or sections
    </p>
   −<mt>
      The Egypt ian city of Cairo has in its history had a significant number of fortified gate s protecting both the inner and outer city Apart from the purpose of
      defense they were also used for differentiation of the varied social and economic classes Often the gates were heavily decorated as artistic pieces
    </mt>
   </bdy>
 </article>
```

Figure 3.4: A scrubbed XML Document Containing an mt Node

of the tree. To parse the new 2009 Wikipedia collection we used the Libxml2 parser instead of the parser used previously which was written in Pearl for the 2008 Collection. The previous parser was unable to handle nested subsections and so some Xpaths that were generated were invalid. The text of a particular node can be found by parsing the XML document with the specified tag. Every tag in the XML document is identified by its Xpath which specifies the location of the node in the XML tree. Using the Xpath and libxml2 parser we get the content of a particular node in the document.

To retrieve an element, we identify a tag set which we call the terminal node set (e.g., *title*, *categories*, *template*, *p*, *image*, *list*, *normallist*, *numberlist*, *definitionlist*, *table*, *ou* and *lu* are identified as terminal nodes.) Some of these tags were subsequently found to identify elements too short to be a meaningful and were eliminated. But some tags, like *title*, even though identified as small, are important in identifying relevant elements. For example, *title* often gives useful information about the document. All tags in the XML document not found in the terminal node set are called non-terminal nodes, and they may contain other non-terminal or terminal nodes within them. The tags *article*, *header*, *bdy*, *sec* and $ss\langle 1-9\rangle$ are all part of the non-terminal node set.

For computing the correlation of an element (irrespective of its position in the

document tree), we need the text associated with each terminal node. Then Flex builds the intermediate elements with the help of the doc-tree and the terminal node index.

The *Lnu-ltu* weighting scheme requires building an all-element representation of the document collection and using this collection to determine the proper values of the slope and pivot. To build the all-element collection, we need an all-element parse, consisting of all elements from the individual element (level) parses.

Using Libxml2, we generated the level parses and the para+mt parses. The para+mt parse is the parse consisting of all terminal nodes and mt(which are also terminal nodes). The level parses are the parses for non-terminal nodes occurring at each particular level in the XML document tree.

3.2.1 Level Parses

Parsing a document for all the nodes at a depth of n from the root node (considering root as level-0) is called as *level-n* parse. In creating the level parses, the maximum depth of the tree is first computed to determine the number of level parses needed for a XML document. The maximum level is the level (maximum distance from article to that section) at which a nonterminal node is present in the document. The maximum level for the document as shown in Figure 3.5 is 4. While parsing a document at *level-n*, each tag in the XML document is checked from the root and if the tag occurs at *level-n*, the content of the tag is added (and there is no parsing further into its child tags). The article and all its level parses are shown in Figure 3.6 and Figure 3.7.

3.2.2 Para+mt Parses

Parsing a terminal node set (the *para+mt* tag set) is different, since the occurrence of terminal nodes is not fixed at a particular level, and we cannot parse them based on levels. As the set of terminal nodes is defined for the entire collection, producing

```
+<ss1></ss1>
+<ss1></ss1>
+<ss1></ss1>
+<ss1></ss1>
+<ss1></ss1>
+<ss1></ss1>
+<ss1></ss1>
+<ss1></ss1>
−<ss1>
    <st> Football (Soccer)</st>
  +<ss4></ss4>
  +<ss4></ss4>
  +<ss4></ss4>
  +<ss4></ss4>
  +<ss4></ss4>
  −<ss3>
      <st> International Tournaments</st>
    +<p></p>
    −<ss4>
        <st> Other</st>
      +<p></p>
      </ss4>
    </ss3>
  </ss1>
+<ss1></ss1>
```

Figure 3.5: 4 Levels in an XML Document

the *para+mt* parse is done by checking each XML node aganist the terminal node set in a pre-order manner and if the node is found to be in the terminal node set then the content of the node is extracted. The sample output of the *para+mt* parse is shown in Figure 3.8.

