2007

Effectiveness of Rich Document Representation in XML Retrieval

F. Raja
University of Tehran, Iran

M. Keikha
University of Tehran, Iran

M. Rahgozar
University of Tehran, Iran

Farhad Oroumchian
University of Wollongong in Dubai, farhado@uow.edu.au

Effectiveness of Rich Document Representation in XML Retrieval

Fahimeh Raja[1], Mostafa Keikha[1], Maseud Rahgozar[1], & Farhad Oroumchian[1,2]
[1]Database Research Group, Control and Intelligent Processing center of Excellence, faculty of ECE, School of Engineering, University of Tehran, Tehran, Iran
{f.raja, m.keikha}@ece.ut.ac.ir
rahgozar@ut.ac.ir
[2]The College of IT, University of Wollongong in Dubai
FarhadO@uow.edu.au

Abstract

Information Retrieval (IR) systems are built with different goals in mind. Some IR systems target high precision that is to have more relevant documents on the first page of their results. Other systems may target high recall that is finding as many references as possible. In this paper we present a method of document representation called RDR to build XML retrieval engines with high specificity; that is finding more relevant documents that are mostly about the query topic. The Rich Document Representation (RDR) is a method of representing the content of a document with logical terms and statements. The conjecture is that since RDR is a better representation of the document content it will produce higher precision. In our implementation, we used the Vector Space model to compute the similarity between the XML elements and queries. Our experiments are conducted on INEX 2004 test collection. The results indicate that the use of richer features such as logical terms or statements for XML retrieval tends to produce more focused retrieval. Therefore it is a suitable document representation when users need only a few more specific references and are more interested in precision than recall.

Introduction

Extensible Markup Language (XML) is becoming the most popular format for information representation and data exchange. The widespread use of XML has brought up a number of challenges for Information Retrieval (IR) systems. These systems exploit the logical structure of documents instead of a whole document. In traditional IR, a document is considered as an atomic unit and is returned to a user as a query result. XML assumes a tree-like structure for the documents for example sentences, paragraphs, sections, etc. Therefore XML retrieval is not only concerned with finding relevant documents but with finding the most appropriate unit in the document that satisfies a user's information need. A meaningful retrievable unit should not be too small because in this case it might not cover all the aspects of users need. It should not be too large either because in this case there could be a lot of non-relevant information that are of no particular interest to a user's current information need. Therefore, XML retrieval is an approach for providing more focused information than traditionally offered by search engines when we know the structure of the documents (Fuhr *et al.*, 2002; Oroumchian *et al.*, 2004).

The most popular document representation in IR is called single term where stemmed single words are used as a representation of document (Salton *et al.*, 1993). A more sophisticated representation is based on single terms and phrases. These phrases could be formed statistically or linguistically. The usefulness of using phrases and their contribution largely depends on the type of the system and weighting scheme used (Fox, 1981). Adding phrases to a single term representation in vector space system with a good weighting such as Lnu.ltu (Greengrass, 2000) will only add 3-5% (Singhal *et al.*, 1996) to precision. However, in a system with weaker weighing the contribution of the phrases could be as much as 10% (Singhal *et al.*, 1996).

In this paper we present an NLP/logical approach for representing XML text in XML information retrieval. In this approach, first text is processed and special relationships are extracted and then these relations are converted into logical forms similar in syntax to the multi-valued logic of Michalski (Collins & Michalski, 1989). Then XML elements are represented by their single words, phrases, logical terms and logical statements. This form of document representation is called RDR (Rich Document Representation) and explained below.

