XML-structured documents: retrievable units and inheritance

Stephen Robertson[1], Wei Lu[2], and Andrew MacFarlane[3]

[1] Microsoft Research, 7 JJ Thomson Avenue, Cambridge, UK
and City University
ser@microsoft.com
[2] Center for Studies of Information Resources, School of Information Management, Wuhan University, China
and City University
sa713@soi.city.ac.uk
[3] Centre for Interactive Systems Research, Department of Information Science, City University, London, UK
andym@soi.city.ac.uk

Abstract. We consider the retrieval of XML-structured documents, and of passages from such documents, defined as elements of the XML structure. These are considered from the point of view of passage retrieval, as a form of document retrieval. A retrievable unit (an element chosen as defining suitable passages for retrieval) is a textual document in its own right, but may inherit information from the other parts of the same document. Again, this inheritance is defined in terms of the XML structure. All retrievable units are mapped onto a common field structure, and the ranking function is a standard document retrieval function with a suitable field weighting. A small experiment to demonstrate the idea, using INEX data, is described.

1 Introduction

In this paper, we explore the translation of some current ideas in text and document retrieval to an environment with XML-structured documents. To illustrate some of these ideas we make use of the INEX document collections and experiments. However, we first set the scene with an overview of a number of facets of retrieval in the domain of text documents, which may be unstructured or have minimal explicit structure.

2 Background

'Unstructured' text documents have been the subject of retrieval systems and experiments for many years. From experiments on small collections of scientific abstracts in the 1960s, through mid-size collections of news and news-wire materials to huge collections of heterogeneous web pages today, we have very

considerable experience and understanding of how to retrieve from such collections. Although there are very many variants and alternatives, the dominant approaches may be summarised as follows:

- Unstructured text queries;
- 'Bag-of-words' indexing of documents;
- Statistically-based scoring functions (to score each bag-of-words document against the current query);
- Ranked output

This is a broad characterisation – maybe a caricature – but does capture some of the important features of the experience and understanding. In particular, the very cavalier 'bag-of-words' approach, which ignores many intuitions and understandings about language and meaning in other contexts, nevertheless has been remarkably successful for search. On the other hand, the necessity for good scoring functions is paramount.

That said, in the following sections we explore some of the variants and alternatives which modify this dominant view, in preparation for discussing our approach to XML-structured documents.

2.1 Text and language

The notion that text itself is 'unstructured' is of course a gross over-simplification: text displays structures of extraordinary diversity and complexity. What is meant by the phrase 'unstructured text' in the document retrieval context is that the structure of text is relatively opaque to a text retrieval system, and also that the (difficult) extraction of linguistic structure has not on the whole shown much benefit in search. We must qualify this, of course, particularly with regard to 'search': the statement is true of traditional document retrieval (finding whole documents satisfying some 'relevance' criterion for a query), but not for other tasks in the broad information retrieval domain, such as query answering. The mainly factoid-based Question Answering track at TREC certainly requires more sophisticated analysis of language than is provided by the bag of words [1].

We observe that the methods employed in that task typically involve a first stage search employing traditional document retrieval techniques, followed by a more detailed analysis and extraction of possible answers from a small set of texts. The first stage is often passage-level (rather than document-level) retrieval. In fact, traditional techniques seem to work well down to some passage level.

2.2 Passage retrieval

The retrieval of (sub-document) passages has been an occasional interest within the document retrieval community for many years. One issue here has been that very few evaluation collections have relevance judgements at the passage level; however, passage retrieval has been tested as a method of doing document retrieval. Many of the problems and possibilities of using passage-level evidence, including all the ideas mentioned here, are discussed in [2].

At the simplest level, passages may be defined more-or-less arbitrarily (for example in terms of fixed word-length windows on the text, or by means of relatively superficial parsing such as sentence or paragraph separation). Then each document is retrieved on the basis of the score of the best-matching passage within it, rather than on the basis of scoring the entire document. The model is effectively that the passage is the relevant part of the document, but since we have relevance judgements at document level only, we have to embed the task in a document retrieval one.