3.2.3 Creation of Doctrees

We generate a pre-order traversal of the tree, called a *doc-tree*, by storing the Xpath of the root, its number of children, and a flag to indicate the presence or

/article[1]/
Sports broadcasting contracts in Australia Television in Australia American Football Gridiron National Football League ox Sports One live game every Monday night Ten HD Two live games per week plus play offs ESPN Australia Super Bowl Network Ten Ten HD Athletics IAAF World Championships in Athletics SBS Australian Rules Football Seven Network Friday night 1 match amp Sunday afternoon 1 match Finals Series 1 Qualifying Final Match 1 Elimination Final Match 1 Semi Final Match and 1 Premliminary Final match Grand Final 2008 2010 Network Ten Saturday afternoon 1 match amp Saturday night 1 match Finals Series 1 Qualifying Final Match 1 Elimination Final Match 1 Semi Final Match and 1 Premliminary Final match Grand Final 2007 2009 2011 Fox Sports Friday night 1 match NSW ACT Qld only Saturday afternoon 1 match Saturday night 1 match occasionally 2 in NSW ACT Qld Sunday lunch 1 match amp Sunday twilight 1 match Replays of all games scheduled throughout the week This includes replays of Finals Series matches and the Grand Final Australian Football League Pre season NAB Cup Seven Network Final 2007 2009 2011 Network Ten Final 2008 2010 Earlier round matches are divided between Seven Network Network Ten and Fox Sports with Fox Sports usually having live coverage of games that would be shown on the other two networks delayed Brownlow Medal Seven Network 2007 2009 2011 Network Ten 2008 2010 Victorian Football League ABC1 1 live game each Saturday afternoon All Finals and Grand Final live In other states it is covered by ABC1 s sister channel ABC2 Games are often shown late at night during the week in other states South Australian National Football League ABC1 In other states it is covered by ABC1 s sister channel ABC2 Games are often shown late at night during the week in other states West Australian Football League ABC1 In other states it is covered by ABC1 s sister channel ABC2 Games are often shown late at night during the week in other states Matches are shown on channel 518 promoted as Fox Sports Plus Matches are shown occasionally on channel 518 depends on who is playing Baseball Major League Baseball Fox Sports ESPN Australia Basketball National Basketball League Nine Network replays one game every Sunday Morning Fox Sports has rights to every game Women s National Basketball League ABC1 has 1 game every Saturday afternoon ABC2 has 1 live game every Friday Night That same game is often replayed throughout the week National Basketball Association ESPN Australia Fox Sports Commonwealth Games 2010 Commonwealth Games Network Ten Fox Sports Cricket Test Cricket in Australia Nine Network One day International Cricket in Australia Nine Network Cricket World Cup Nine Network to show Eight Super 8 matches One Semi final and the Final Fox Sports to cover all matches ICC World Twenty20 Nine Network to cover all games featuring Australia Semi finals and the Final Fox Sports to cover all matches Overseas Test cricket Fox Sports covers most games all that involve Australia Setanta Sports covers tours of Bangladesh not involving Australia and some tours of India and Pakistan Overseas One day International Cricket Fox Sports covers most games all that involve Australia Setanta Sports covers tours of Bangladesh not involving Australia and some tours of India and Pakistan Allan Border Medal Fox Sports live coverage Nine Network delayed coverage Sheffield Shield Fox Sports final only Ford Ranger One Day Cup Fox Sports KFC Twenty20 Big Bash Fox Sports Indian Premier League Network Ten Matches shown live all nights except saturday where games are delayed by 1 2 hours due to Saturday Night AFL These games shown live exclusively in high definition on Ten HD Ten HD also airs occasional replays of matches and a Highlights show also airs every sunday afternoon ICC Champions Trophy Fox Sports 2009 Ashes series SBS XXXX Gold Beach Cricket Network Ten Cycling Tour de France SBS live coverage Fox Sports highlights Giro d Italia Eurosport Asia Pacific Tour of Spain SBS Football Soccer Leagues A league Fox Sports English Premier League Fox Sports English Football Championship Fox Sports English Conference Setanta Sports Spanish La Liga ESPN Australia French Ligue 1 Setanta Sports German Bundesliga Setanta Sports Italian Serie A ESPN Australia Setanta Sports RAI International Scottish Premier League Setanta Sports Portuguese Liga Setanta Sports Eredivisie Setanta Sports Major League Soccer ESPN Australia Turkcell Sper Lig Lig TV Domestic Cups English FA Cup Setanta Sports English League Cup Fox Sports German DFB Cup Setanta Sports Scottish Cup Setanta Sports