

# An IR Approach to XML Retrieval based on the Extended Vector Model

Carolyn J. Crouch  
Department of Computer Science  
University of Minnesota Duluth  
Duluth, MN 55812  
(218) 726-7607  
ccrouch@d.umn.edu

S. Apte  
Department of Computer Science  
University of Minnesota Duluth  
Duluth, MN 55812  
(218) 726-7607  
apte0002@d.umn.edu

H. Bapat  
Department of Computer Science  
University of Minnesota Duluth  
Duluth, MN 55812  
(218) 726-7607  
bapa0005@d.umn.edu

## ABSTRACT

The authors describe their approach to XML retrieval based on the extended vector space model of Fox [3]. Complete implementation of the system, using the Smart experimental retrieval system, is currently underway.

## Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Retrieval Models.

## General Terms

Design, Experimentation

## Keywords

XML retrieval, extended vector model

## 1. INTRODUCTION

When Vannevar Bush [1] first envisioned an ability to retrieve relevant information while sitting at his desk—that is, to have the data he sought immediately available at his fingertips—one could argue that he was in fact foreseeing the capabilities available to today's researchers through the development of facilities based on hypertext, multimedia, networking, and theoretical and applied research in information retrieval, among others. As XML becomes more dominant in the representation of web documents, it is a natural extension for information retrieval research.

When we first became familiar with the XML task posed by INEX, we were struck by the similarity of portions of the task to earlier work we had done [2] based on the extended vector model proposed by Fox [3]. Since our interests lie in information retrieval, we chose this approach for our initial investigations in XML retrieval.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

It seems safe to assume that everyone in the retrieval community is familiar with the vector space model [6], developed by Salton and used so successfully over the years by both his researchers at Cornell and others. This model is the basis of the Smart retrieval system, which evolved over a 30 year period under the direction of Salton, Buckley, and others. In the vector space model, each document (and query) is viewed as a set of unique words or phrases and is represented as a weighted term vector. The weight assigned to each term is indicative of the contribution of that term to the meaning of the document. The similarity between vectors (e.g., document and query) is represented by the mathematical similarity of their corresponding term vectors.

Fox [3], recognizing that the vector space model could be modified to include concepts other than the normal content terms, extended it as follows. He developed a method for representing in a single, extended vector different classes of information about a document, such as author name, terms, bibliographic citations, etc. Here a document vector consists of a set of subvectors, where each subvector represents a different concept class or c-type. Similarity between extended vectors is calculated as a linear combination of the similarities of corresponding subvectors. Subsequent work by both Fox and others [4, 2] focused on the problem of automatic generation of extended queries in this domain. (Of course, if we utilize the extended vector model for XML retrieval, this particular problem is no longer an issue because the query that is given can easily be translated into an extended vector query.)

The XML experiments are designed to handle two types of queries: the content-only (CO) query (the traditional query in information retrieval) and the content-and-structure (CAS) query. For CO queries, the retrieval system is expected to return a ranked list of the most relevant elements (article, paragraph, section, etc.). No target element is specified. For the CAS queries, the retrieval system should return a ranked list of elements specified in the target element (<te>) field, rather than a ranked list of documents. Search words themselves are specified in the <cw> element, and the context of the search words is specified in the context element (<ce>). In a relevant document, the search words in the <cw> element should occur in the element specified in the <ce>. Otherwise (if no <ce> is specified), the search words can occur anywhere in the document.

## 2. OUR APPROACH

In our approach, based on Fox's extended vector model, documents and queries are represented in extended vector form. The extended vector itself is a combination of subvectors, some containing normal text and others containing objective identifiers associated with the document. (Our current representation of an XML document/query consists of 18 subvectors.) For CO queries, we chose at this point to return a ranked list of documents. Keywords are not confined to a specific context, and we search for them throughout the document. The challenge for those using vector-based systems like ours is to deal with CAS queries, which consist of pairs of `<cw>`, `<ce>` elements. Consider, for example, the title section of CAS query 8:

```
<title>
  <te>article</te>
  <cw>ibm</cw><ce>fm/aff</ce>
  <cw>certificates</cw><ce>bdy/sec</ce>
</title>
```

In this case, the query is to return a ranked list of articles as specified by the target element `<te>`. The narrative specifies that the body or sections of relevant documents should contain information about the use of certificates for authenticating users on the Internet. And since the context of the content word *ibm* is *fm/aff*, the author(s) of those documents must be affiliated with IBM. Thus the query should retrieve only those articles on the use of certificates whose author(s) are affiliated with IBM.

The vector space model is not designed to handle this essentially Boolean query. Direct use of the extended vector model does not guarantee that each keyword will occur in the specified context. We deal with this issue by splitting the query into two parallel queries as follows:

Query 1: `<cw>ibm</cw><ce>fm/aff</ce>`

Query 2: `<cw>certificates</cw><ce>bdy/sec</ce>`

Affiliation and section are two different c-types. So query 1 searches for documents containing the objective identifier *ibm* in the affiliation subvector. Query 2 seeks documents whose section(s) contain the subjective identifier *certificate*. Our retrieval system (Smart) returns a ranked list of documents for both queries. The intersection of these lists is the final, ranked list of documents returned for query 8. Our retrieval is based on untuned *Lnu.ltu* [7] weighting of the collection.

## 3. CONCLUSIONS

Our efforts to date have been limited by the small size of our team, the substantial commitment required to produce the team's contribution to the XML project, the deferral of deadlines to meet the needs of the participants and the restrictions of the academic schedule. We will defer our discussion of results until the INEX tool for evaluation is available.

Our current efforts center on the implementation, within the extended vector model, of providing what is quite accurately referred to as "flexible retrieval"—i.e., "the retrieval over arbitrary combinations and nestings of element types"—by Grabs and Schek [5]. An excellent description of this task within the vector space model may be found in this reference. A related issue involves weighting within the local environment, weighting within the larger XML collection, and weighting among subvectors in the extended vector model.

## 4. REFERENCES

- [1] Bush, V. As we may think. The Atlantic online. <http://www.theatlantic.com/unbound/flashbks/computer/bush.html>.
- [2] Crouch, C. J., Crouch, D. B., and Nareddy, K. R. The automatic generation of extended queries. In Proc. of the 13<sup>th</sup> Annual International ACM SIGIR Conference, (Brussels, 1990), 369-383.
- [3] Fox, E. A. Extending the Boolean and vector space models of information retrieval with p-norm queries and multiple concept types. Ph.D. Dissertation, Department of Computer Science, Cornell University (1983).
- [4] Fox, E. A., Nunn, G. L., and Lee, W. C. Coefficients for combining concept classes in a collection. In Proc. of the 11th Annual International ACM SIGIR Conference, (Grenoble, 1988), 291-307.
- [5] Grabs, T. and Schek, H. Generating vector spaces on-the-fly for flexible XML retrieval. <http://www-dbs.inf.ethz.ch/~grabs>.
- [6] Salton, G., Wong, A., and Yang, C. S. A vector space model for automatic indexing. *Comm. ACM* 18, 11 (1975), 613-620.
- [7] Singhal, A., Salton, G., Mitra, M., and Buckley, C. Document length normalization. In *IP&M* 23, 5 (1996), 619-633.