XML Multimedia Retrieval

Zhigang Kong and Mounia Lalmas

Department of Computer Science, Queen Mary, University of London
{cskzg, mounia}@dcs.qmul.ac.uk

Abstract. Multimedia XML documents can be viewed as a tree, whose nodes correspond to XML elements, and where multimedia objects are referenced in attributes as external entities. This paper investigates the use of textual XML elements for retrieving multimedia objects.

1 Introduction

The increasing use of eXtensible Mark-up Language (XML) [6] in document repositories has brought about an explosion in the development of XML retrieval systems. Whereas many of today's retrieval systems still treat documents as single large blocks, XML offers the opportunity to exploit the logical structure of documents in order to allow for more precise retrieval.

In this work, we are concerned with XML multimedia retrieval, where multimedia objects are referenced in XML documents. As a kind of hypermedia with controlled structure, XML multimedia documents are usually organized according to a hierarchical (tree) structure. We believe that exploiting this hierarchical structure can play an essential role in providing effective retrieval of XML multimedia documents, where indexing and retrieval is based on any textual content extracted from the XML documents.

An XML document can be viewed as a tree composed of nodes, i.e. XML elements. The root element corresponds to the document, and is composed of elements (i.e. its children elements), themselves composed of elements, etc., until we reach leaf elements (i.e. elements with no children elements). An XML multimedia element, which is an element that references in an attribute a multimedia object as an external entity, has a parent element, itself having a parent element, all of them constituting the ancestor elements of that multimedia element. It can also have its own (i.e. self) textual content, which is used to describe the referenced multimedia entity. Our aim is to investigate whether “hierarchically surrounding” textual XML elements of various sizes and granularities (e.g., self, parent, ancestor, etc.) in a document or any combination of them can be used for the effective retrieval of the multimedia objects of that document.

The exploitation of textual content to perform multimedia retrieval is not new. It has, for example, been used in multimedia web retrieval [1,4]. Our work follows the same principle, which is to use any available textual content to index and retrieve non-textual content; the difference here is that we are making use of the hierarchical tree structure of XML documents to delineate the textual content to consider.

The paper is organized as follows. In Section 2, we describe our XML multimedia retrieval approach. In Section 3, we describe the test collection built to evaluate our approach. In Section 4, we present our experiments and results. Finally we conclude in Section 5.

2 Our Approach

A multimedia object is referenced as an external entity in the attribute of an XML element that is specifically designed for multimedia content. We call this element a multimedia element. Some textual content can appear within the element, describing (annotating) the multimedia object itself. The elements hierarchically surrounding the multimedia element can have textual content that provides additional description of the object. Therefore, the textual content within a multimedia element and the text of elements hierarchically surrounding it can be used to calculate a representation of the multimedia object that is capable of supplying direct retrieval of this multimedia data by textual (natural language) query.

We say that these hierarchically surrounding elements and the multimedia element itself form regions. The regions of a given multimedia object are formed upward following the hierarchical structure of the document containing that multimedia object: the self region, its sibling elements, its parent element, and so on; the largest region being the document element itself. We define the text content of the region used to represent the multimedia object as its region knowledge (RK).

As the elements of XML are organized in a hierarchical tree and nested within each other, the regions are defined as hierarchically disjoint. This is important to avoid repeatedly computing the text content. We therefore define N+1 disjoint RKs, where N is the maximum depth in the XML multimedia document collection:

- Self level RK: It is a sequence of one or more consecutive character information items in the element information item, which is a multimedia element in which the multimedia object is referenced as an external entity. This is the lowest level region knowledge of a given multimedia object.
- Sibling level RK: It is a sequence of one or more consecutive character information items in the element information items, which is at the same hierarchical level of the multimedia element and just before or after it.
- 1st ancestor level RK: It is a sequence of one or more consecutive character information items in the element information item, which is the parent element of the multimedia element, excluding those text nodes having been used for Self and Sibling RKs.
- …
- Nth ancestor level RK: It is a sequence of one or more consecutive character information items in the element information item, which is the parent of the element of N-1th ancestor level RK, excluding those text nodes having been used for its lower level RKs.

The RKs are used as the basis for indexing and retrieving XML multimedia objects. At this stage of our work, we are only interested in investigating whether regions can indeed be used for effectively retrieving multimedia XML elements. Therefore, we use a simple indexing and retrieval method, where it is straightforward to perform experiments that will inform us on the suitability of our approach. For this purpose indexing is based on the standard tf-idf weighting and retrieval is based on the vector space model [3].