Scottish League Cup Setanta Sports Coppa Italia ESPN Australia SBS French Cup Setanta Sports French League Cup Setanta Sports Dutch Cup Setanta Sports International Club Competitions AFC Champions League Fox Sports UEFA Champions League SBS ESPN Australia UEFA Cup Setanta Sports ESPN Australia SBS EuroSport Copa Libertadores Setanta Sports Copa Sudamericana Setanta Sports FIFA Club World Cup SBS Club Channel Coverage Chelsea TV MUTV Celtic TV Rangers TV Barca TV Setanta Sports International Matches Australian National Mens Team Socceroos matches Fox Sports Australian Youth Womens National Teams SBS English Home Internationals Setanta Sports European World Cup Qualifying Setanta Sports ESPN Australia South American World Cup Qualifying Setanta Sports Other International Matches Setanta Sports ESPN Australia SBS International Tournaments UEFA Euro 2008 Setanta Sports SBS AFC Asian Cup Fox Sports FIFA World Cup SBS Other FIFA Tournaments SBS 2008 African Cup of Nations Eurosport Other Victorian Premier League Channel 31 Melbourne Hyundai Club Challenge Sydney FC vs Los Angeles Galaxy Aired by Network Ten in November 2007 SBS shows a football fixture as part of their Sunday afternoon football show The World Game This is often from the UEFA Champions League but is sometimes a game from one of a number of South American European or Asian leagues Aurora has a one hour highlights program showing highlights from Australia s various amateur leagues each week called the Australian Premier League Highlights Show Golf Australian Open Seven Network British Open ESPN Australia Ryder Cup Fox Sports PGA European Tour ESPN Australia PGA Tour ESPN Australia U S Open Fox Sports ESPN Australia U S Masters Network Ten Fox Sports HSBC New Zealand PGA Championship Ten HD WGC Accenture MatchPlay Championship Ten HD WGC CA Championship Ten HD WCG Bridgestone Invitational Ten HD WGC Mission Hills World Cup Ten HD Women s Australian Open ABC1 Horse racing Melbourne Cup Seven Network Melbourne Spring Racing Carnival Nine Network Melbourne Sydney Hong Kong and other major thoroughbred race meetings TVN All metropolitan country and international race meetings Sky Racing Kentucky Derby ESPN Australia Ironman Surf lifesaving Australian Club Championship Nine Network Lacrosse National Lacrosse League ESPN Australia Motor Racing FIA Formula One Network Ten V8 Supercars Seven Network 200712 A1 Grand Prix Fox Sports live Nine Network highlights FIA World Rally Championships SBS Motorcycle Grand Prix MotoGP Fox Sports live Network Ten delayed highlights live for Australian MotoGP only NASCAR Fox

Figure 3.6: Article Parse of the XML Document

Level-3 parse

/article[1]/bdy[1]/ss1[9]/ss4[1]/
Leagues A league Fox Sports English Premier League Fox Sports English Football Championship Fox Sports English Conference Setanta Sports Spanish La Liga ESPN Australia French Ligue 1 Setanta Sports German Bundesliga Setanta Sports Italian Serie A ESPN Australia Setanta Sports RAI International Scottish Premier League Setanta Sports Portuguese Liga Setanta Sports Eredivisie Setanta Sports Major League Soccer ESPN Australia Turkcell Sper Lig Lig TV
/article[1]/bdy[1]/ss1[9]/ss4[2]/
Domestic Cups English FA Cup Setanta Sports English League Cup Fox Sports German DFB Cup Setanta Sports Scottish Cup Setanta Sports Scottish League Cup Setanta Sports Coppa Italia ESPN Australia SBS French Cup Setanta Sports French League Cup Setanta Sports Dutch Cup Setanta Sports
/article[1]/bdy[1]/ss1[9]/ss4[3]/
International Club Competitions AFC Champions League Fox Sports UEFA Champions League SBS ESPN Australia UEFA Cup Setanta Sports ESPN Australia SBS EuroSport Copa Libertadores Setanta Sports Copa Sudamericana Setanta Sports FIFA Club World Cup SBS
/article[1]/bdy[1]/ss1[9]/ss4[4]/
Club Channel Coverage Chelsea TV MUTV Celtic TV Rangers TV Barca TV Setanta Sports
/article[1]/bdy[1]/ss1[9]/ss4[5]/
International Matches Australian National Mens Team Socceroos matches Fox Sports Australian Youth Womens National Teams SBS English Home Internationals Setanta Sports European World Cup Qualifying Setanta Sports ESPN Australia South American World Cup Qualifying Setanta Sports Other International Matches Setanta Sports ESPN Australia SBS
/article[1]/bdy[1]/ss1[9]/ss3[1]/
International Tournaments UEFA Euro 2008 Setanta Sports SBS AFC Asian Cup Fox Sports FIFA World Cup SBS Other FIFA Tournaments SBS 2008 African Cup of Nations Eurosport Other Victorian Premier League Channel 31 Melbourne Hyundai Club Challenge Sydney FC vs Los Angeles Galaxy Aired by Network Ten in November 2007 SBS shows a football fixture as part of their Sunday afternoon football show The World Game This is often from the UEFA Champions League but is sometimes a game from one of a number of South American European or Asian leagues Aurora has a one hour highlights program showing highlights from Australia s various amateur leagues each week called the Australian Premier League Highlights Show

