Adapting the Extended Vector Model for XML Retrieval

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

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

Harsh Bapat
Persistent Systems Pvt. Ltd.
Pune, India 411016
+91 (20) 567 8900
harsh_bapat@persistent.co.in

ABSTRACT
In this paper, we describe our approach to XML retrieval, which is based on the extended vector space model initially proposed by Fox [5]. The current implementation of our system and results to date are reported. The basic functions are performed using the Smart experimental retrieval system. Early results confirm the viability of the extended vector space model in this environment.

1. INTRODUCTION
When we began our work with INEX last year, our goal was to confirm the utility of Salton's vector space model [10] in its extended form for XML retrieval. Long familiarity with Smart [9] and it capabilities led us to believe that it could be used for this purpose. Our approach was described in the Proceedings of last year's workshop [3]. Much initial effort was spent on the translation of documents and topics from XML to internal Smart format and the subsequent translation of results back into INEX format. When we reported our results in [3], our system was still in a very rudimentary stage and far from error-free.

During the past year, we built upon and extended this work. We now have an operational system. For the sake of clarity, a brief overview follows.

1.1 Background
Everyone involved in information retrieval is familiar with the vector space model, wherein documents and queries are represented as weighted term vectors. The weight assigned to a term is indicative of the contribution of that term to the meaning of the document. Very commonly, *tf-idf* weights [11] or some variation thereof [12] are used. The similarity between vectors (e.g., document and query) is represented by the mathematical similarity of their corresponding term vectors.

In 1983, Fox [5] proposed an extension of the vector space model—the so-called extended vector space model—to allow for the incorporation of objective identifiers with content identifiers in the representation of a document. An extended vector can include different classes of information about a document, such as author name, bibliographic citations, etc., along with content terms. In this model, a document vector consists of a set of subvectors, where each subvector represents a different class of information (i.e., concept class or c-type). Our current representation of an XML document/query consists of 18 c-types (i.e., *article, ti, atl, pub_yr, sec, st, fgc, article_au_fnm, article_au_snm, abs, kwd, ack, tig, bibl_au_fnm, bibl_au_snm, bibl_ti, bibl_atl, p*) as defined in INEX guidelines. Similarity between extended vectors is calculated as a linear combination of the similarities of corresponding subvectors.

Use of the extended vector model for document retrieval normally raises at least two problems: the construction of the extended search request [4, 6] and the selection of the coefficients for combining subvector similarities. For XML retrieval, of course, the query is posed in a form that is easily translated into an extended vector. The second problem—the weighting of the subvectors themselves—remains open to investigation. Another issue of some interest here is the weighting of terms within the subvectors—objective vs. subjective. (We have produced some useful results in relation to the term weighting issue; our work on the weighting of subvectors is promising but not well developed. In any case, subvector weighting is unlikely to have a measurable effect within the large INEX window.)

The extended vector capability of Smart appeared to us well suited for XML with respect to the retrieval of documents. But there is no facility for retrieving at the element level (or at various levels of granularity), which is a requirement of INEX tasks. We are interested in determining the feasibility of incorporating the functionality (i.e., flexibility and granularity) required for XML retrieval within the extended vector environment. (Our first experiment in this vein, based on the methods of Grabs and Schek [7, 8], was not successful. However, more work is necessary before conclusions can be drawn.)

1.2 System Description
Our system handles the processing of XML text as follows:

(1) The documents are parsed using a simple XML parser available on the web. Each of our 18 c-types is now identifiable in terms of its XML path.

(2) The documents and queries are translated into Smart format and indexed by Smart as extended vectors. (For the results reported in this paper, we used only the *article*-based indexing.)

(3) Retrieval takes place by running the queries against the indexed collection. The result is a list of *articles* ordered by decreasing similarity to the query. (A variety of term weighting schemes are available through Smart.)

(4) For each query, results are sorted by correlation and the top 100 elements are converted to INEX format and reported.