A variation on this theme is to mix the scores of the best-matching passage and the whole document. We might see this as based on the same model (the passage is the relevant part), but with the additional assumption that the rest of the document also tells us something, provides some context, about the passage. Thus we evaluate the score of the passage, but allow it to *inherit* information from the document. This intuition informs the present paper.

2.3 Fields

One of the most obvious departures from the simplest bag-of-words model is the notion that some parts of the document/record are more informative than others. For a scientific paper, for example, we may guess that occurrences of words in the title or perhaps abstract are stronger indicators of the content of the paper than occurrences in the body text. We may operationalise this intuition by identifying a number of 'fields' or 'streams' in the documents – that is, a structure of fields that may be identified in or applied to every document in the collection. This is an important notion, whose value has been demonstrated both in formal experiments such as TREC and by the web search engines. In the latter case, the field structure must be applied to a very heterogeneous collection of documents. This implies a mapping of the features of each document type onto the common field structure. This mapping is typically a manual operation.

The interaction of fields and bag-of-words scoring methods is not entirely trivial. A common approach has been to construct a separate bag-of-words for each field, derive separate scores for the query against each field, and then combine them (usually a weighted linear combination) into a single document score. However, this approach conflicts with some characteristics of good bag-of-words scoring functions [3]. Instead, we need to construct a bag-of-words representation of the whole document which reflects the relative weights of the fields. Below, we use the BM25F method of doing this.

2.4 Inheritance between documents

One of the techniques which has amply proved its value in the web context is the inheritance of anchor text. Web search engines typically place heavy reliance on this technique. The principle is that an HTML link between documents (the `<a href="..."> ... </a>` tag) encloses a piece of text in the source document which describes the destination document. The technique is to associate this

piece of text with the destination document. Thus each document contains a single field which is the accumulation of all the anchor text from incoming links.

Experiments on web data [4] indicate that (despite the noise it may contain) this is an extremely valuable clue: anchor text is typically weighted highly, considerably higher than body text. Clearly, it depends on using a good form of field weighting.

2.5 Tuning

Any scoring function that contains free parameters must be tuned. For example, if we distinguish five fields, then we have four or five free field weights to choose (we may fix one of the fields at weight 1). Most basic bag-of-words scoring functions also contain their own free parameters.

The usual way to do this is to use some set of evaluation data (queries and relevance judgements) as a training set, and discover the combination of values of the free parameters that optimises some evaluation measure on this training set. There is a whole host of issues associated with this approach [5], which will not be further explored in the present paper.

3 Passages and inheritance in XML documents

We assume that we have text documents with XML structure. That is, the content of the documents is (largely or entirely) text, quite possibly including relatively large blocks of undifferentiated text like paragraphs, but with some overall XML structure. Blobs of text are held within XML elements.

3.1 INEX documents

We take the documents used in the INEX experiments to exemplify the idea; these are scientific papers and other contributions published in IEEE journals. They have XML structure that combines traditional structural elements (title, abstract, sections at various levels, bibliography etc.) with presentational matters (emphasis, font changes, mathematical material etc.). In fact structural and presentational matters are mixed with gay abandon: for example several different elements define 'paragraphs', either in particular presentation styles or with attributes which define the style.

The outline structure of an example INEX document is shown in figure 1. In this outline, lower levels of the structure and all text have been removed; multiply-occurring elements have been reduced to one or two.

