Indexing Units

Jaap Kamps
University of Amsterdam
http://staff.science.uva.nl/~kamps/

SYNONYMS

None

DEFINITION

Indexing units refers to the granularity of information in the retrieval system's index, which can be in principle any document part of a structured text, and as a consequence determines the possible units of retrieval. There are three basic approaches: The first approach is to index every potentially retrievable unit as a whole—the so-called element-based approach [13]. The second approach is to index disjoint nodes—and relying on aggregation or score propagation methods for scoring higher-level nodes [e.g., 1, 12]. The third approach is to index only selected elements, for example by indexing particular element types in separate indexes [10]. Various mixtures of these approaches have also been applied.

All approaches make implicit or explicit assumptions on the (most likely) *unit of retrieval*. Although there may be no designated retrieval unit (such as the document or root node of the structured document), this also does not mean that every document part (such as a sub-tree of the structured document) is an equally desirable retrieval unit. Such assumptions may be relatively generic (such as paragraphs and sections being more informative than very short excerpts in bold or italics) or may depend on the query at hand (such as a structured query requesting elements with a particular tag). In all cases these assumptions depend on the sort of structured documents (which may range from strict XML databases to loosely structured textual documents with mark-up), and on the sort of information need (which may range from a strict database query with well defined semantics to a vague information retrieval topic of request). Structured text retrieval typically deals with loosely structured textual documents and vague information retrieval queries.

HISTORICAL BACKGROUND

Structured text retrieval has a long pre-history in text retrieval. The first test collections consisted of short document surrogates such as bibliographic descriptions (in various forms) or abstracts. Since the 1990s, test collections consisted predominantly of full text newspaper and newswire data. Interestingly, the bibliographic descriptions were highly structured catalogue records, and the newspaper data were typically structured in SGML, yet no particular use was made of the internal document structure. In fact, the use of the internal structure was usually explicitly outlawed, the main motivation for this being the desire to develop retrieval techniques that would work on all (flat) texts.

The use of document structure derived from SGML mark-up was pioneered by Wilkinson [15], studying ad hoc SGML element retrieval, and by Myaeng et al. [11], exploring structured queries that contain references to the SGML document structure. Similar early work on HTML is in [4]. There are also many similarities with the early work on multimedia retrieval, such as the DOLORES system [5] and the FERMI model [3], which are addressing the problem of retrieving information from structured documents. Ad hoc XML element retrieval and best entry point retrieval was studied in the Focus project [9]. In recent years, the main thrust of research in structured text retrieval is the annual initiative for the evaluation of XML retrieval, INEX [7].

SCIENTIFIC FUNDAMENTALS

Structured text retrieval typically deals with loosely structured textual documents and vague information retrieval queries, and the discussion is focused exclusively on this case. Indexing structured texts presents a number of challenges since such documents can be decomposed according to their internal structure. XML documents have a hierarchical structure of nested elements (or subtrees). That is, for example, an entire article consisting of front matter, body, and back matter. The body, in turn, consists of sections. The sections, again, consist of subsections. The subsections of paragraphs, and so on. Since there is no fixed unit of retrieval it is an open question what should be put in the index.

A prototypical structured text retrieval task is XML element retrieval. In text retrieval, evaluation is based on a "frozen" set of search requests (or "topics") with a set of known relevant results—XML elements regarded as relevant by the topic author. In XML retrieval, topics consist of a short keyword (or content-only) title; a structured (or content-and-structure) query in NEXI [14]; a single sentence description; and a long narrative statement of the search request. The retrieval system takes as *input* the standard keyword query, or the structured NEXI query. To give a concrete example, the ⟨title⟩ field of the INEX 2004 topic number 104 is

> *Toy Story*

The requested *output* is a ranked list of document components (in this case XML elements in the INEX IEEE collection containing full-text articles). There is no fixed *unit of retrieval*: if a whole article is relevant, return the ⟨article⟩ element, but if only a section is relevant, return the ⟨sec⟩ element. Retrieval systems will, of course, rank the XML elements based on the occurrences of query terms (or possibly phrases, stems, or synonyms based on these terms). However, whether a result is indeed relevant for the query is determined by a human judgment on whether the information in the element satisfies the topic author's information need. For the above mentioned INEX 2004 topic 104, the ⟨narrative⟩ field reads

> *To be relevant, a document/component must discuss some detail of the techniques or computer infrastructure used in the creation of the first entirely computer-animated feature-length movie, "Toy Story."*