Level-4 Parse

/article[1]/bdy[1]/ss1[9]/ss3[1]/ss4[1]/
Other Victorian Premier League Channel 31 Melbourne Hyundai Club Challenge Sydney FC vs Los Angeles Galaxy Aired by Network Ten in November 2007 SBS shows a football fixture as part of their Sunday afternoon football show The World Game This is often from the UEFA Champions League but is sometimes a game from one of a number of South American European or Asian leagues Aurora has a one hour highlights program showing highlights from Australia s various amateur leagues each week called the Australian Premier League Highlights Show

Figure 3.7: Various Level Parses for the XML Document

absence of a sibling to the right. The entire XML document can be reconstructed using the doc-tree by populating the parent node(s) from the leaf nodes. Doc-trees provide context for the nodes, as we generate each parent node from its children and

```
/article[1]/bdy[1]/ss1[9]/st[1]/
Football Soccer
/article[1]/bdy[1]/ss1[9]/ss4[1]/st[1]/
Leagues
/article[1]/bdy[1]/ss1[9]/ss4[1]/p[1]/
A league Fox Sports English Premier League Fox Sports English Football Championship Fox Sports    English Conference    Setanta Sports    Spanish La Liga    ESPN
Australia    French Ligue 1    Setanta Sports    German Bundesliga    Setanta Sports    Italian Serie A    ESPN Australia    Setanta Sports    RAI International
Scottish Premier League    Setanta Sports    Portuguese Liga    Setanta Sports    Eredivisie    Setanta Sports    Major League Soccer    ESPN Australia    Turkcell Sper
Lig    Lig TV
/article[1]/bdy[1]/ss1[9]/ss4[2]/st[1]/
Domestic Cups
/article[1]/bdy[1]/ss1[9]/ss4[2]/p[1]/
English FA Cup    Setanta Sports      English League Cup    Fox Sports    German DFB Cup    Setanta Sports      Scottish Cup    Setanta Sports    Scottish League Cup
Setanta Sports    Coppa Italia    ESPN Australia    SBS    French Cup    Setanta Sports    French League Cup    Setanta Sports    Dutch Cup    Setanta Sports
/article[1]/bdy[1]/ss1[9]/ss4[3]/st[1]/
International Club Competitions
/article[1]/bdy[1]/ss1[9]/ss4[3]/p[1]/
AFC Champions League    Fox Sports    UEFA Champions League    SBS    ESPN Australia    UEFA Cup    Setanta Sports    ESPN Australia    SBS    EuroSport    Copa
Libertadores    Setanta Sports    Copa Sudamericana    Setanta Sports    FIFA Club World Cup    SBS
/article[1]/bdy[1]/ss1[9]/ss4[4]/st[1]/
Club Channel Coverage
/article[1]/bdy[1]/ss1[9]/ss4[4]/p[1]/
Chelsea TV    MUTV    Celtic TV    Rangers TV    Barca TV    Setanta Sports
/article[1]/bdy[1]/ss1[9]/ss4[5]/st[1]/
International Matches
/article[1]/bdy[1]/ss1[9]/ss4[5]/p[1]/
 Australian National Mens Team Socceroos matches    Fox Sports    Australian Youth Womens National Teams    SBS    English Home Internationals    Setanta Sports
European World Cup Qualifying    Setanta Sports    ESPN Australia    South American World Cup Qualifying    Setanta Sports    Other International Matches    Setanta
Sports    ESPN Australia    SBS
/article[1]/bdy[1]/ss1[9]/ss3[1]/st[1]/
International Tournaments
/article[1]/bdy[1]/ss1[9]/ss3[1]/p[1]/
UEFA Euro 2008    Setanta Sports    SBS      AFC Asian Cup    Fox Sports    FIFA World Cup    SBS    Other FIFA Tournaments    SBS    2008 African Cup of Nations
Eurosport
/article[1]/bdy[1]/ss1[9]/ss3[1]/ss4[1]/st[1]/
Other
/article[1]/bdy[1]/ss1[9]/ss3[1]/ss4[1]/p[1]/
Victorian Premier League    Channel 31 Melbourne    Hyundai Club Challenge  Sydney FC  vs  Los Angeles Galaxy  Aired by  Network Ten  in November 2007    SBS  shows a
football fixture as part of their Sunday afternoon football show  The World Game  This is often from the  UEFA Champions League  but is sometimes a game from one of a
number of South American European or Asian leagues    Aurora  has a one hour highlights program showing highlights from Australia s various amateur leagues each week
called the  Australian Premier League Highlights Show
```

Figure 3.8: The para+mt Parse for the XML Document

continue until the *article* node is generated.

A pre-order traversal is performed on the cleaned XML document and all nodes are identified in terms of their Xpaths and saved as the doc-tree, as shown in Figure 3.10. While the doc-tree is required to reconstruct the document at run time, it does not reflect the node in the structure of the original XML documents, (which is required by INEX for evaluation purpose). Thus a corresponding set of doc-trees is also created for the original XML documents (as shown in Figure 3.9). There is a relation between the Xpath formed from the cleaned XML document and the Xpath that is generated from the corresponding doc in the Wikipedia collection.

