# Strict and Vague Interpretation of XML-Retrieval Queries

Andrew Trotman
Department of Computer Science
University of Otago
Dunedin, New Zealand

andrew@cs.otago.ac.nz

Mounia Lalmas
Department of Computer Science
Queen Mary University of London
London, UK

mounia@dcs.qmul.ac.uk

## ABSTRACT

Structural hints in XML-retrieval queries can be used to specify both the granularity of the search result (the target element) and where in a document to search (support elements). These hints might be interpreted either strictly or vaguely, but does it matter if an XML search engine interprets these in one way and the user in another? The performance of all runs submitted to INEX 2005 content and structure (CAS) tasks were measured for each of four different interpretations of CAS. Runs that perform well for one interpretation of target elements do so regardless of the interpretation of support elements; but how to interpret the target element does matter. This suggests that to perform well on all CAS queries it is necessary to know how the target structure specification should be interpreted. We extend the NEXI query language to include this, and hypothesize that using this will increase the overall performance of search engines.

## Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *query formulation, Search process*.

## General Terms

Measurement, Performance, Experimentation.

## Keywords

Element retrieval, XML retrieval, INEX.

## 1. INTRODUCTION

In an element retrieval system it is possible to search not only whole documents, but also document elements. Considerable resources have been spent designing and implementing query languages for exactly this purpose. Languages like XPath [2] have been designed to enable searchers to specify exactly not only which documents they are looking for but also which elements within those documents.

Investigations into user behavior on web search engines has shown that the average number of terms per query is 2.2 [1]. It is hard to recon this result with the complex query languages seen for XML. Work done as part of the INEX initiative has shown that even expert searchers find it hard to correctly specify queries in such languages [3]. INEX consequently uses none of these languages but has adopted its own called NEXI [4], which has been shown to be effective for both expert and novice users [5,6].

Given the NEXI query *//article[about(., IR)]//sec[about(., XML)]*, there are two structural constraints in effect. The first specifies the granularity of the result (the target element), in this case *//article//sec*. The second specifies where to look (support elements), in this case there are two, *//article* when looking for information about *IR* and *//article//sec* when looking for information about *XML*. Collectively these constraints are referred to as structural hints and the queries have content and structure (CAS) considerations.

When specifying these structural hints users may have a clear picture of what they mean. When searching for *smith* in an *author* structure they do not mean *smith* in the *profession* structure; however, when searching for *sections* of articl*es* about *Golomb compression* they are likely to be happy with a *subsection* of an article on the topic. The interpretation of the structure constraint can be strict or vague (loose).

In this investigation, conducted as part of INEX 2005, we ask the question: does it matter if we build an XML search engine that uses one interpretation of the query when the user has another? We find that it does and suggest extensions to the NEXI query language to allow users to specify this in their queries.

## 2. Methods

We define two interpretations of a structural hint: vague and strict. In the strict interpretation an element is relevant if and only if the path of that element exactly matches the structural hint and the content of the element matches the information need. In the vague interpretation any element whose content satisfies the information need is considered relevant regardless of the path.

These interpretations can be applied to both the target element and the support elements. This gives four possible interpretations of a query referred to as VVCAS, VSCAS, SVCAS and SSCAS in which xyCAS refers to a vague (V) or strict (S) interpretation of the target element (x) or the support elements (y).

In this investigation we are looking for indications of the different interpretations using the queries and judgments of assessors in an IR evaluation forum.

As part of INEX, topics were solicited from participants. Where a topic contained more than one clause, separate topics for each were also solicited. For example for *//article[about(., IR)]//sec [about(., XML)]*, it is both clauses: *//article[about(., IR)]* and *//article//sec[about(., XML)]*; which are referred to as child top-

ics. Participants were asked to submit runs[1] for each interpretation against all topics (including child topics)[2].

Assessments were made against the narrative of the topic, the natural language description of the information need. These assessments are those that satisfy the vaguest interpretation of the query, the VVCAS assessment pool. The SVCAS (strict target) pool is a subset of these judgments; only those strictly matching the target element constraint – computed by removing all relevant elements that did not match the target.

For VSCAS and SSCAS, a relevant element must come from a document containing elements that strictly conform to all given child topics. For most topics these documents were computed as the set intersection of those relevant to all that topic's children[3]. This document list is then used to filter the VVCAS pool.

In total 10 topics have relevant elements in the pools for VVCAS and SVCAS (topics 253, 256, 257, 260, 261, 264, 265, 270, 275, and 284). Of these, three topics (253, 261, and 265) have no elements conforming to a strict interpretation of the support element and so were not used in VSCAS and SSCAS evaluation. Performance was measured using MAep with generalized quantization as is standard in XML retrieval.

