CSIRO's Participation in INEX 2006

Alexander Krumpholz and David Hawking

CSIRO ICT Centre, Canberra, Australia
{Alexander.Krumpholz,David.Hawking}@csiro.au

Abstract. In this year's INEX participation, CSIRO took part in the Ad-hoc Track, contributing to three of the four given tasks, namely the Thorough Task, the Focused Task and the Best in Context Task.

We relied on our own text-and-metadata retrieval system PADRE for indexing the data and processing the queries. Since PADRE is designed to retrieve documents rather than XML elements we preprocessed the collection to enable retrieval of sub-elements. From the set of queries we identified all the element-types required to be retrieved. We then added new pseudo documents corresponding to all the retrievable elements from the originals. Finally, this expanded collection was indexed with PADRE.

When processing queries, query elements were extracted from the INEX topics and pre-processed to generate queries in PADRE syntax. Results were post-processed to achieve the requirements of each particular task, such as elimination of super- and sub-elements of elements already retrieved.

Results obtained from this simple-minded approach were not particularly competitive but serve as a good basis for identification of necessary future enhancements.

1 Introduction

CSIRO's involvement in the INEX2006[1] Ad Hoc Track[1] had three motivations. First, to explore the new Wikipedia collection[2], second to see how the naive splitting approach compares to current state-of-the-art XML retrieval engines[2], and third to establish a baseline for our future work.

PADRE, CSIRO's free-text and metadata retrieval system [4], implements a slightly modified Okapi BM25 algorithm [5], combined with Web-oriented evidence such as link counts, anchor text, URL length, and penalisation of multiple results from the same "site". In order for PADRE to retrieve sub-document parts like XML elements, the original XML files were split into smaller documents according to the XML elements relevant for retrieval. Thus, a much larger collection was indexed, comprising the original documents plus multiple, potentially overlapping, elements of those documents.

[1] Fully described elsewere in this volume.

[2] In our only previous participation (in 2002) [3] we successfully applied a similar approach but used manually generated queries, worked with a different collection and compared ourselves to a smaller, and presumably less sophisticated set of systems.

N. Fuhr, M. Lalmas, and A. Trotman (Eds.): INEX 2006, LNCS 4518, pp. 73–81, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In all of the Ad Hoc tasks, an ideal retrieval system would be able to identify all the passages of text which are relevant to the query. This may potentially require that it be capable of recognizing alternative forms of a query word (e.g. run, ran and running) and also synonyms etc. (e.g. sprinted, jogged). In our submitted runs we didn't use any forms of stemming or query expansion.

In the Focused task, an ideal retrieval system needs to be able to choose which of a hierarchical set of elements should be retrieved – an element at too high a level will include too much irrelevant material, while an element at too low a level may be only part of a passage spanning adjacent elements. We implicitly relied on the Okapi BM25 scoring algorithm to rank elements within the same hierarchy (and across hierarchies).

PADRE has the ability to index metadata using metadata classes, which are represented by a single character. Default metadata classes are used for HTML to map the title, author and date to different classes. This way a PADRE query can restrict the search for keywords to specific elements. For example the query "t:architecture" would only return pages having a title element matching the term "architecture". In addition to the default metadata classes defined, PADRE allows the user to specify project specific configuration files containing the mapping of XML elements (and attributes) to metadata classes. The xml.cfg mapping file used for this experiment is shown in Listing **??**.

2 Approaches to INEX Ad-Hoc Tasks

This year the INEX community defined four challenges for the Ad-hoc track:

Thorough Task: The goal of the Thorough Task is to retrieve a ranked list of elements over the whole collection, regardless of overlap.

Focused Task: The Focused Task also asks for elements ranked over the whole collection, but does not allow overlaps. Overlaps occur when multiple retrieved search results contain identical XML elements. This happens when a sub-element or super-element of an already returned retrieval result is returned.

All In Context Task: The All In Context task aims for getting a list of ranked documents including the relevant, non-overlapping elements within the document.

Best In Context Task: The goal is to return a list of ranked documents with the best entry point for a reader.

Retrieval of structured data is quite related to database query languages and requires selection and projection operations.

A summary of the different runs is shown in Table 1.

The selection is achieved using the PADRE retrieval engine. In a pre-processing step we converted the NEXI queries into different PADRE queries which could be used by the Query Processor. For the two main query types i.e. Context and Structure (CAS) and Context Only (CO) the PADRE version of the NEXI query could be used. However, the keywords in the CAS and the CO topics are not identical

(see Table 3). In order to explore the effect of this difference a third query has been constructed by removing the structural hints from each CAS topic.