It seems likely that some structural elements would be very useful for part-document retrieval. We might reasonably define sections or some level of sub-sections as suitable units for retrieval; possibly also other elements such as bibliographic references or figures. However, it is equally clear that not all the xml elements are suitable for this purpose. To take an extreme example, section titles in some documents look like this: `<st>C<scp>oncluding</scp>`

```
<article>
   <fno/>
   <doi/>
   <fm>
      <hdr> info about journal in which article appears </hdr>
      <tig> the title </tig>
      <au sequence="first"> 1st author name, affiliation </au>
      <au sequence="additional"> ditto for 2nd author </au>
      ...
      <abs> abstract as paragraphs of text </abs>
      <kwd> list of keywords </kwd>
   </fm>
   <bdy>
      <sec>
         <st> section title </st>
         paragraphs of text
      </sec>
      <sec>
         ...
         <ss1> (subsection)
            <st> subsection title </st>
            paragraphs of text; bulleted lists; figures etc
         </ss1>
         <ss1> ... </ss1>
      </sec>
      ...
   </bdy>
   <bm>
      <footnote id="T02001aff"> paragraphs etc. </footnote>
      <bib>
         <bibl>
            <h> header </h>
            <bb id="bibT02001"> bibliographic details </bb>
            <bb id="bibT02002"> ditto </bb>
            ...
         </bibl>
      </bib>
      <vt id="T0200a1"> author 1 details and picture </vt>
      <vt id="T0200a2"> ditto author 2 </vt>
      ...
   </bm>
</article>
```

Fig. 1. Outline of an example INEX document. Lower levels of the structure (including all presentation elements) and all text have been removed; many multiply occurring elements have been reduced in number; notes indicate the content of some elements in this example.

R<scp>emarks</scp></st>. The <scp> tag represents a font (small capitals), and the enclosed text is a fragment of a word only, quite unsuitable as a retrievable unit.

The section title as a whole might possibly be regarded as retrievable (separately from the section which it heads); this is at least arguable, though it is questionable whether the title actually contains information or merely contributes to the information in the section. However, as indicated above, in this paper we approach the issue from the point of view of passage retrieval, and hope to apply the kinds of methods which work at the document or passage level. It seems likely that a stand-alone section title is too short for such methods, so we reject this possibility.

Our choice is to make complete articles, sections, subsections at all levels and paragraphs into retrievable units. This is not to say that this choice is correct; it merely stresses the fact that this choice has to be made, and cannot simply be inferred from the XML structure. Retrievable units defined for our INEX 2005 experiments [6] included elements in the <bm> (back matter) tag and the abstract, but we might better consider these as ancillary matter, to be inherited as appropriate by other retrievable units, but not to be retrieved in their own right.

3.2 Inheritance

We have already discussed the notion of *fields* in a document. In this context, a very obvious field to consider is *title*. Complete articles have titles; sections and subsections also have or may have titles.

However, the interesting question arises concerning the relation between (say) the article title and the sections of the document. The article title describes the entire article, and therefore presumably also forms part of the description of each section or subsection within it. Similary, the main section title contributes to the description of the subsections.

Rather like the role of anchor text in web retrieval, we might reasonably assume that this information should be inherited appropriately. That is, if we consider a section as a retrievable unit, we should allow it to inherit information from outside itself that might be assumed to describe it. Thus sections should inherit article titles as well as their own; subsections should inherit both article and section titles. There may be other elements that should be inherited by the different levels: as indicated above, the abstract and bibliographic references are obvious examples.

The notion of inheritance may be compared to the mixing of whole-document score and passage score for passage retrieval; however, it is a more powerful and perhaps cleaner notion. It makes it quite clear that the passage (section, paragraph) is the retrievable unit; the role of the rest of the document is to tell us more *about the passage*. It also implies that if we are considering the passage for retrieval, we may choose to allow it to inherit selectively from the rest of the document, rather than simply inheriting its score.

3.3 Field structure

Following the discussion above, we would like to regard each retrievable item as a kind of field-structured document. Given a field structure, we have good methods for scoring the unit against a query. One issue, as indicated above, is that we need every retrievable item in the collection to be mapped onto *the same* field structure. Clearly, it is possible for a given field in a given item to be empty, but insofar as the different items will be competing against each other for retrieval, we need to ensure that items are balanced as far as possible.