This representation is the main document representation of a system called PLIR (PLausible Information Retrieval). PLIR assumes that retrieval is an inference as pointed out by Prof. Van Rijsbergen (Van Rijsbergen, 1988). However, unlike Van Rijsbergen work which has only a single inference, PLIR uses many inferences of the theory of Human Plausible Reasoning and offers calculations for the certainty of the inferences. For example if a document is indexed as related to 'Mac OS X' for 'iMac' and a user is interested in operating systems for personal computers. PLIR is able to reason that since 'Mac OS X' is a kind of operating system and 'iMac' is a kind of personal computer then the user could be interested in this document, although it does not share a single word with the query. Of course the confidence on this conclusion depends on how dominant is 'Mac OS X' among all other things called 'operating system' and how typical is 'iMac' as a personal computer. PLIR has outperformed a typical vector space model (Oroumchian & Oddy, 1996). It seemed interesting to examine whether the power of PLIR comes from its reasoning or from its rich document representation. In this line of thought, the document representation (RDR) was separated from reasoning and was tested on vector space model in different collections and settings. One such experiment was conducted on clustering documents in the second stage of a two stage retrieval system with the OHSUMED collection. In those tests it was shown that RDR is better representation, than single words and phrases with respect to different clustering methods and evaluation criteria in the second stage of a two stage retrieval system (Oroumchian & Jalali, 2004). However, in other retrieval experiments on OHSUMED, when different vector space systems were built by choosing single words, single words+phrases, and RDR, the result was inconclusive. Single words+phrases and RDR were better than single words but they were not that different from each other. By examining the results, it was realized that RDR is pulling a lot of new unseen and unjudged documents. When for a few queries those documents were judged, for those queries RDR showed better performance than single words+phrases. However, since judging all those unjudged documents was costly and time consuming and the experimenters were not expert in the medical field, the experiments on OHSUMED collection was abandoned.

The rest of this paper is organized as follows: next section will introduce RDR as our method of representing documents. Third section gives a brief description of INEX 2004 test collection. In fourth section, we will explain our experiments and weighting scheme used in this study. Fifth section depicts our results and last section is the conclusion of the article.

Rich Document Representation

Rich Document Representation (RDR) is a method of representing documents by logical forms with the syntax of multi-valued logic. These logical forms could be any of the following:

1. Concepts (single stemmed words and phrases)
2. Logical terms: logical terms are in the form of A(B) where A is the descriptor and B is the argument. Logical forms are similar to predicates in Predicate logic. A logical term can have a descriptor and one or more arguments.

3. Logical statements: logical statements are in the form of A(B)= {C} Where A is the descriptor and B is the argument and C is the referent. For example Flower (England) = {Daffodil} which basically means Daffodils are flowers of England.

Multi-valued logic allows logical statements to have multiple referents or refer to an infinite set. However in the context of IR, there is no infinite document or sentence therefore it is not possible for logical statements to reference an infinite set. Also, in order to make matching concepts easier, logical statements have been restricted to have only a single referent. Statements that could translate to Logical statements with multiple referents are written as multiple logical statements with the same descriptor and argument and a single referent.

PLIR uses RDR as its main document representation. PLIR treats all the logical statements representing all the documents as a single knowledge base that describes a possible world which includes some documents. In such a space, statements of different documents could provide complementary information for each other. For example, a document could state that DB2 is a relational database system from IBM and another document could indicate SQL as a data manipulation language for relational databases. Therefore, system can infer that SQL is also a data manipulation language for DB2. PLIR uses the rich set of inferences available in the theory of Human Plausible reasoning to guess the relevancy of documents to queries. PLIR's strength comes from representing document content as logical statements, the accuracy of ISA relations that it extracts from the documents and uses as an ontology, the power of its inferences and its local weighting system called dominance (Oroumchian & Oddy, 1996). In a normal vector space model, there is no reasoning and ISA (kind of) relations are not as useful, so the only use of RDR would be providing a deeper representation for the text. In 2004, PLIR has been tested on INEX 2004 collection and the results after fixing a few problems in first runs were average (Karimzadegan *et al.*, 2005). This research explores RDR representation's application in retrieving more specific XML elements without any reasoning. Our hypothesis is that since RDR is a better representation of the content then it should lead to retrieval of more specific elements.

RDR representation is extracted automatically from the text. The process of producing these logical forms is as follows:

1. Tag the text: The text has to be tagged by Part of Speech tags.
2. Rule based or Clue based extraction: in this process the output of the POS tagger is scanned for clues. These clues signify the existence of the relations in the text. For example a proposition such as 'of' or 'in', signifies a relationship between two noun phrases that it connects.

Table 1 shows a few sentence fragments and their equivalent in logical forms.