The weight of term t in the RK is given by the standard tf-idf, where idf, is computed across elements (and not across documents) as it was shown to lead to better effectiveness in our initial experiments. The weight of a term in the combination of RKs, which is then the representation of a given multimedia object, is calculated as the weighted sum of the tf-idf value of the term in the individual RKs. The weight is the importance associated with a given RK in contributing to the representation of the multimedia object.

3 The Test Collection

To evaluate the effectiveness of our proposed XML multimedia retrieval approach, we requires a test collection, with its set of XML documents containing non-textual elements, its set of topics, and relevance assessments stating which non-textual elements are relevant to which topics. We used the collection developed by INEX, the INitiative for the Evaluation of XML Retrieval [2], which consists of 12,107 articles, marked-up with XML, of the IEEE Computer Society's publications covering the period of 1995-2002, and totaling 494 mega-bytes in size. On average an article contains 1,532 XML elements, where the average depth of an element is 6.9 [2]. 80% of the articles contain at least one image, totaling to 81,544 images. There is an average of 6.73 images per articles. The average depth of an image is 3.62.

We selected six volumes of the INEX document set, in which the average number of images per XML document is higher than in others. Due to resource constraint, we restricted ourselves to those articles published in 2001. The resulting document collection is therefore composed of 7,864 images and 37 mega-bytes XML text contained in 522 articles. There is an average of 15.06 images per article. On average, the depth of an image ranges between 2 and 8, where the average depth is 3.92. In addition, we calculated the distribution of the images across various depths (levels). 62% of images are at depth 4, 23.5% of them have depth 3, 13.5% of them have depth 5. If an image has depth 4, its highest level RK is a 4th level RK, etc. In most cases, 4th level RK corresponds to the article excluding the body elements (it contains titles, authors and affiliation, i.e. heading information, classification keywords, abstracts); 3rd level RK corresponds to the body element excluding the section containing the image; 2nd level RK corresponds to the section excluding the sub-section containing the image; and 1st level RK corresponds to the sub-section excluding the caption and sibling RKs of the image.

The topics designed for this test collection are modified versions of 10 topics of the original INEX 2004 topics. These topics were chosen so that enough relevant XML elements (text elements) were contained in the 522 XML articles forming the created collection. These topics have a total of 745 relevant elements, with an average of 74.5 relevant elements per topic. Each topic was modified so that indeed images were searched for.

Our topics are based on actual INEX topics, for which relevance assessments are available. As such, for a given query, only articles that contained at least one relevant element were considered. This simplified greatly the relevance assessment process. The assessments were based on images and their captions, and performed by computer science students from our department following the standard TREC guidelines [5]. The relevance assessment identified a total of 199 relevant images (out of 7,864 images), and an average of 20 relevant images per topic.

4 Experiments, Results and Analysis

The purpose of our experiments is to investigate the retrieval effectiveness of the so-called RKs for retrieving multimedia objects referenced in XML elements, which in our test collection are image objects. Our experiments include self, sibling, and 1st ancestor level up to 6th ancestor level RKs, used independently or in combination to represent multimedia XML elements. We report average precision values for all experiments. The title component of the topics was used, stop-word removal and stemming were performed.

4.1 Individual RKs

Experiments were performed to investigate the types of RKs for retrieving image elements. The average precision values for self level, 1st level, ..., to 6th ancestor level RKs are, respectively: 0.1105, 0.1414, 0.1403, 0.2901, 0.2772, 0.2842, 0.0748, and 0.0009. Therefore using lower (self, sibling, 1st) level RK leads to low average performance. The reason could be two-fold: (1) the text content in self level RK are captions and titles that are small so the probability of matching caption terms to query terms is bound to be very low – the standard mismatch problem in information retrieval; and (2) captions tend to be very specific – they are there to describe the images, whereas INEX topics may tend to be more general, so the terms used in captions and the topics may not always be comparable in terms of vocabulary set.

The 2nd, 3rd, and 4th level RKs give the best performance. This is because they correspond to regions (1) not only in general larger, but also (2) higher in the XML structure. (1) seems to imply - obviously - that there is a higher probability to match query terms with these RKs, whereas (2) means that in term of vocabulary used, these RKs seems to be more suited to the topics. Furthermore, 2nd level RKs perform best, meaning usually the sections containing the images, and then 4th level RK performs second best, meaning the heading information, abstract and reference of the article containing the images.

Since a large number (i.e. 62%) of image objects in the INEX collection are within lower level elements (i.e. depth 4), the images within a document will have the same 4th level RK, i.e made of same abstract and heading elements. Our results seem to indicate that retrieving all the images of a document whose abstract and heading match the query is a better strategy than one based on exploiting text very near to the actual image.