The projection aspect has been addressed in a post-processing phase. Using PADRE and the collection of element level sub-documents, we based our runs on a number of considerations: the standard PADRE version based on the Okapi BM25 algorithm delivers exactly what the Thorough Task (A) requires. The simplest possible way to achieve the Focused Task (B) would be to run the same query used for case (A), while suppressing overlaps by skipping results which are in fact descendants (sub-elements) or ancestors (super-elements) of a previously returned result. In order to generate a baseline for the Best In Context Task we took the simple assumption that Wikipedia articles are covering very specific topics and that the article itself would be a reasonable entry point for a user.

The following eight runs were submitted:

Table 1. CSIRO's runs

Run name	Selection based on	Projection based on
CSIRO-CAS1-A	CAS-Title	n/a
CSIRO-CAS2-A	CAS-Title	query specified elements
CSIRO-CO1-A	CO-Title	n/a
CSIRO-CO1-B	CO-Title	suppression of overlapping results
CSIRO-CO1-D	CO-Title	<article> elements
CSIRO-CO2-A	CAS-Title without structure	n/a
CSIRO-CO2-B	CAS-Title without structure	suppression of overlapping results
CSIRO-CO2-D	CAS-Title without structure	<article> elements

3 Architecture

The architecture used is shown in Figure 1.

3.1 Defining Metadata Classes

The xml.cfg specifies the elements indexed and the PADRE metadata classes used to represent them. The DOC element is an artificial element introduced to create a collection of sub-documents extracted from the original XML file. (see section 3.2)

3.2 Splitting Documents

The original XML documents were split into sub-documents in order to exploit PADRE's document retrieval capability to actually retrieve sub-documents. We split each XML document of the collection into sub-documents by extracting all elements matching one of the following XPath expressions and storing them in a new XML container document:

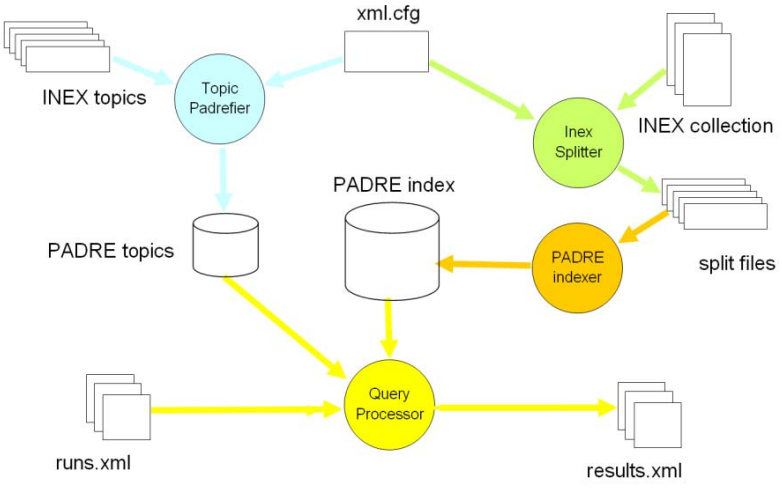

Fig. 1. Architecture

- /article
- //section
- //p
- //template

One problem with the current prototype is that sub-elements are not indexed with the super-element, i.e. metadata classes are disjunct. This would reduce the chance of a super-element being retrieved, because part of its text is effectively not indexed.

```
document,//DOC
a,1,,//DOC/article
b,1,,/body
e,1,,/caption
f,1,,/p
g,1,,/figure
j,1,,/title
l,1,,/table
n,1,,/template
o,1,,/section
p,1,,/template@name
q,1,,/collectionlink
r,1,,//DOC/article/name
w,1,,/link
x,0,,/DOCNO
```

Listing 1.1. The metadata classes defined in xml.cfg.

3.3 Transforming NEXI Topics

INEX topics are specified using the NEXI query language [6]. NEXI is XPath inspired, but allows the usage of vague selectors. The query `//section[about(.//p, security)]` matches sections that have a descendent element <p> containing the keyword 'security'.