3.2.4 Statistics of Parses

After parsing the XML documents the number of tags occurring at various levels were analyzed. The number of tags occurring deeper down the document tree (higher levels) from the root has decreased; there were fewer documents with higher levels. We found patterns such as (1) a sub-section named ss4 can be at levels 3 to 6 but

```
/article[1]/     1      0
/article[1]/series[1]/fictional_character[1]/research_worker[1]/header[1]/       4      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/header[1]/title[1]/      0      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/header[1]/categories[1]/         4      0
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/          12     0
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/template[1]/      2      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/p[1]/     7      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/p[2]/     6      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/           5      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/st[1]/     0      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[1]/    4      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[1]/st[1]/      0      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[1]/p[1]/       11     1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[1]/p[2]/       3      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[1]/p[3]/       4      0
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[2]/    3      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[2]/st[1]/      0      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[2]/p[1]/       4      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[2]/p[2]/       1      0
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[3]/    4      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[3]/st[1]/      0      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[3]/p[1]/       4      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[3]/p[2]/       5      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[3]/p[3]/       7      0
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[4]/    5      0
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[4]/st[1]/      0      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[4]/p[1]/       4      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[4]/p[2]/       1      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[4]/p[3]/       6      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[1]/ss1[4]/p[4]/       1      0
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[2]/           6      1
/article[1]/series[1]/fictional_character[1]/research_worker[1]/bdy[1]/sec[2]/st[1]/     0      1
```

Figure 3.9: Partial Doctree Generated from the Original XML Document

```
/article[1]/bdy[1]/sec[8]/st[1]/         0      1
/article[1]/bdy[1]/sec[8]/ss1[1]/        5      1
/article[1]/bdy[1]/sec[8]/ss1[1]/st[1]/          0      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/         3      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/st[1]/           0      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/p[1]/    0      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/          5      0
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/st[1]/    0      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[1]/           2      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[1]/st[1]/     0      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[1]/p[1]/      0      0
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[2]/           2      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[2]/st[1]/     0      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[2]/p[1]/      0      0
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[3]/           2      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[3]/st[1]/     0      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[3]/p[1]/      0      0
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[4]/           3      0
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[4]/st[1]/     0      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[4]/p[1]/      0      1
/article[1]/bdy[1]/sec[8]/ss1[1]/ss2[1]/ss3[1]/ss4[4]/p[2]/      0      0
```

Figure 3.10: Part of the Doc-tree Generated from Cleaned XML Document

not at level 7, (2)*bdy* and header are always present at level 1, (3) and *sec* is always present in level 2 and (4) sometimes the subsection (*ss* tags) preceded the "body" (*bdy* tag). The Figure 3.11 shows the various tags occurring at different levels. The statistics related to para+mt parse are not confined to a single level but are spread across all the levels due to their occurrence at different levels. The count of terminal nodes are shown in Table 3.1.

```
Level-0
-----------
article 2666183

Level-1
-----------
bdy 2653501
header 2666183

Level-2
-----------
ss4 4
ss2 11318
sec 4075889
ss1 114901
ss3 4135

Level-3
-----------
ss4 32
ss2 53585
ss1 1586824
ss3 4189

Level-4
-----------
ss4 169
ss2 200024
ss3 6836

Level-5
-----------
ss4 315
ss3 14958

Level-6
-----------
ss4 467

Level-7
-----------
ss5 2
```

Figure 3.11: Statistics of Various Tags in Level Parses

Table 3.1: Statistics of Various Tags in para+mt Parse

Tag	count
mt	1764368
p	16754731
template	912012
table	392279
st	6072603
categories	2547498
title	2666183

4 Results

4.1 Overview of Snippet Task

A snippet as defined by INEX is "text which allows the user to determine the relevance of each document, without needing to view the document itself" [4 , p.1]. A snippet as defined here is 300 characters in length and can serve as a summary or an extract of the document. The basis for snippet generation in these experiments is the focused elements retrieved by the query. Focused elements by definition are non-overlapping. Thus the focused element with the highest correlation to the query, for example, can be viewed as a precise description of text that correlates highly with the query. So in generating the snippet, output from the focused task is taken as input for further processing. The snippet refinement process is detailed below.

Step 1. The query is divided into tokens.

Step 2. For each sentence in the text (i.e., focused) element

a. If the length of the sentence is less than min, then go to Step 2.

b. Initialize the score of the sentence to 0.

c. For each token in the query

Increment the score of sentence by the frequency of the token in the sentence.

d. The score of the sentence = score divided by number of tokens in the sentence.

Step 3. Sort the sentences in descending order of their scores.

Step 4. Merge the sorted sentences in order to form a single text element.

Step 5. The first 300 characters of the paragraph form the snippet.

As mentioned in Chapter 2, there are mechanisms for choosing a focused element from a set of focused elements associated with a given document. We have used three

sub-strategies in previous experiments [8], namely, the child, section and correlation strategies. In the child strategy, the highest correlated focused child element along a path is chosen. In the section strategy, the highest correlated non-body element (which is usually a section) is selected. In the correlation strategy, the element with the highest correlation is selected, irrespective of its location in the tree.