In experiments for INEX 2005 [6], as an example of part of a mapping, we had a field for article title, for all retrievable elements. In other words, the article title was assumed to play the same role with respect to articles and to smaller units (sections and paragraphs). Section titles went to another field. Here we have reconsidered this mapping: a possible view is that a section title has the same relationship to the section in which it belongs as an article title has to the article. The relation of the article title to the section is different.

Another issue here is training. We need at the least a trainable weight for each separate field (possibly fixing one of them to 1). The more parameters, the more difficult training is and the more training data is needed. There is therefore good reason to start at least with a small number of fields.

Table 1 shows both the inheritance and field structure used in the present experiments. Again, it is not necessarily the best way to do it – it is offered as an example of how it might be done. Again, it is necessary to point out that this is a manually defined structure – we have no automatic methods of doing this.

	Field		
Retrievable element	Current title	Parent title(s)	Body text
article	article title	–	all text in article
section	section title (if present)	titles of article and all ancestor sections	all text in section
paragraph	–	ditto	all text in paragraph

(a dash indicates that the field is left empty)

Table 1. Retrievable elements mapped onto fields

4 Experiments

4.1 Data

The general XML structure of INEX documents has already been described. We want to try out some of the above ideas with INEX test data, including topics

and relevance judgements. Because we want to retrieve appropriate elements, we adopt an INEX-like methodology [7]. The methodology of INEX has evolved over the years, and the data collected (specifically relevance judgements) has evolved along with it. The most detailed relevance judgements on elements was made in INEX 2005. We therefore wish to use the INEX 2005 data, specifically that used for the CO (content only) task.

The INEX 2005 data consists of 16819 articles, 40 topics and relevance judgements (see further below on the topics and judgements). Under our definition of retrievable unit (article, section, all levels of subsection and all types of paragraph), we identify 1529256 retrievable units in total.

We also need to perform some training in order to establish suitable values for the free parameters, particularly field weights. As is usual when training parameters, we want to test the resulting trained algorithms on a different test set, to avoid overfitting. The set of topics for INEX 2005 CO is quite small (40 topics), which makes a training-test split somewhat problematic; for this reason the results reported here should be taken as indicative only. We split the 40 topics pseudo-randomly into 30 for training and 10 for test (that is, every 4th query was used for test). We note also that in fact only 29 of the 40 topics have relevance judgements, so in effect we have 20 for training and 9 for test. Thus the test results reported below are for the 10 test topics, but based on relevance judgements for only 9 of them. We use the standard INEX tool for evaluation.

4.2 Ranking function and baselines

We use the BM25F function mentioned above, as used in [3]. The BM25 parameters k_1 and b are taken as field-independent and fixed at $k_1 = 0.75$ and $b = 0.75$ (unlike in [4], where b_f is taken as being field-specific and all parameters are tuned). BM25 also makes use of the average document length; in this context it would make sense to interpret this as the average length of a retrievable unit. However, for simplicity we have used average article length in the present experiments.

We compare our present runs to the following:

- BM25 without field weights and without inheritance. For this we include the current title in with the section being retrieved – that is, we give the current title a weight of 1, the same as body text. Parent title is not included.
- Our INEX 2005 runs. These were based on training of field weights at document level only. The field structures used in those experiments is slightly different from the present one.

All these runs are evaluated on the test set of 10 topics used in the present experiments.

4.3 Performance measures

INEX uses eXtended Cumulated Gain (XCG) Metrics as its official INEX 2005 metrics [8]. The XCG metrics are a family of metrics that are an extension of

the cumulated gain (CG) based metrics which include the user-oriented measures of normalised extended cumulated gain (nxCG) and the systemoriented effort-precision/gain-recall measures (ep/gr). nxCG is a nomalised xCG measure. For a given rank i, the value of nxCG[i] reflects the relative gain the user accumulated up to that rank. Each measure includes 3 quantization measures: quant(strict), quant(gen) and quant(genLifted).

