Measuring Similarity of Semi-structured Documents with Context Weights

Christopher C. Yang, Nan Liu
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
Ph: (852) 2609 8239
Email: {yang,nliu}@se.cuhk.edu.hk

ABSTRACT
In this work, we study similarity measures for text-centric XML documents based on an extended vector space model, which considers both document content and structure. Experimental results based on a benchmark showed superior performance of the proposed measure over the baseline which ignores structural knowledge of XML documents.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval models – *retrieval models*.

General Terms
Models, Experimentation

Keywords
XML Search and Retrieval

1. INTRODUCTION
In traditional IR, both documents and queries are expressed in free text which provides no clue about the logical structure of the documents. In contrast, a XML document has a hierarchical structure, where the different tags of nodes indicate different semantic properties of the text underneath such as it being the "title" of an article. Its hierarchical nature also reveals the semantic relationships among different element types. The richer representation of XML allows a user to formulate their information need more precisely by imposing constraints on both the content and structure. In recent years, there has been an increasing interest on XML search and retrieval in the IR community such as the INEX evaluation initiative [3]. However, the focus of most existing research is on finding the most relevant component in an XML document to return as answer to a short query. The problem of document to document comparison is not well studied yet, while it is important for many applications such as similarity search, clustering and classification.

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.

2. VECTOR SPACE MODEL FOR XML
The vector space is a standard model for text retrieval, in which queries and documents are both represented by vectors in a high dimensional space whose axes correspond to the set of terms in the document corpus. The coordinate of each document vector is determined by $w_d(t)$ which stands for the weight of t in document d and is commonly computed by a score of the *tf *idf* family. The relevance of a document to a query is usually evaluated by the similarity of their corresponding vectors such as the cosine measure:

$$\cos(x, y) = \frac{|x \cap y|}{|x| \times |y|} = \frac{\sum_t w_x(t) * w_y(t)}{\sqrt{\sum_t w_x^2(t)} * \sqrt{\sum_t w_y^2(t)}} \quad (1)$$

Several recent works [1,3,4] proposed extensions of the regular vector space model for XML documents so that both content and structure are taken into account. The basic idea underlying these approaches is to use pairs of the form *(t, c)* instead of single terms *t* as the dimensions of the vector space, where a term *t* is further differentiated by the context *c* of its appearance which is typically identified by the path leading to the term from the root of the hierarchical structure of the XML document. Thus the weight of individual terms $w_x(t)$ should be replaced by weight of terms in context denoted by $w_x(c,t)$. Moreover, when computing vector similarity, it is proposed that terms occurring in different but similar contexts should also be accounted for by utilizing a context resemblance measure:

$$cr(c_x, c_y) = \begin{cases} \dfrac{1 + \min(|c_x|, |c_y|)}{1 + \max(|c_x|, |c_y|)} & \text{if } c_x \text{ or } c_y \text{ is prefix of the other} \\ 0 & \text{otherwise} \end{cases}$$

Thus the cosine similarity between two XML documents become

$$\cos(x, y) = \frac{\sum_{(t,c_x) \in x} \sum_{(t,c_y) \in y} w_x(c_x, t) * w_y(c_y, t) * cr(c_x, c_y)}{\sqrt{\sum_t w_x^2(c,t)} * \sqrt{\sum_t w_y^2(c,t)}}$$

To instantiate this similarity measure, we need to specify the computation of the weight $w_x(c,t)$. In [1], $w_x(c,t)$ is computed as $tf_x(c,t) \times idf(c,t)$ where

- $idf(c,t) = \log(|N|/|N_{(c,t)}|)$ with $|N|$ = total number of documents in the collection and $|N_{(c,t)}|$ = number of documents containing *(c,t)*.
- $tf_x(c,t)$ is the number of occurrences of *(t,c)* in *x*.

To further exploit the structural information, [4] introduced the concept of "weighted term frequency" to reflect that different

locations carries different importance when comparing two documents. For example, in an article, a word occurring in the "title" is more important than in a "paragraph", which in turn is more important than in the "reference". In [2], the importance of different tags are manually assigned a weights $\sigma(x_i)$ and the weight $w(c)$ for a particular context c with a tree path $x_0x_1...x_n$ is estimated by $\prod_{i=0}^{n}\sigma(x_i)$. And $w_x(c,t)$ is replaced by $w(c)\times tf_x(c,t)\times idf(c,t)$.