The retrieval itself is straight-forward. The only variation is the splitting of certain CAS queries into separate portions which are then run in parallel to ensure that the elements retrieved meet the specified criteria. See [3] for an example of this type.

2. EXPERIMENTS

In the following sections, we describe the experiments performed with respect to the processing of the CO and CAS topics, respectively. In all cases, we use only the topic title and keywords as search words in query construction.

2.1 Using CO Topics

Our first task is to formulate the CO topic in extended vector form. Of the 18 c-types composing the extended vector, 8 contain subjective identifiers (i.e., *abs*, *kwd*, *bdy*, *atl*, *edintro*, *ack*, *bibl_atl*, and *bibl_ti*). The extended vector topic is formed by associating the search words of the topic with each of these 8 c-types. The remaining c-types contain objective identifiers and are not used in formulating CO queries. Our more interesting experiments are discussed briefly below; see [1] for more information. The subvectors were equally weighted in all these cases.

2.1.1 2002 Topics

All the results based on 2002 topics were produced through the original inex_eval.

Tuned *Lnu-ltu* Term Weighting: In this experiment, we tuned the collection as indicated by Singhal, *et. al*., in [13]. Results under generalized quantization were 0.065 whereas strict quantization produced 0.095.

Augmented *tf-idf* (*atc*) Term Weighting: 2002 topics under generalized quantization produced an average precision of 0.033.

Retrieval at the Element Level: In this experiment, we used indexings of the collection at the paragraph and section levels in addition to the article level. Untuned (estimated) *Lnu-ltu* weights were used in each case. For each query, the rank-ordered lists were sorted and the top 100 elements reported. Average precision was 0.042 under generalized quantization.

Flexible Retrieval: In this experiment, we used the method of Grabs and Schek [7, 8]. Identifying the paragraph as the basic indexing node, we used the paragraph indexing of the collection and reported the best elements by calculating statistics for other elements on the fly. Average precision was 0.018 under generalized quantization.

2.1.2 2003 Topics

Our 2003 CO submission was based on parameters that produced the best results for 2002 CO topics, i.e., *Lnu-ltu* term weighting with equal subvector weights. The recall-precision graphs for 2003 CO topics under the revised inex_eval are given below in Figures 1 and 2. The results under inex_eval_ng (overlap ignored) are shown in Figures 3 and 4.