Table 2. Comparison of the NEXI and PADRE query for topic 293

NEXI	//article[about(.,wifi)]//section[about(.,wifi security encryption)]
PADRE	a:wifi o:wifi o:security o:encryption

Table 3. A <title> and <castitle> element of topic 349

<title>	proprietary implementation +protocol +wireless +security
<castitle>	//article[about(., wireless)and about(.//p, security)] //link[about(., proprietary +implementation)]

```
<inex-submission participant-id="22" run-id="CSIRO-CO1-B"
task="Focused query="automatic">
  <topic-fields title="yes" castitle="no" description="no"
               narrative="no" ontopic_keywords="no"/>
  <description>Using the title as a query to padre,
              but suppressing overlapping results
  </description>
  <processing-instructions
    element-restriction="None"
    suppress="Overlap"
    padre-blocksize="1600"
  >
    <query>query.padre_title.uniq.join(' ')
    </query>
  </processing-instructions>
  <collections>
    <collection>wikipedia</collection>
  </collections>
</inex-submission>
```

Listing 1.2. The run configuration file 'run cfg'

In order to use the PADRE search engine we had to convert the official NEXI queries into PADRE queries. The script TopicPadrefier.rb (see figure 1) extracts the selection and projection component of each original CAS and CO topic, constructs the closest possible PADRE queries and stores them for future use. However, PADRE does not allow the definition of an infinite number of metadata classes and is therefore not expressive enough to correctly build PADRE queries for all possible NEXI queries. Some of the complex relationships of elements can therefore not be expressed in a generic manner.

3.4 Running INEX Topics

The result for each run has to be submitted as an XML file matching a DTD specified by the organizers. Since the Query processor needed processing instructions for each run, an extended version of the result DTD has been used as input for the Query processor. The Query processor extracts the processing instructions from the run specification file and replaces it with the result for the run. This way the run specification is automatically well documented.

3.5 Post-processing Results

The results have to be post-processed to meet the given tasks: i.e. suppressing multiple documents, or limiting to required element types or articles. For the Best In Context task all results other than articles have been suppressed in the runs CSIRO-CO1-D and CSIRO-CO2-D. For the Content and Structure task we suppressed all elements different than the structural request extracted from the query for the run CSIRO-CAS2-A, while we did not modify the result from PADRE for the runs CSIRO-CAS1-A, CSIRO-CO1-A, and CSIRO-CO2-A. For the runs CSIRO-CO1-B and CSIRO-CO2-B we matched each result with the results already delivered to filter out ascended or descended elements.

4 Results

Figures 2-4 show our results in respect to the other participant's submissions. Generally we obtained average results, with the following interesting observation: our structured results are worse than the unstructured ones. The graph and table in figure 2 show a significant gap between the CO*-A runs and the CAS*-A runs. (Note that the two CO*-A runs and the two CAS*-A lines in the graph essentially overlie each other.)

The tables list the runs submitted by CSIRO as well as the best and worst result for comparison. Notice, that the largest sample number at the bottom always shows the number of runs submitted for that task.

thorough		
Metric:ep-gr Quantization: gen,Overlap=off		
S.No	RunId	MAep
1		0.0709
33	CSIRO_CO2_A	0.0320
36	CSIRO_CO1_A	0.0313
78	CSIRO_CAS1_A	0.0122
80	CSIRO_CAS2_A	0.0111
106		0.0000

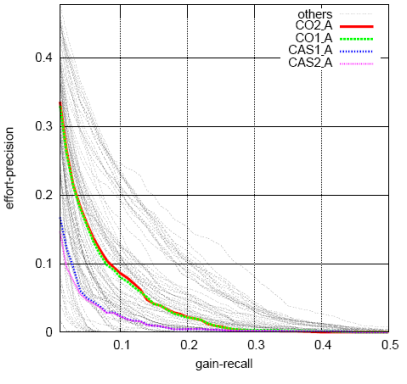

Fig. 2. Results: Thorough Task

All result sets show the related pairs of CSIRO runs almost side by side in the rankings, indistinguishable from each other in the small graphs attached. Figure 2 is the only one containing four runs by our team, one pair at position 33 and 36, the second pair at 78 and 80. We have not yet identified the reason why the Content Only runs clearly beat the Content And Structure runs. This is especially interesting since we would expect the additional information coming from the structure to improve retrieval quality. We suspect the disjunct metadata classes and the small number of element types we used for splitting contribute to that phenomenon.

The collocation of the other graphs are due to fact that the Context Only title and the Context And Structure title usually do not differ a lot.