Sentence fragment	Representation in multi-valued logic and PLIR	Type
Resonances for glutamine	Resonance(glutamine)	Logical Term
Syngeneic tumor of BALB	Syngeneic _tumor(BALB)	Logical Term
Linux as an operating system for PCs	Operting_system(PC) ={Linux}	Logical Statement
Kish, which is an island in Persian Gulf	Island (Persian_Gulf) ={Kish} ISA (island, Kish)	Logical Statement

Several different tools have been developed and used for processing text and extracting relationships so far, some of them are written in Perl, some others in Java. The most general tool so far is a general purpose NLP package written in Java. We are in the process of completing it in order to cover all aspects of relation extraction and generating representation. After that we can make it available for public use.

INEX 2004 test collection

We have used INEX 2004 test collection for evaluation of our XML retrieval system. The INEX document collection is made up of the full-texts, marked up in XML that consists of 12,107 articles of the IEEE Computer Society's publications from 12 magazines and 6 transactions, covering the period of 1995-2002. Its size is about 494 megabytes. The collection contains scientific articles of varying length. On average an article contains 1,532 XML nodes, where the average depth of a node is 6.9. Overall, the collection contains over eight millions XML elements of varying granularity, each representing a potential answer to a user's query (Fuhr *et al.*, 2002).

The test collection contains two types of topics:
• Content-only (CO): these queries are standard information retrieval (IR) queries. In this type of queries, users are unaware of the structure of the documents. In this task, it is left to the retrieval system to decide the best retrievable unit in response to the user query.
• Content and structure (CAS): these queries contain conditions referring both to the content and structure of the requested answer elements.

There are 40 Co and 40 CAS queries in INEX 2004. For more information about CO and CAS queries, one can refer to (Sigurbjörnsson *et al.*, 2004). The focus of this paper is on 34 CO topics of this collection which have relevance assessments.