Four different sets of snippets were generated and submitted to INEX for evaluation in 2011. We chose the focused element generated using the child and the correlation strategies. The correlation strategy is chosen as a basis for snippet generation because highly correlating focused elements seem to be good candidates for this purpose. A highly correlating focused child element is attractive because it contains necessary information without additional terms. The algorithm specified above is used for generating two sets of snippets, one based on focused child elements and other set based on focused correlation elements. INEX uses crowd sourcing to evaluate the snippets reported by the participants. In crowd sourcing, each participant serves as an assessor. He or she is provided with a set of participant-generated snippets and asked to mark each document as relevant or non-relevant based on the snippet read. (Relevance of the document has already been established by the organizers; information which is not available to the assessors when accessing the snippets.) All assessments are reported back to INEX, where the data is analyzed with respect to the pre-established relevance of the document to the query.

The metrics used by INEX to arrive at the results are listed below:

1) Mean prediction accuracy (MPA), the ratio of results the assessor correctly assessed, averaged over all topics: (TP+TN)/(TP+FN+TN+FP)

2) Mean normalized prediction accuracy (MNPA), the average of the relevant results correctly assessed, and the irrelevant results correctly assessed, averaged over all topics: 0.5*TP/(TP+FN) + 0.5*TN/(TN+FP)

3) Recall, the percentage of relevant documents correctly assessed, averaged over all topics: TP/(TP+FN)

4) Negative recall (NR) or specicity, the percentage of irrelevant documents correctly assessed, averaged over all topics: TN/(TN+FP)

5) Positive agreement (PA), the conditional probability of agreement between snippet assessor and document assessor (i.e. ground truth), given that one of the two is judged relevant. Equivalent to F1 score: 2*TP/(2*TP+FP+FN)

6) Negative agreement (NA), the conditional probability of agreement between snippet assessor and document assessor (i.e. ground truth), given that one of the two judged irrelevant: 2*TN/(2*TN+FP+FN)

Here TN stands for true negative, where a snippet assessed as irrelevant is actually coming from an irrelevant document; FN for false negative, where the snippet that is accessed irrelevant is coming from a relevant document; TP for true positive, where the snippet that is assessed relevant is coming from a relevant document; and TM for true negative, where the snippet that is accessed relevant is coming from a irrelevant document.

Results of the 2011 Snippet Retrieval Task are shown in Table 4.1. Only the top-ten results are of interest. The UMD team placed one run in the top ten out of 50 runs. This run, ranked at 9 in the top-ranked returns, produced a geometric mean (GM) value of 0.5264. This run [p65-UMD SNIPPET RETRIEVAL RUN 3] was generated by applying the snippet refinement algorithm to the focused elements produced from the correlation strategy. geometric mean (GM), the primary evaluation metric and the one which determines the ranking, is the geometric mean of recall and negative recall averaged over all topics: sqrt (TN/(TN+FP) * TP/(TP+FN)). The top-ranked runs, from the list of runs submitted by the participants in the INEX snippet retrieval task in 2011, are listed below.

The other runs submitted by UMD are listed in the table Table 4.2. Run p65-UMD SNIPPET RETRIEVAL RUN 4, generated by applying the snippet refinement strategy to the focused elements produced by the child strategy, was ranked 26th . p65-UMD SNIPPET RETRIEVAL RUN 1, generated by taking the first three

Table 4.1: 2011 Snippet Track Final Results

Rank	Run	Score
1	p72-LDKE-1111	0.5705
2	p23-baseline	0.5505
3	p72-LDKE-0101	0.5472
4	p20-QUTFirst300	0.5416
5	p73-PKU ICST REF 11a	0.5341
6	p72-LDKE-1110	0.5317
7	p23-expanded-40	0.5294
8	p72-LDKE-0111	0.5270
9	p65-UMD SNIPPET RETRIEVAL RUN 3	0.5264
10	p20-QUTFocused	0.5242

hundred characters from the focused element produced by the correlation strategy, was ranked 30th, and run p65-UMD SNIPPET RETRIEVAL RUN 2, generated by taking the first 300 characters of the focused element produced by the child strategy, was ranked 35th.

Table 4.2: 2011 Snippet Track Results for Different Runs

Run	GM
p65-UMD SNIPPET RETRIEVAL RUN 3	0.5264
p65-UMD SNIPPET RETRIEVAL RUN 4	0.4680
p65-UMD SNIPPET RETRIEVAL RUN 1	0.4470
p65-UMD SNIPPET RETRIEVAL RUN 2	0.4270

Observations: From our overall results, our sentence scoring technique needs detailed analysis and improvement. We know from our previous experiments with focused elements [1] that they are top-ranked. From INEX experiments last year, we also learned that removing unnecessary content in the form of references and external links helps to generate better focused elements. We will incorporate this information in the next set of experiments.