A human judge (usually the topic author) will assess the relevance of the retrieved results, where relevant means that the element is both *exhaustive* (it provides useful information on the topic of request) and *specific* (there is a minimal amount of off-topic material).[1]

Although, in principle, any document part or XML element can be retrieved, some document parts tend to be more likely to be relevant. Table 1 shows the distribution of relevant elements over tag-names.[2] In this case, most frequently paragraphs (⟨p⟩), sections (⟨sec⟩), and subsections (⟨ss1⟩, ⟨ss2⟩) are judged relevant, but also entire articles (⟨article⟩). The precise tags are a direct result of the particular mark-up structure of the IEEE collection, which is mostly based on the logical structure of the articles. Generalizing over the particular tag-names, there is also a suggestion on the granularity of information that is most likely to be a relevant result. The following two observations present themselves. First, the most frequent elements such as paragraphs and (sub)sections are of relatively short-length. Second, there is great variety of elements regarded as relevant. Even for a single topic there is a very similar variety of elements, making clear that relevancy is both depending on the topic of request, and on the precise structure of the document at hand.

Recall the question of what to put in the index. The most obvious approach is to index all information in the structured text. But already here different options present themselves, as is illustrated in Table 2. On the left-hand side of Table 2 is a very simple example document, an article with title, abstract and two sections. The first approach, shown in the middle of Table 2, is to index every retrievable unit, in this case every XML element. In a "bag of words" approach all six XML elements of the document are indexed separately, but each with all the content or text contained inside the element. That is, an element is indexed with both the text nodes directly contained in it, and all text nodes of its descendants. The indexing of subtrees of the XML hierarchy is known as

[1] There have been different measures developed over the years, see the entry on Evaluation Metrics for Structured Text Retrieval for details.

[2] Here relevant is according to the strict quantization function, see Evaluation Metrics for Structured Text Retrieval for details.

Table 1: Distribution of relevant elements over tag names (reproduced from [8]).

2002 assessments			2003 assessments		
Tag-name	Frequency	%	Tag-name	Frequency	%
⟨p⟩	383	27.47%	⟨sec⟩	303	20.89%
⟨article⟩	309	22.16%	⟨p⟩	303	20.89%
⟨sec⟩	291	20.87%	⟨article⟩	172	11.86%
⟨ss1⟩	115	8.24%	⟨bdy⟩	167	11.51%
⟨bdy⟩	90	6.45%	⟨ss1⟩	146	10.06%
⟨ip1⟩	61	4.37%	⟨ip1⟩	69	4.75%
⟨ss2⟩	25	1.79%	⟨ss2⟩	36	2.48%
⟨abs⟩	22	1.57%	⟨fig⟩	32	2.20%
⟨fm⟩	13	0.93%	⟨app⟩	20	1.37%
⟨st⟩	11	0.78%	⟨bb⟩	19	1.31%
⟨item⟩	8	0.57%	⟨art⟩	18	1.24%
⟨app⟩	7	0.50%	⟨bm⟩	17	1.17%

Table 2: Example Document and Indexing Units

Example Document	Indexing subtrees	Indexing disjoint nodes
⟨article⟩	1. ⟨article⟩XXX YYY ZZZ VVV⟨/article⟩	1. ⟨title⟩XXX⟨/title⟩
⟨title⟩XXX⟨/title⟩	2. ⟨title⟩XXX⟨/title⟩	2. ⟨abstract⟩YYY⟨/abstract⟩
⟨abstract⟩YYY⟨/abstract⟩	3. ⟨abstract⟩YYY⟨/abstract⟩	3. ⟨sec⟩ZZZ⟨/sec⟩
⟨body⟩	4. ⟨body⟩ZZZ VVV⟨body⟩	4. ⟨sec⟩VVV⟨/sec⟩
⟨sec⟩ZZZ⟨/sec⟩	5. ⟨sec⟩ZZZ⟨/sec⟩	
⟨sec⟩VVV⟨/sec⟩	6. ⟨sec⟩VVV⟨/sec⟩	
⟨/body⟩		
⟨/article⟩		

the element-based approach [13]. Indexing subtrees is closest to traditional information retrieval since each XML node is a *bag of words* of itself and its descendants, and can be scored as ordinary plain text document. This directly relates structured text retrieval to the more general and well-understood problem of document retrieval. The indexing scheme is only using the structure to decompose the document into all retrievable units, and hence is applicable to any structured text. The main disadvantage is that it leads to a highly redundant index: text occurring at depth n of the XML tree is indexed n times.