Figure 1 depicts a sample query in the collection. In this figure, "-" and "+" signs are used to show the importance of terms in the query.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE inex_topic (View Source for full doctype...)>
- <inex_topic topic_id="166" query_type="CO" ct_no="20">
    <title>+"tree edit distance" + XML - image</title>
    <description>We are looking for documents presenting approaches for evaluating the structural similarity between two labeled
      trees. We are mainly interested in approaches that could be applied to XML documents. We are not interested in articles
      dealing with tree structure representation of images and approaches related to the evaluation of the edit distance between
      two strings</description>
    <narrative>The possibility to evaluate the structural similarity between two XML documents is very attractive for clustering
      together documents presenting similar structures. A lot of work has been done in the area of evaluation of similarity
      between two labeled trees by means of tree edit distance. Some approaches specifically tailored for XML documents or
      semi-structure data have been recently developed. We are interested in articles dealing with the problem of measuring the
      structural similarity between two labeled trees. We are not interested in articles dealing with tree structure representation
      of images and approaches related to the evaluation of the edit distance between two strings.</narrative>
    <keywords>tree edit distance XML</keywords>
  </inex_topic>
```

Figure 1: A sample INEX query

Experiments

In our implemented system, first we indexed all the elements of a document by a vector of its single terms, phrases and logical terms and statements. In this system, a query consisting of a sentence fragment can be treated as a regular text. It can be scanned for extracting its logical

terms. For example, in the query “an algorithm for index compression”: “index compression” will be detected as a phrase and “algorithm(index_compression)” will be identified as a logical term. The retrieval process starts with first scanning the query and extracting single terms, phrases and logical terms and then finding all the references in the collection for the followings:

1. All the single words such as “algorithm” in the query.
2. All the phrases such as “index_compression” in the query.
3. All the logical terms such as “algorithm(index_compression)” that are in query.

This is a case of direct retrieval where the document is indexed by the term. This action is similar to regular keyword matching by search engines except that search engine do not index logical terms.

Any kind of weighting can be used for weighting these logical terms and statements. The best way of weighting them is to treat them as phrases and weight them accordingly. However, PLIR uses a different scheme which is called Dominance but we do not make use of dominance in this experiment. In this work we use *"tf.idf"* in our weighting model. We apply the following formula for weighting each index term (single term, phrase, logical term and statement) of queries:

$$w(t,q) = \alpha * termFreq(t,q) * idf(t) * nf(q) \quad (1)$$

Where:

α = 2/3 , for terms with "_" before them in the query (2)
=4/3 , for terms with "+" before them in the query
=1 , otherwise

$$idf(t) = \ln(\frac{N}{n(t)}) \quad (3)$$

$$nf(q) = \frac{1}{lenq} \quad (4)$$

termFreq(t,q): frequency of occurrence of term t within the query q
idf(t): inverse document frequency of term t
N: number of documents in the collection
n(t): number of documents in the collection that contain term t
nf(q): normalization factor of query
lenq: query length which is equal to number of terms in the query

It should be mentioned that the parameter *"nf"* is computed separately for single terms, phrases, and logical terms and statements; i.e. *"lenq"* is calculated three times as follows:

- number of single terms in the query
- number of phrases in the query
- number of logical terms and statements in the query

For weighting index terms of document elements we use:

$$w(t,e) = tf(t,e) * idf(t) * nf(e) * childEffec\ t(e) \quad (5)$$

Where:

$$tf(t,e) = \frac{(1+\log(termFreq(t,e))}{(1+\log(avg(termFreq(e))} \quad (6)$$

$$idf(t) = \ln(\frac{N}{n(t)}) \quad (7)$$

$$nf(e) = \frac{1}{\sqrt{len(e)}} \quad (8)$$

$$childEffect(t,e) = \frac{\#\text{of sublements of e with term t}}{\#\text{of sublements of e}} \quad (9)$$

termFreq(t,e): frequency of occurrence of term t within the element e
avg(termFreq(e)): average term frequency in element e