In inex 2005, relevance assessments are given according to two relevance dimensions: exhaustivity and specificity. Exhaustivity is measured using 4 levels: highly exhaustive, somewhat exhaustive, not exhaustive and too small. Specificity is measured on a continuous scale with values in [0, 1], where s = 1 represents a fully specific component.

For quant(strict) measure, only fully specific and highly exhaustive components are considered worthy; quant(gen) measure considers all relevant elements except "too small"; and quant(genLifted) measure considers all relevant elements but adds +1 to lift all exhaustivity values.

In our experiment the main measure used is nxCG(50) quant(strict).

4.4 Tuning

Using the very simple set of fields and inheritance indicated above, and also using the above standard values for the BM25 parameters, we train by a simple grid search on a range of values of each parameter. The two parameters we train are the weight of the 'Current title' field wf(cur) and that of the 'Parent title' field wf(par) (the weight of the 'Body text' field is set to 1). Thus we train over a 2-dimensional grid.

The granularity of the grid is significant. We first tune over a relatively coarse grid, to obtain an approximate optimum. We then fine-tune within a smaller range, to obtain a better approximation.

A 3-dimensional plot of the performance surface over this grid (the second stage fine-tuning, for the strict-50 measure) is shown in Figure 2. It can be seen that the surface shows a peak at a very high weight for the current title, and a much lower one for the parent title, but both much higher than body text. It should also be noted that the surface is not very clean, showing various local optima. This is probably primarily a function of the small training set size (30 topics). Note also that the graph shows only a very small range of the performance axis – the differences shown are quite small.

4.5 Testing

We now look at the performance of our tuned runs on the 10 test topics, and compare this to the baselines: see Table 2. The INEX 1 run was the one that was tuned at document level only; the other two were guesses only. Tuned 1 and Tuned 2 use the two best parameter combinations according to the tuning presented in the previous section.

These results are at least encouraging, although they must be taken with a considerable pinch of salt, because of the small size of the test set. Also, even

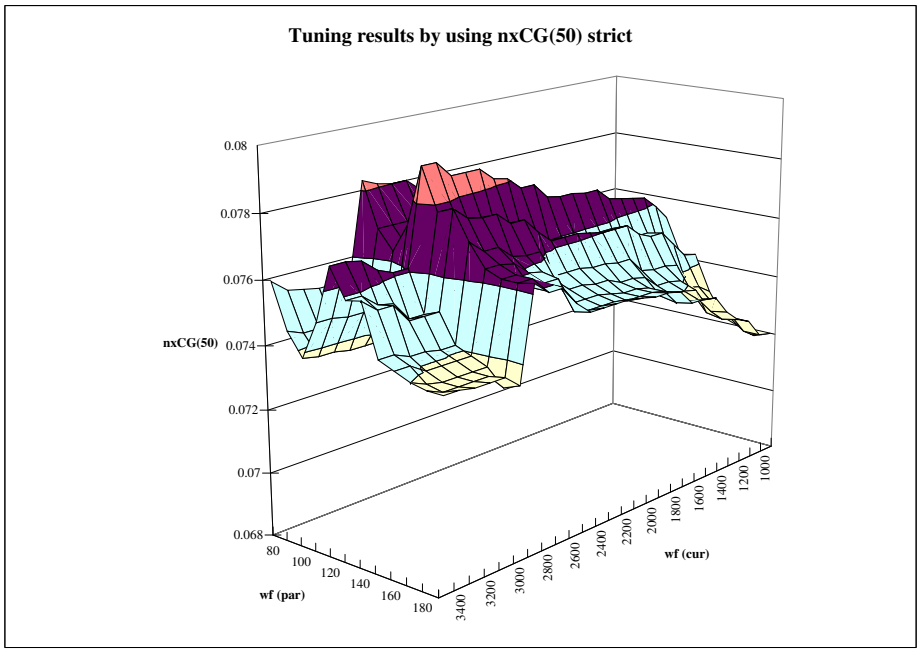

Fig. 2. Performance in the training set over a grid of values of the field weights.