Performance decreases when using higher levels RKs. This is because most images have a 4th level RK (since they have depth of 4), much fewer have 5th level RK (13.5%), and less than 1% have a 6th level RKs. Thus nothing should be concluded from these results. We therefore do not discuss performance using these RKs.

We also looked at the amount of overlaps between the images retrieved using the various RKs (i.e. percentage of retrieved images that were also retrieved by another RK). Although not reported here, our investigation showed that a high number of images retrieved using the self RK are also retrieved using most of the other RKs. The reverse does not hold; many of the images retrieved using 2nd level RK are not retrieved using smaller RKs. This would indicate that higher level RKs have definitively an impact on recall. The "many but not all" is a strong argument for combining low level and high level RKs for retrieving multimedia objects.

4.2 Combination of RKs

This section investigates the combinations of various RKs for retrieving multimedia objects. The combinations are divided into three sets: (1) combinations from lower level up to higher level with the same weights for each participating level, i.e. self RK is combined with sibling RK, and together they are combined with 1st ancestor level RK, etc; (2) combinations of self, sibling, 1st up to 3rd level RKs, with the 4th ancestor level with the same weight for each participating level – level 4 RK was chosen as it led to good performance in Section 4.1 (although images within a document could be differentiated); and (3) combinations of all level RKs but with different weights to each level.

The average precision values for the first set are: 0.1832, 0.2329, 0.2748, 0.2897, and 0.3716. We can see that performance increases when a lower level RK is combined with an upper level RK. In addition, we can see that the combinations up to 4th level RK obtain much better performance than any single level (i.e. 2nd: 0.2901 and 4th: 0.2842). These results show clearly that combining RKs in a bottom-up fashion lead to better performance, as they indicate that the RKs seem to exhibit different (and eventually complementary) aspects, which should be combined for effective retrieval.

The average precisions for our second set of experiments are: 0.3116, 0.3061, 0.3336, 0.3512, and 0.3716, which are very comparable. We can see that by combining the self RK with the 4th ancestor level leads already to effectiveness higher than when using any single level RK. As discussed in Section 4.1, using the 4th level RK retrieves all the images in a document (as long as the RK matches the query terms), so our results show that using in addition lower level RKs - which is based on elements closer to the multimedia objects and thus will often be distinct for different images - should be used to differentiate among the images in the document.

To further justify our conclusion, that is to make sure that our results are not caused by the way our test collection was built, we looked at the relevant elements for the original topics in the INEX test collection: 3.7% of them are at the document level, whereas 81% have depth 3, 4 and 5. Therefore, this excludes the possibility that the document level RKs (4th level RK combined with all lower level RKs) lead to the best strategy for XML multimedia retrieval in our case, because they were the elements assessed relevant to the original topics. This further indicates that higher level

RKs seem best to identify which documents to consider, and then using lower level RKs allows selecting which images to retrieve in those documents.

The last set of experiments aims to investigate if assigning different weights to different levels can lead to better performance. We did four combinations (including all RKs) and the average precisions for the four combinations are: 0.3904 (same weight to every level), 0.3796 (emphasize lower level RKs), 0.3952 (emphasize higher level RKs), and 0.3984 (emphasizes 2nd and 4th level RKs). The performances are better when weights are introduced - compared to previous experiments, although there is not a great difference with the various weights. However, this increase could also be due to the fact that the 5th and 6th ancestor level RKs are used, which corresponds for some (few) images to the abstract and heading elements, which were shown to lead to good performance.

5 Conclusions and Future Work

Our work investigates the use of textual elements to index and retrieve non-textual elements, i.e. multimedia objects. Our results, although based on a small data set, show that using elements higher in a document hierarchical structure works well in selecting the documents containing relevant multimedia objects, whereas elements lower in the structure are necessary to select the relevant images within a document. Our next step is to investigate these findings on larger and different data sets, as that being built by an XML multimedia track as INEX 2005.

References

1. Harmandas, V., Sanderson, M., & Dunlop, M.D. (1997). Image retrieval by hypertext links. Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval, Philadelphia, US, pp 296–303.
2. INEX. Initiative for the Evaluation of XML Retrieval http://inex.is.informatik.uni-duisburg.de/
3. Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic retrieval. Information Processing & Management, 24(5):513-523.
4. Swain, M. J., Frankel, C., and Athitsos, V. (1997). WebSeer: An image search engine for the World Wide Web. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (San Juan, Puerto Rico).
5. Text REtrieval Conference (TREC). http://trec.nist.gov/
6. XML (eXtensible Markup Language). http://www.w3.org/XML/