idf(t): inverse document frequency of term t
N: number of documents in the collection
n(t): number of documents in the collection containing term t
nf(e): normalization factor of element e
len(e): element length which is equal to number of terms in the element
childEffect(t,e): effect of occurrence of term t within subelements of element e in the weight of element e

As the case for queries, the parameter *"nf"* for document elements is calculated separately for single terms, phrases and logical terms and statements.

After weighting index terms of queries and document elements, we made three separate vectors from the weights of single terms, phrases, logical terms and statements for each query and element in the given collection. For example, the corresponding vectors for the query "a_b(c)" are:

$V_{Single\_Terms} = (w(a,q), w(b,q), w(c,q))$
$V_{Phrases} = (w(a\_b,q))$
$V_{Logical\_Terms} = (w(a\_b(c),q))$

Once vectors have been computed for the query and for each element, using a weighting scheme like those described above, the next step is to compute a numeric "similarity" between the query and each element. The elements can then be ranked according to how similar they are to the query.

The usual similarity measure employed in document vector space is the "inner product" between the query vector and a given document vector (Greengrass, 2000). We use this measure to compute the similarity between the query vectors (for single terms, phrases, logical terms and statements) and an element vectors. Finally, we simply add these three relevance values to get the total relevance value for each element and rank the elements based on this relevance value.

Results

Relevance in INEX is defined according to the following two dimensions (Kazai *et al.*, 2004):
• Exhaustivity (E): the extent to which the document component discusses the topic of request.
• Specificity (S): the extent to which the document component focuses on the topic of request.

Both dimensions of an XML element are measured in 4-point scale with degrees highly (3), fairly (2), marginally (1) and not (0) exhaustive/specific. Hence each assessed element is assigned one relevance degree combined by its exhaustivity and specificity as $(e,s) \in ES$ where

$$ES = \{(0,0),(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3)\} \qquad (10)$$

To apply different metrics, two relevance dimensions are mapped to a single relevance scale by a quantization function $f_{quant}(e,s) : ES \rightarrow [0,1]$. Figure 2 depicts four different quantization functions that we use to evaluate our system.

Strict Quantisation:

$$f_{strict}(e,s) := \begin{cases} 1 & \text{if}(e,s) = (3,3) \\ 0 & \text{otherwise} \end{cases}$$

Generalised Quantisation:

$$f_{gen}(e,s) := \begin{cases} 1 & \text{if}(e,s) = (3,3) \\ 0.75 & \text{if}(e,s) \in \{(2,3),\ (3,2),\ (3,1)\} \\ 0.5 & \text{if}(e,s) \in \{(1,3),\ (2,2),\ (2,1)\} \\ 0.25 & \text{if}(e,s) \in \{(1,2),\ (1,1)\} \\ 0 & \text{if}(e,s) = (0,0) \end{cases}$$

Specificity Oriented Generalised:

$$f_{sog}(e,s) := \begin{cases} 1 & \text{if}(e,s) = (3,3) \\ 0.9 & \text{if}(e,s) = (2,3) \\ 0.75 & \text{if}(e,s) \in \{(1,3),\ (3,2)\} \\ 0.5 & \text{if}(e,s) = (2,2) \\ 0.25 & \text{if}(e,s) \in \{(1,2),\ (3,1)\} \\ 0.1 & \text{if}(e,s) \in \{(2,1),\ (1,1)\} \\ 0 & \text{if}(e,s) = (0,0) \end{cases}$$

Highly Specific:

$$f_{s3\_e321}(e,s) := \begin{cases} 1 & \text{if } e \in \{3,\ 2,\ 1\} \wedge s = 3 \\ 0 & \text{otherwise} \end{cases}$$

Figure 2: Quantization functions

We have different runs on the "title" and "description" parts of the queries. We rank the results of these runs with "thorough" and "focused" tasks. Thorough task returns elements ranked in relevance order where specificity is rewarded and overlap is permitted. Unlike thorough task, in focused task overlap is not permitted (Clarke *et al.,* 2006).

In figures 3 and 4, the red lines depict the precision-recall graphs of our results in comparison with the results of other systems participating in INEX 2004. We used INEX 2004 evaluation software (EvalJ) to evaluate our results.

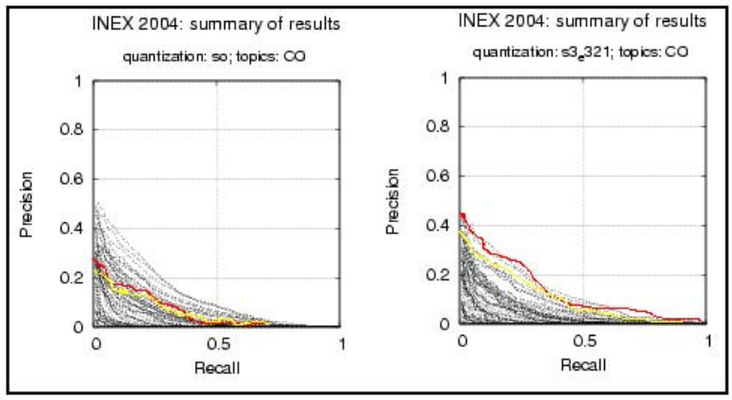

Figure 3: our results in comparison with others in INEX 2004 with SO quantization in the left and s3e321 quantization in the right.

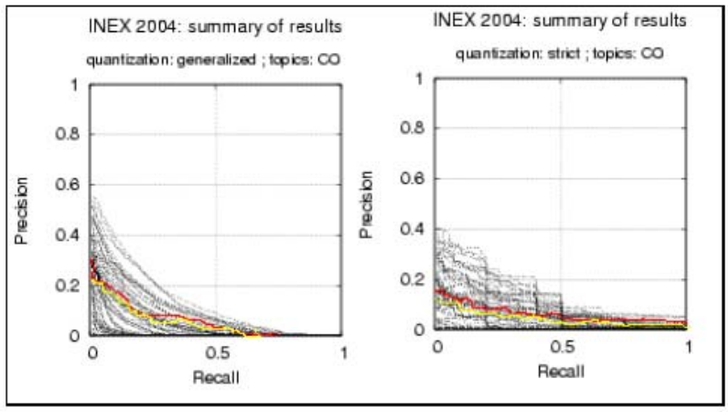

Figure 4: our results in comparison with others in INEX 2004 with generalized quantization in the left and strict quantization in the right.