4.2 Snippet Task 2012

The 2011 experiments raised the question of using multiple focused elements rather than a single focused element in snippet generation. Here we used three extraction strategies as the basis for generation snippet generation. One method uses the best focused element, the second method uses the text from all focused elements (sorted in descending order by correlation with the query and merged), and the third method uses the article as a whole. The focused elements are again selected based on the child and the correlation strategies. Two main sentence scoring methods are used for ranking sentences within the focused element. In the first, the presence of the query terms is given importance, while the second is a natural language processing approach where the occurrence of query terms in specified order is the determining factor. Eleven experiments were performed for the 2012 INEX snippet competition.

This first sentence scoring method is as follows. Given a focused element, we (1) count the number of query terms that appear in that focused element, (2) count the number of query terms in each of the sentence of the element, and (3) square value in step-2, squaring the number of terms establishes the importance of the sentence with those terms. The squared value is divided by the number of query terms in the element to get the score of the sentence. The sentences are sorted in descending order by their scores and a snippet is generated by extracting the first 180 characters (as 2012 snippet is of 180 characters).

In the second sentence scoring strategy, weight is given to the combination of query terms rather than individual query terms, so a sentence with a phrase made up of query terms ranks higher than one having query terms scattered in the sentence. In short, importance is given to the sentence containing query terms as a phrase. This strategy is modeled after the BLEU [12] method which is used in evaluating text translation. In this strategy, a sentence score equals one fourth of the total number of query N-grams (as in natural language processing) in the sentence divided by the total number of distinct N-grams in sentence.

4.2.1 2012 Experiments

Snippet-1 (UMD focused child 1) is the first 180 characters of a paragraph generated from a ranked list of focused elements generated for each document using the child strategy, where arithmetic scoring is used to rank the sentence, the text is joined to form a paragraph and the first 180 characters from that paragraph is used to generate the snippet. Snippet-2 (UMD focused correlation 1) is generated in exactly the same way except the focused elements come from correlation strategy instead of child strategy. Snippet-3 (UMD focused child 2 bleu) follows snippet-1, but the sentence scoring is changed to the BLEU method. Snippet-4 (UMD focused correlation 2 bleu) follows snippet-2, but the sentence scoring uses the BLEU method instead of the arithmetic scoring.

Snippets 5 to 7 differ from the first 4 snippets in that the source of the snippet is the article itself rather than the focused elements. Snippet-5 extracts the first 180 characters from the article to form the snippet. Snippet-6 (UMD snippets article score 180) uses the arithmetic sentence ranking to rank the sentences within the article, and then sorts (according to their score in descending order) those sentences and extracts the first 180 characters. Snippet-7 (UMD snippets article bleu 180) duplicates snippet-6 but the sentence ranking is changed to the BLEU method.

Snippets 8 through 11 are similar to snippets 1 through 4, but only the highest ranked focused element is used in generating the snippet in each case. Snippet-8 (UMD snippets child score 300 Single) corresponds to snippet-1, etc. All these snippets are submitted to INEX for evaluation by crowd sourcing.

The top ten results are incorporated below in Table 4.3. The metrics used are same as those of the previous year. The snippet (named UMD focused child1) generated by ranking sentences using arithmetic scoring on multiple focused elements(generated by child strategy) is ranked first in the competition. The snippet (called UMD focused child1) generated by ranking sentences using arithmetic scoring on multiple focused elements (generated by correlation strategy) is ranked 2nd. The snippet (named

UMD snippets article bleu 180) generated by applying BLEU sentence scoring on article ranked 6th, and the snippet generated (UMD snippets article score 180) by applying arithmetic scoring on sentences in article ranked 9th. The ranking is based on the geometric mean.

The results of the runs submitted by UMD are listed in the Table 4.4

Observations: There is a significant difference between the snippets that were generated using a single child focused element and those of multiple child focused elements. In the case of snippet generated from multiple focused elements the highly ranked sentences (within each focused element) form the snippet, unlike a snippet which is generated from a single focused element. The difference in the results between the snippet (UMD snippets child score 3) generated from a single focused element and snippet (UMD focused child 1) generated from multiple focused elements signifies that highly ranked sentences need be a part of the snippet instead of the sentences from the highest ranked focused element, as highly ranked sentences can be spread across different focused elements.