The runs CAS1-A and CAS2-A share similar qualities as well. Apparently the element PADRE delivers automatically is often the one that would be specified in a CAS scenario.

focused		
Metric:nxCG Quantization: gen,Overlap=on		
S.No	RunId	nxCG@5
1		0.3944
18	CSIRO_CO1_B	0.3322
21	CSIRO_CO2_B	0.3304
85		0.0000

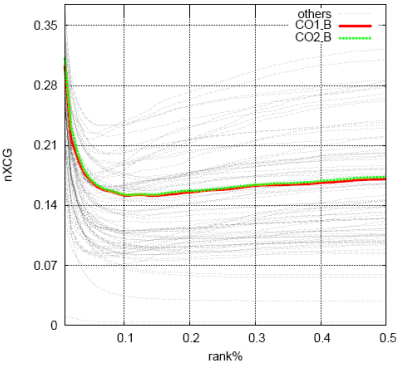

Fig. 3. Results: Focused Task

The Focused task turned out to be the best of our results, even though the approach of just suppressing duplicates is quite simplistic.

4.1 Failure/Success Analysis

Due to pressure from competing projects, we have been unable to make much progress on a failure/success analysis of our results.

We plan to construct some simple tools to facilitate comparison of our submitted result sets against the official results. We would like to identify cases:

Case 1 where we retrieved completely irrelevant items
Case 2 where we failed to retrieve items judged relevant
Case 3 where we retrieved items at higher levels in the XML hierarchy than the official answer.
Case 4 where we retrieved items at lower levels in the XML hierarchy than the official answer.

We are particularly interested in using these tools to answer the following questions:

BestInContext		
Metric: EPRUM-BEP-Exh-BEPDistance		
S.No	RunId	At A=0.01
1		0.0407
41	CSIRO_CO1_D	0.0166
43	CSIRO_CO2_D	0.0161
77		0.0000

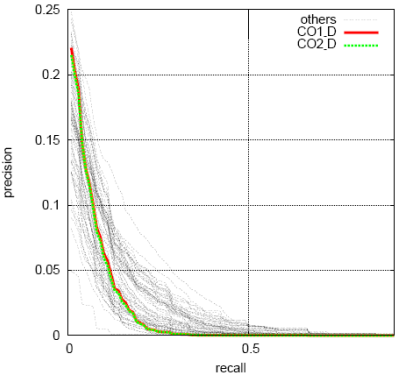

Fig. 4. Results: Best in Context Task

- Why were our attempts to exploit structure so unsuccessful? The scores for our CAS runs were substantially worse than our CO runs.
- For the Thorough task, Cases 1 and 2 are of particular interest. Did we need to apply stemming or query expansion? Could we exploit the available link structure? Were there bugs in our basic system or in pre and post processing scripts?
- For the Focused task, all four cases may adversely affect results.
- How much improvement might be gained by using a version of PADRE capable of indexing the same word occurrence as part of multiple overlapping elements?
- Were PADRE parameters set optimally? For example, were there adverse effects caused by the web-oriented ranking switched on by default in PADRE? Was the Okapi b parameter set to optimally take into account the relative length of documents?

5 Conclusions

One possible conclusion from our relatively unimpressive results this time is that the XML retrieval community has progressed significantly in recent years and that using a full text search engine to 'mimic' structured retrieval does not deliver competitive results.

This needs to be confirmed by the failure/success analysis described above.

We look forward to identifying the factors which are most significant in bridging the gap to state-of-the-art results.

References

1. DELOS Network of Excellence for Digital Libraries: Initiative for the Evaluation of XML retrieval `http://qmir.dcs.qmw.ac.uk/inex/`
2. Denoyer, L., Gallinari, P.: The Wikipedia XML Corpus. SIGIR Forum (2006)

3. Vercoustre, A.M., Thom, J.A., Krumpholz, A., Mathieson, I., Wilkins, P., Wu, M., Craswell, N., Hawking, D.: Csiro inex experiments: Xml search using padre. In: INEX Workshop 2002, Schloss Dagstuhl, Germany (December 2002)
4. Hawking, D., Bailey, P., Craswell, N.: Efficient and flexible search using text and metadata. Technical Report TR2000-83, CSIRO Mathematical and Information Sciences (2000) http://www.ted.cmis.csiro.au/~dave/TR2000-83.ps.gz
5. Robertson, S.E., Walker, S., Hancock-Beaulieu, M.M., Jones, S., Gatford, M.: Okapi at TREC-3. In: Harman, D.K. (ed.) Proceedings of TREC-3, Gaithersburg MD, pp. 500–225. NIST special publication (November 1994) http://trec.nist.gov/pubs/trec3/papers/city.ps.gz
6. Trotman, A., Sigurbjörnsson, B.: Narrowed extended xpath i (nexi). In: Fuhr, N., Lalmas, M., Malik, S., Szlávik, Z. (eds.) INEX 2004. LNCS, vol. 3493, pp. 16–40. Springer, Heidelberg (2005)