Instead of manually assigning the importance weights to different element tags and computing the importance for a context by the product of the weights of the elements on the path, we propose to apply genetic algorithm to search for the optimal context weights $w(c)$'s based on a set of training documents for which the relevant documents have been manually selected. Genetic algorithms(GAs) provide a optimization procedure motivated by biological evolution. Given the current population of hypotheses, GAs generates successor hypotheses by mutating and recombining the best currently known hypotheses so that the members of the population will approach the optimal hypothesis in the long run. Given the collection of XML documents, we may find the set of all possible contexts in which a term may occur, denoted by C. Each hypothesis thus represents a set of $|C|$ numbers in the range [0, 1], each of which corresponds to the importance weight of a particular context and is encoded by a 5-bit binary string. So a hypothesis is encoded by a $5\times|C|$ bit binary string. To evaluate a hypothesis, we instantiate the similarity measure with the set of context weights and use it to rank the collection for the set of training documents and compute the mean average precision (MAP) as the *fitness* of the hypothesis. Mutation is simulated by randomly inverting some bits in the representation and crossover is implemented by swap bits randomly sampled from two input hypotheses. The detailed algorithm is presented in Table 1.

P ← Generate p hypothesis randomly

For i = 1 to MAX_ITER do

(1) Create new generation P'

(2) For each h in P, compute $Fitness(h)$

(3) Rank h's by Fitness(h)

(4) Compute a probability distribution Pr(h) such that the probability for a hypothesis h_i is proportional to its rank produced in (3)

(5) Probabilistically select (1 – r)p members of P to add to P_s following Pr(h)

(6) Probabilistically select rp/2 pairs of hypotheses from P following Pr(h)

(7) Choose m percent of the members of P_s uniformly. For each, invert k randomly selected bits in its binary string representation.

(8) P ← P'

Return the hypothesis from P that has the highest fitness

Table 1: Algorithm for optimizing context weights where the parameters p, r, m, k controls the population size, crossover rate, mutation rate and number of bits to invert by mutation.

3. Evaluation:

The dataset used for evaluation consists of 1894 XML documents describing items in the Museum of Qin Terracotta Warriors and Horses with a total size of 7.8 MB. Each document contains the following types of elements: title, description, record type, resource type, object condition, location, repository, reference and source. We chose 40 documents for evaluation and manually selected their relevant documents from the collection. To study the effectiveness of the structure based similarity measure for XML documents, we compare it with a baseline, which ignores the structure and compare two documents using the regular vector space with each term weighted by $tf_x(t)\times idf(t)$. For the structure based similarity measure, we compare two schemes for computing $w_x(c,t)$ both based on $w(c)\times tf_x(c,t)\times idf(c,t)$. The first scheme, denoted by structure I, uses the same weight 1.0 for any context c. The second scheme, denoted by structure II, uses the $w(c)$'s tuned by GAs. For structure II, we evaluate its performance through 8-fold cross validation, where in each run, 32 documents are used for training and the remaining 8 for testing and the final performance is computed by averaging the performance on the testing data in the 8 different runs.

The experimental results are summarized in Figure 1, based on which we can make the following interesting observations:

- Utilizing structure has shown clearly superior performance over the baseline. Both structure I and structure II are above the baseline at nearly all recall points.
- The context weight is useful for improving the structure based similarity measure as shown by the 0.09 improvement in average performance.

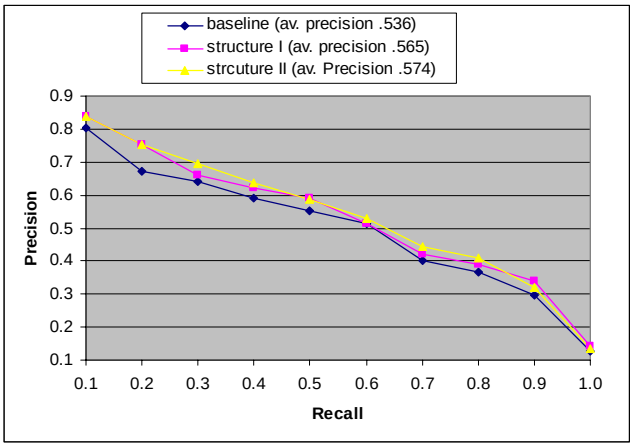

Figure 1: Performance of different similarity

4. REFERENCES

[1] D. Carmel, Y.S. Maarek, M. Mandelbrod, Y. Mass and A. Soffer. "Searching XML Documents via XML Fragments", In Proceedings of SIGIR' 2003, Toronto, Canada, 2003

[2] V. Kakade and P. Raghavan. "Encoding XML in Vector Spaces", In Proceedings of ECIR'2005, Santiago, Spain

[3] Initiative for the evaluation of XML retrieval http://qmir.dcs.qmul.ac.hk/INEX/

[4] S. Liu, Q. Zhu and W.W. Chu. "Configurable Indexing and Ranking for XML Information Retrieval", In Proceedings of SIGIR' 2003, Toronto, Canada