On the right-hand side of Table 2 an alternative approach is shown, in which the text is only indexed once at the node where it occurs. The ⟨article⟩ and ⟨body⟩ elements are missing since they have no textual content. Since there are only four elements with content, the index is much smaller, and there is no redundancy of information. But the main advantage of indexing disjoint nodes is also creating a new problem of scoring higher level nodes. For example, as shown above, the whole article is a reasonably attractive XML element type, but it may not even occur in the index. This creates both a practical and a fundamental problem. The practical problem is that articles may contain thousands of XML elements, and hence may require considerable propagation of scores over the navigational axis. The fundamental problem is that it is not evident how to aggregate scores to ancestor elements, and various approaches exist in the literature. One of the earliest proposals is the augmentation approach of Abolhassani et al. [1], a straightforward propagation of scores to ancestor elements. The following simplified example illustrates the main idea behind augmentation. Let us assume the following document

⟨body⟩
  ⟨sec⟩cat...⟨/sec⟩
  ⟨sec⟩dog...⟨/sec⟩
⟨/body⟩

and a query consisting of the two terms "cat dog." Furthermore assume that: /body/sec[1] scores 0.7 for cat;

`/body/sec[2]` scores 0.4 for `dog`; and the rest 0 (that is, `dog` does not occur in the first section, and `cat` not in the second section). The problem is to determine the score of the body, which is not indexed itself. The *augmentation* approach propagates scores up with a certain weight (the augmentation factor) which is set to 0.3 based on experiments. The motivation for the augmentation factor is to avoid larger elements accumulating scores, and thus (almost) always get higher rankings than elements deep in the hierarchy. So in this case, the `/body[1]` will score $0.3 * 0.7 = 0.21$ for `cat`; and $0.3 * 0.4 = 0.12$ for `dog`. So the element `/body[1]`, although not in the index, will be returned. In fact, it will be the highest ranked result for "andish" query evaluation where only results containing all query terms are returned. A very similar approach is taken by the GPX model and its decay factor [6]. The element specific language models of Ogilvie and Callan [12] provide an elegant alternative approach within the language modeling framework. Here, every XML element forms a particular language model, and the ancestor elements are modeled as mixture language models of their direct children. A final alternative is to just propagate term frequencies and effective reconstructing the element-based index discussed above, e.g., using the region models of Burkowski [2].

A third indexing approach is to index only selected elements in their entirety. This is essentially the approach taken in the FERMI model of Chiaramella [3]. The selection is tailored to the collection at hand, usually based on the human interpretation of the tags in the collection. An example of this approach is to index particular types of elements separately [10]. Mass and Mandelbrod [10] create separate indexes for articles (⟨article⟩), abstracts (⟨abs⟩), sections (⟨sec⟩), subsections (⟨ss1⟩), sub-subsections (⟨ss2⟩), and paragraphs (⟨p⟩ and ⟨ip1⟩). Each index provides statistics tailored to particular components, which may be an advantage if language statistics deviate significantly between element types. This is essentially a distributed approach were queries are issued to all indexes, and the results of each of the indexes are combined after score normalization. The selective indexing approach turned out to be effective for the IEEE collection used at INEX. The requirement to select particular element-types makes it strongly collection-dependent, and it is less straightforward to apply this indexing approach to arbitrary structured text.

Three prototypical indexing approaches for structured text have been discussed above. Some observations present themselves. First, there is a trade-off between exploiting the document structure, and being generically applicable to all structured text. The element-based approach uses the document structure for decomposing documents by focusing on the hierarchical structure only. This ignores (potentially) useful structure like the tag-names, their attributes, the schema or DTD, etc. However, to phrase it positively, since it is completely schema-ignorant the approach can handle data with any type of tag-structure, even mixed-schema XML. The selective indexing approach is on the opposite side of the spectrum. Here, the specific tags and their semantics and importance have to be taken into account when selecting the element types to index. In cases where it is known what the more important elements are, this gives powerful handles to exploit this information. The downside is, of course, that the particular choice of element to index, and thereby the effectiveness of the approach, is completely dependent on the collection at hand. Second, there is a trade-off between indexing and query time complexity. The element based approach seems unattractive since its index is highly redundant: text appearing in a given element, will also appear in all the index entry for all the ancestors of this element. This may not be as undesirable as it may appear at first glance, since it can be viewed as a trade-off between query time and storage space complexity. The redundant index has essentially "precomputed" term frequencies per element, that otherwise need to be computed at query run time, and hence has relatively low query time complexity. The indexing complexity of the disjoint nodes approach requires much less storage space. However, an article may contain thousands of XML elements, and hence will require considerable propagation of scores over the navigational axes of the document at query time. Third, and finally, the indexing methods and retrieval models are standard information retrieval approaches or straightforward extensions of them.