As can be seen in figure 3, our system is one of the best systems with respect to s3e321 quantization (the graph in the upper right corner). The s3e321 quantization targets specificity and RDR produces more specific representation of XML content. This is consistent with other tests and reported results by others. In all previous reported experiments, PLIR and RDR have produced high precision but very low recall.

Table 2 shows the precision of our system with different quantization. RDR yields in higher precision with the quantization in which the specificity is more important.

Quantization	SO	s3e321	Generalized	strict
Avg. Precision	0.067813461	0.130817927	0.05816055	0.039179341

Table 2: Average precision of the results with different quantizations

The average precision for s3e321 is at least twice better than the other quantizations. Our goal was never to build a general purpose system that would do well in all aspects. We are mostly

interested in building high precision systems that return very specific and focused material on top ranks (top 20) of their results. In such as scenario, this representation seems to be a right choice at the cost of having a larger document representation. The number of logical terms and statements is normally around half of the number of phrases.

Conclusion and future works

In this paper we presented an NLP/Logical approach, called RDR (Rich Document Representation), which uses a rich set of features to represent XML elements. These features are single terms, phrases, logical terms and logical statements. Logical terms and statements are extracted from text by using linguistic clues. The simplest clue is a proposition.

We conducted our experiments using INEX 2004 test collection and satisfactory results seems to suggest the RDR representation could provide a better representation for elements. The results show that RDR improves specificity of the elements returned to the user which is one of the goals of XML information retrieval. In INEX2004 collection with S2e123 measure, this method almost outperformed any other system.

In future, we are going to conduct more tests on different domains and collections and to improve on document representation by using automatic relevance feedback. We also need to focus on improving our NLP methods for extracting the logical forms. We are hoping, to improve our information extraction methods and produce better and more reliable logical statements which will result in even higher precision.

Another direction could be using relevance feedback to make it possible for the system to adapt itself to the user judgments.

References

Clarke, C., Kamps, J. & Lalmas, M. (2006). INEX 2006 Retrieval Task and Result Submission Specification.

Collins, A. & Michalski, R. (1989). The Logic of Plausible reasoning: A Core Theory. *Cognitive Science*, 1--49.

Fox, E.A. (1981). Lexical relations: enhancing effectiveness of information retrieval systems. *SIGIR Newsletter*, 15(3), 5--36.

Fuhr, N., Gövert, N., Kazai, G. & Lalmas, M. (2002). INEX: INitiative for the Evaluation of XML retrieval. In Ricardo Baeza-Yates, Norbert Fuhr, and Yoelle S. Maarek (Eds.), *Proceedings of the SIGIR 2002 Workshop on XML and Information Retrieval.*

Greengrass, E. (2000). Information Retrieval: A Survey. *DOD Technical Report TR-R52-008-001.*

Karimzadegan, M., Habibi, J., & Oroumchian, F. (2005). XML Document Retrieval by means of Plausible Inferences. In N. Fuhr, M. Lalmas, S. Malik and Z. Szlávik (Eds.), *Advances in XML Information Retrieval, Third Workshop of the INitiative for the Evaluation of XML Retrieval INEX 2004.*

Kazai, G., Lalmas, M. & Piwowarski, B. (2004). INEX 2004 Relevance Assessment Guide.

Oroumchian, F. & Jalali, A. (2004). Rich document representation for document clustering. *RIAO 2004 Conference Proceedings: Coupling approaches, coupling media and coupling languages for information retrieval, Le Centre de Hautes Etudes Internationalies d'Informatique Documenataire - C.I.D.* (pp. 1--9).

Oroumchian, F. & Oddy, R.N. (1996). An application of plausible reasoning to information retrieval. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp.18--22).

Oroumchian, F., Karimzadegan, M. & Habibi, J. (2004). XML Information Retrieval by Means of Plausible Inferences. *5th International Conference on Recent Advances in Soft Computing, RASC* , (pp. 542--547).

Salton, G., Allan, J. & Buckley, C. (1993). Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 49--58).

Sigurbjörnsson, B., Larsen, B., Lamas, M. & Malik, S. (2004). INEX 2004 Guidelines for Topic Developemnt.

Singhal, A., Buckley, C. & Mitra, M. (1996). Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 21--29).

Van Rijsbergen, C. J. (1988). Logics for Information Retrieval. *AL4T 88.*