It appears that snippets generated from multiple focused elements, when compared (randomly) to those with a single focused element, are essentially the same considering that the 2012 snippet consists of only 180 characters. But results (UMD snippets child

Table 4.3: 2012 Snippet Track Results

Rank	Run	Score
1	UMD focused child 1	0.6121
2	QUT 2012 Focused	0.6096
3	UMD snippets article score 180	0.5798
4	QUT 2012 Focused Split	0.5712
5	TheCNGL DCU SnippetTrack2012 SRReferenceRun02	0.5648
6	TheCNGL DCU SnippetTrack2012 SRReferenceRun03	0.5510
7	UMD focused child 2 bleu	0.5466
8	SR2012-Baseline	0.5431
9	TheCNGL DCU SnippetTrack 2012 SRun04	0.5390
10	UMD focused correlation 1	0.5319

Table 4.4: 2012 Snippet Track Results for Different Runs

Run	GM
UMD focused child 1	0.6121
UMD snippets article score 180	0.5798
UMD focused child 2 bleu	0.5466
UMD focused correlation 1	0.5319
UMD snippets article bleu 180	0.5305
UMD focused correlation 2 bleu	0.5007
UMD snippets child score 3 100 Single	0.4991
UMD snippets correlation score 3 100 Single	0.4927
UMD snippets child score 300 Single	0.4706
UMD snippets correlation score 300 Single	0.4268
UMD snippets article first 180	0.4063

score 3 vs UMD focused child 1) demonstrate that there can be multiple sentences which are a part of those 180 characters and these sentences come from different focused elements as explained previously.

The other striking difference between the 2011 and the 2012 snippet results is that snippets generated in 2012 using focused elements from the correlation strategy are ranked below the ones generated from focused elements from child strategy. In 2011 the correlation strategy performed better than child strategy which is not the case in 2012. We need to look in detail as to why such a difference exists. The possible answers lie in the impact of these queries on the snippet generation and the way the documents might have been indexed (removal of some of the unnecessary tags). Some of the queries in 2011 (like "best movie") retrieved very few documents (as stated by INEX informally), thereby requiring a larger element (in general an element coming from correlation strategy) to produce a effective score between the query and the element. If the query retrieves more documents, a smaller element (as in child strategy) may be enough to produce a effective score between the query and the element. A good query in this context is one which retrieves a useful number of documents.

When comparing the sentence ranking strategies, it appears in most cases a snip-

pet extracted using the arithmetic scoring technique ranks higher than a snippet extracted using the BLEU scoring technique. A possible reason for this is the number of terms in the query. A query containing a large number of terms performs poorly with the BLEU strategy as the number of higher N-grams that match with the query diminish within the document. Particular phrases are not necessarily present in the text; in fact, this possibility diminishes with a longer query.

5 Conclusions and Future Work

The improved results that we got in 2012 compared to 2011 can be attributed to better sentence scoring techniques used in 2012. The other aspect to be consider is that removing some tags such as references and external links may have contributed better to snippets in 2012, though a more thorough analysis is needed to decide what tags contribute to a good snippet. The other factor is using multiple focused elements in forming a snippet which led to good results.

Another interesting problem taken up during this work relates to finding the *top* n documents related to a query, given only its terminal node set. Though initial work has been done in this area, no definitive results have yet been achieved.

We find that sufficient ground has been established to conclude that good focused elements can be important in producing good snippets. We anticipate that predicting the document subset associated with the query will be used in snippet generation next year.

References

[1] Crouch, C. Dynamic element retrieval in structured environment. ACM TOIS, 24(4): 437-454, 2006.

[2] Geva, S., Kamps, J., Trotman, A. About INEX
https://inex.mmci.uni-saarland.de/about.html

[3] Geva, S., Kamps, J., Trotman, A. INEX Document Collection
http://www.inex.otago.ac.nz/data/documentcollection.asp

[4] Geva, S., Trotman, A., Scholer, F., Trappett, M., Sanderson, M. Overview of the inex 2011snippet retrieval track. In *Proceedings of the 9th international conference on Initiative for the evaluation of XML retrieval: comparative evaluation of focused retrieval*, INEX'11, Berlin, Heidelberg, 2011. Springer-Verlag.

[5] Khanna, S. Design and implementation of a flexible retrieval system.
Department of Computer Science, University of Minnesota, Duluth, 2008.

[6] Libxml2 parser http://www.xmlsoft.org/.

[7] Nagalla, S. INEX 2011 and 2012 Snippet Tasks.
Department of Computer Science, University of Minnesota, Duluth, March 2013.
http://www.d.umn.edu/cs/thesis/nagalla.pdf

[8] Narendravarapu, R. Improving Results for the 2009 and 2010 INEX Relevant in Context Tasks.
Department of Computer Science, University of Minnesota, Duluth, 2011.

[9] Salton, G., ed. *The Smart Retrieval System-Experiments in Automatic Document Processing*. Prentice-Hall,1971.

[10] Salton, G., Wong, A., Yang, C. *A Vector Space Model for Automatic Indexing, Comm.* ACM, 18(11), 613-620, 1975.

[11] XML specifications http://www.w3.org/XML/.

[12] Zhu, W., Papineni, K., Roukos, S., Ward, T. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (ACL '02). Association for Computational Linguistics, Stroudsburg, PA, USA, 311-318