	wf(cur)	wf(par)	nxCG(50) strict	change
Unweighted	1	0	0.0214	–
INEX 1	2356	22	0.0214	0%
INEX 2	1000	22	0.0218	+2%
INEX 3	15	8	0.065	-70%
Tuned 1	2800	120	0.0249	+16%
Tuned 2	2900	120	0.0249	+16%

Table 2. Results for the 'nxCG(50) strict' measure on the 10 test topics.

our best INEX run does not compare very well on this measure with the best runs by other participants.

Unfortunately, our results for the other measures (strict/general at different rank cutoffs) are not so clear. Probably the size of the test set is the limiting factor.

5 Conclusions

We have presented a way of thinking about retrieval in structured (specifically XML-structured) documents. We propose to treat element retrieval as similar to document or passage retrieval, but to allow elements to inherit information from other elements. This can be done in the following way. First, the elements that are to be made retrievable must be selected. Next, a simple flat field-structure is defined, a single structure applicable to all retrievable elements. Next, a mapping is defined from the document elements to the field structure, for each type of retrievable element. This mapping defines the inheritance: that is, which elements will inherit information from other elements.

We may then use a standard document scoring-and-ranking function which is capable of differential weighting of fields. This is then likely to need tuning on a training set.

A very small-scale experiment on INEX 2005 data has shown that this approach has some promise. First, it is possible to make a reasonable mapping of different elements onto a single flat field structure, and second, we have provided some experimental evidence that the inheritance idea is a useful one: in particular, it is useful for lower-level elements to inherit higher-level titles.

It is quite clear, however, that this test is a very partial one, and can be taken only as indicative in a small way. We expect in the future to conduct further and better-designed experiments to validate the ideas. These experiments should include much more detailed examination of the results, as well as being on a larger scale. Many questions remain, for example about how best both to choose inheritable elements and to map them onto a common field structure.

References

1. Voorhees, E.M.: Overview of the TREC 2004 question answering track. In Voorhees, E.M., Buckland, L.P., eds.: The Thirteenth Text REtrieval Conference, TREC 2004. NIST Special Publication 500-261. Gaithersburg, MD: NIST (2005) 52–69
2. Callan, J.: Passage-level evidence in document retrieval. In Croft, W.B., van Rijsbergen, C.J., eds.: SIGIR '94: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Springer-Verlag (1994) 302–310
3. Robertson, S., Zaragoza, H., Taylor, M.: Simple BM25 extension to multiple weighted fields. In Evans, D.A., Gravano, L., Hertzog, O., Zhai, C.X., Ronthaler, M., eds.: CIKM 2004: Proceedings of the 13th ACM Conference on Information and Knowledge Management, New York, ACM Press (2004) 42–49

4. Craswell, N., Hawking, D.: Overview of the TREC 2004 web track. In Voorhees, E.M., Buckland, L.P., eds.: The Thirteenth Text REtrieval Conference, TREC 2004. NIST Special Publication 500-261. Gaithersburg, MD: NIST (2005) 89–97
5. Taylor, M., Zaragoza, H., Craswell, N., Robertson, S.: Optimisation methods for ranking functions with multiple parameters. Submitted for publication (2006)
6. Lu, W., Robertson, S., MacFarlane, A.: Field-weighted XML retrieval based on BM25. INEX 2005; Submitted for publication (2006)
7. INEX: Initiative for the evaluation of xml retrieval. `http://inex.is.informatik.uni-duisburg.de/2005/` (Visited 13 February 2006)
8. Kazai, G., Lalmas, M.: INEX 2005 evaluation metrics. (Visited Feb. 22nd 2006) (2005)