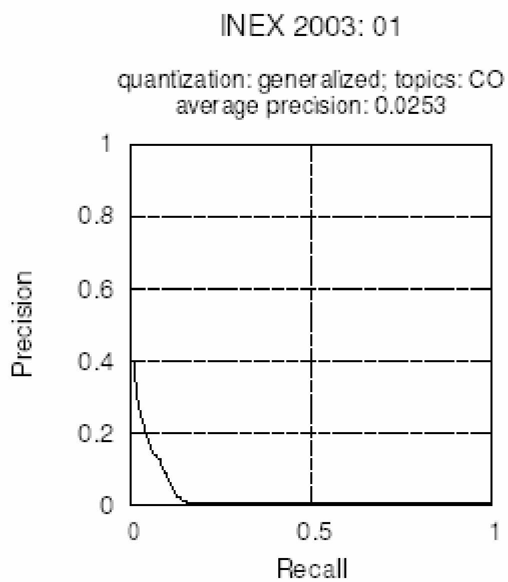

Figure 1. Recall-precision for CO, Generalized

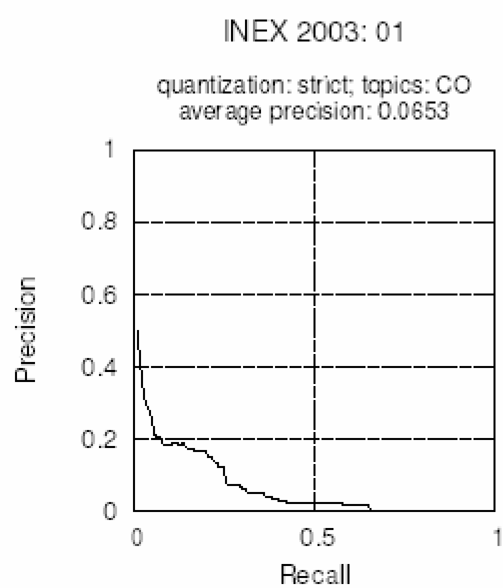

Figure 2. Recall-precision for CO, Strict

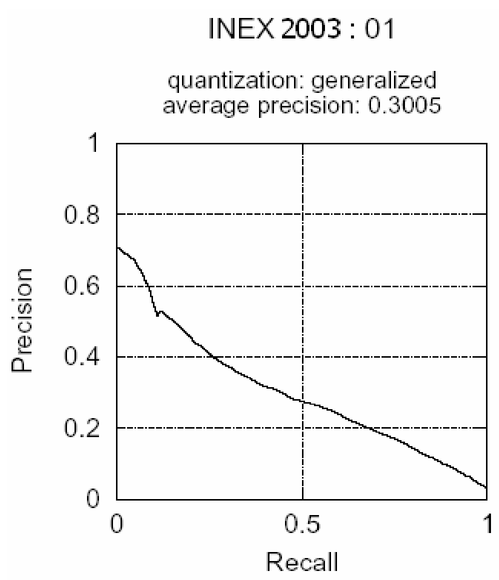

Figure 3: Recall-precision for CO, Gen under ng

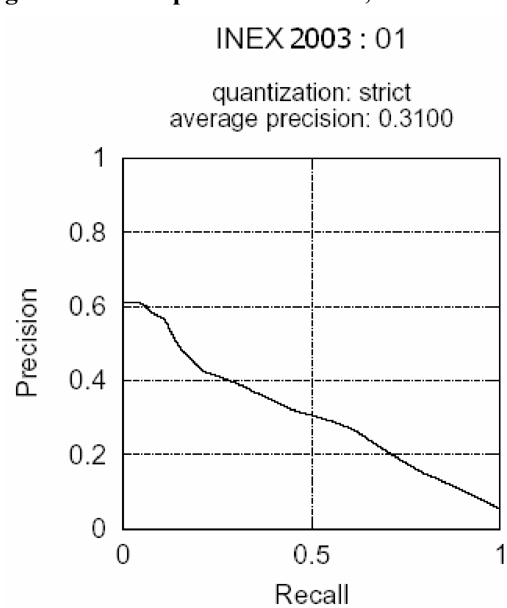

Figure 4: Recall-precision for CO, Strict under ng

2.2 Using CAS Topics

Although we were able to formalize the formulation of extended vector CO topics very easily, CAS topics are more of a challenge. At present, we are still working on this task. Thus the 2003 topic formulations were performed manually. Equal subvector weighting was applied in each case. Experiments performed during the past year using the INEX 2002 queries are described briefly below; see [2] for more information. Evaluation was performed through the original inex_eval.

2.2.1 2002 Topics

Untuned *Lnu_ltu* Term Weighting: All subvectors are weighted in this fashion. Not unexpectedly, average precision was low—0.179 under generalized and 0.222 under strict quantization.

Lnu_ltu (for subjective subvectors) and *nnn* (for objective subvectors) Term Weighting: Here we used simple term frequency weights (*nnn*) for the objective subvectors combined with *Lnu_ltu* weights for the subjective subvectors. Average precision was 0.187 under generalized and 0.235 under strict quantization.

Augmented tf-idf (*atc*) Term Weighting: All subvectors were weighted with *atc* weights. Average precision was 0.194 and 0.238 under generalized and strict quantization, respectively.

Augmented tf-idf (*atc*—for subjective subvectors) and *nnn* (for objective subvectors) Term Weighting: These weights returned an average precision of 0.192 under generalized and 0.243 under strict quantization.