KEY APPLICATIONS

Indexing units are a crucial component of structured text retrieval, and key applications are discussed in related entries on XML Retrieval and INitiative for the Evaluation of XML retrieval (INEX).

CROSS REFERENCE

Aggregation-based Structured Text Retrieval
Evaluation Metrics for Structured Text Retrieval
INitiative for the Evaluation of XML retrieval (INEX)
Propagation-based Structured Text Retrieval
XML Retrieval

RECOMMENDED READING

[1] M. Abolhassani, N. Fuhr, and S. Malik. HyREX at INEX 2003. In N. Fuhr, M. Lalmas, and S. Malik, editors, *INEX 2003 Workshop Proceedings*, pages 27–32, 2004.

[2] F. J. Burkowski. Retrieval activities in a database consisting of heterogeneous collections of structured text. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '92)*, pages 112–125, New York, NY, USA, 1992. ACM Press.

[3] Y. Chiaramella. Browsing and querying: Two complementary approaches for multimedia information retrieval. In N. Fuhr, G. Dittrich, and K. Tochtermann, editors, *Hypertext - Information Retrieval - Multimedia (HIM'97)*, pages 9–26. Universitätsverlag Konstanz, 1997.

[4] M. Cutler, Y. Shih, and W. Meng. Using the structure of HTML documents to improve retrieval. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, 1997.

[5] N. Fuhr, N. Gövert, and T. Rölleke. DOLORES: A system for logic-based retrieval of multimedia objects. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 257–265. ACM Press, New York, 1996.

[6] S. Geva. GPX – gardens point XML IR at INEX 2004. In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors, *Proceedings of the Initiative for the Evaluation of XML Retrieval (INEX 2004)*, volume 3493 of *LNCS*, pages 211–223. Springer Verlag, Heidelberg, 2005.

[7] INEX. INitiative for the Evaluation of XML Retrieval, 2007. `http://inex.is.informatik.uni-duisburg.de/`.

[8] J. Kamps, M. de Rijke, and B. Sigurbjörnsson. Length normalization in XML retrieval. In M. Sanderson, K. Järvelin, J. Allan, and P. Bruza, editors, *Proceedings of the 27th Annual International ACM SIGIR Conference*, pages 80–87. ACM Press, New York NY, USA, 2004.

[9] G. Kazai, M. Lalmas, and J. Reid. Construction of a test collection for the focussed retrieval of structured documents. In F. Sebastiani, editor, *Advances in Information Retrieval, 25th European Conference on IR Research (ECIR 2003)*, volume 2633 of *Lecture Notes in Computer Science*, pages 88–103. Springer, 2003.

[10] Y. Mass and M. Mandelbrod. Retrieving the most relevant XML components. In N. Fuhr, M. Lalmas, and S. Malik, editors, *INEX 2003 Workshop Proceedings*, pages 53–58, 2004.

[11] S. H. Myaeng, D.-H. Jang, M.-S. Kim, and Z.-C. Zhoo. A flexible model for retrieval of SGML documents. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 138–145. ACM Press, New York NY, USA, 1998.

[12] P. Ogilvie and J. Callan. Using language models for flat text queries in XML retrieval. In N. Fuhr, M. Lalmas, and S. Malik, editors, *INEX 2003 Workshop Proceedings*, pages 12–18, 2004.

[13] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. An Element-Based Approch to XML Retrieval. In N. Fuhr, M. Lalmas, and S. Malik, editors, *INEX 2003 Workshop Proceedings*, pages 19–26, 2004.

[14] A. Trotman and B. Sigurbjörnsson. Narrowed Extended XPath I (NEXI). In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors, *Proceedings of the Initiative for the Evaluation of XML Retrieval (INEX 2004)*, Lecture Notes in Computer Science, pages 16–40. Springer Verlag, Heidelberg, 2005.

[15] R. Wilkinson. Effective retrieval of structured documents. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 311–317. Springer-Verlag, New York NY, 1994.