2.2.2 2003 Topics

Our 2003 submission used *atc* term weighting for all subvectors with equal subvector weights. Due to the exigencies of the academic schedule, we were able to submit only under VCAS. We will report these results when they become available through INEX.

2.3 Results

During the past year, we produced a working system. Our results for CO topics are on the whole very good, ranking first or second in 3 of the 4 evaluations (e.g., in Figures 3 and 4) under inex-eval-ng. Yet although we were able to produce decent results for the 2002 CO topics under the original inex_eval, our results for the 2003 CO topics under the revised inex_eval fall far from the top. We are still assessing the causes. The assessment of our CAS submission awaits evaluation from INEX. An overview of results to date may be seen in Table 1.

3. CONCLUSIONS

Our system is still in an early stage of development. The issue of term weighting has now become clearer; the weighting of the subvectors themselves is still an open question. The next challenge is to develop a method of returning results at the element level, i.e., to retrieve at the desired level of granularity. Our plans include further investigation of the methods of others along with the development of an approach that may be better suited to our own environment.

Table 1. Comparison of Best Case Avg Precision for CO Topics

	UMD		INEX	
	gen	strict	gen	strict
'02 Topics	0.0650	0.0950	0.0700	0.0880
'03 Topics: inex_eval	0.0253	0.0653	0.0968	0.1140
'03 Topics: inex_eval_ng*	0.2971	0.2961	0.3051	0.2961
'03 Topics: inex_eval_ng**	0.2789	0.2873	0.3408	0.2966

* overlap ignored; ** overlap considered

4. REFERENCES

[1] Apte, S. Adapting the extended vector space model for content-oriented XML retrieval. Master's Thesis, Dept. of Computer Science, University of Minnesota Duluth (2003).

[2] Bapat, H. Adapting the extended vector space model for structured XML retrieval. Master's Thesis, Dept. of Computer Science, University of Minnesota Duluth (2003).

[3] Crouch, C., Apte, S., and Bapat, H. Using the extended vector model for XML retrieval. In Proc of the First Workshop of the Initiative for the Evaluation of XML Retrieval (INEX), (Schloss Dagstuhl, 2002), 99-104.

[4] Crouch, C., Crouch, D. and Nareddy, K. The automatic generation of extended queries. In Proc. of the 13th Annual International ACM SIGIR Conference, (Brussels, 1990), 369-383.

[5] 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).

[6] Fox, E., Nunn, G. and Lee, W. Coefficients for combining concept classes in a collection. In Proc. of the 11th Annual International ACM SIGIR Conference, (Grenoble, 1988), 291-307.

[7] Grabs, T. and Schek, H. Generating vector spaces on-the-fly for flexible XML retrieval. In Proc of the ACM SIGIR Workshop on XML and Information Retrieval, (Tampere, Finland, 2002), 4-13.

[8] Grabs, T. and Schek, H. ETH Zurich at INEX: Flexible information retrieval from XML with PowerDB-XML. INEX 2002 Workshop Proceedings, (Dortland, 2002), 35-40.

[9] Salton, G. Automatic information organization and retrieval. Addison-Wesley, Reading PA (1968).

[10] Salton, G., Wong, A., and Yang, C. S. A vector space model for automatic indexing. Comm. ACM 18, 11 (1975), 613-620.

[11] Salton, G. and Buckley, C. Term weighting approaches in automatic text retrieval. In IP&M 24, 5 (1988), 513-523.

[12] Singhal, A. AT&T at TREC-6. In The Sixth Text REtrieval Conf (TREC-6), NIST SP 500-240 (1998), 215-225.

[13] Singhal, A., Buckley, C., and Mitra, M. Pivoted document length normalization. In Proc. Of the 19th Annual International ACM SIGIR Conference, (Zurich,1996), 21-19.