## 3. Results

The performance of each run was computed for each interpretation. Presented in Figure 1 is the performance of these runs for both VVCAS and the VSCAS. Regardless of the task to which a run was submitted, those that perform well at VVCAS also perform well at VSCAS. The two interpretations are similar. A similar result can be seen for SVCAS and SSCAS, but not for the others (see, for example, Figure 2).

Table 1 presents the Pearson product moment correlation coefficients for the performance of each run against each pair of interpretations – that is, how well the performance on one task correlates to the performance on another (irrespective of submitted task). This suggests two separate interpretations of CAS, that in which the target element is interpreted strictly and that in which it is interpreted vaguely. The interpretation of the support elements does not appear to be important.

## 4. Conclusions

When users supply structural hints in their query they are expecting it to be interpreted in a particular way. We have shown that there is (presently) no one "answer fits all" interpretation of the structural hints, and that different runs (ranking algorithms) perform well for different interpretations. It follows that to perform well at all CAS queries an XML search engine must know how to interpret the target element structural hints.

We now extend NEXI by adding an optional *strict* operator ($) to the end of a path specification. In this way the path *//article//sec* is a vague structural constraint, but *//article//sec$* is a constraint requiring strict conformance. By taking this new operator into consideration an XML search engine can choose the most appropriate ranking algorithm for the query, which we hypothesize will increase result in a search engine precision increase.
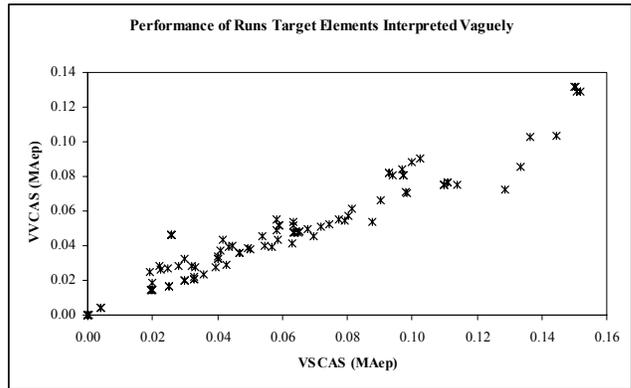
---

[1] Against version 1.8 of the INEX IEEE document collection.

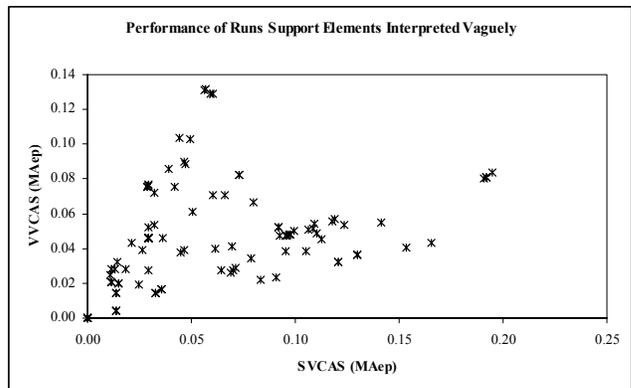[2] Only parent topics are used in evaluation.

[3] The exception is topic 250 which requires a set union.

**Table 1: Pearson's correlation shows two distinct tasks.**

|  | SSCAS | SVCAS | VSCAS | VVCAS |
|---|---|---|---|---|
| SSCAS | 1.00 | 0.89 | 0.39 | 0.37 |
| SVCAS | 0.89 | 1.00 | 0.33 | 0.36 |
| VSCAS | 0.39 | 0.33 | 1.00 | 0.96 |
| VVCAS | 0.37 | 0.36 | 0.96 | 1.00 |



**Figure 1: Vague target elements correlate.**



**Figure 2: Vague support elements do not correlate**

## 5. Acknowledgements

## 6. References

[1] Beitzel, S. M., Jensen, E. C., Chowdhury, A., Grossman, D., & Frieder, O. (2004). Hourly analysis of a very large topically categorized web query log. In *Proc 27th SIGIR*, (pp. 321-328).

[2] Clark, J., & DeRose, S. (1999). XML path language (XPath) 1.0, W3C recommendation. The WWW Consortium.

[3] O'Keefe, R. A., & Trotman, A. (2003). The simplest query language that could possibly work. In *Proc 2nd INEX*.

[4] Trotman, A., & Sigurbjörnsson, B. (2004). Narrowed Extended XPath I (NEXI). In *Proc INEX 2004*, (pp. 16-40).

[5] Trotman, A., & Sigurbjörnsson, B. (2004). NEXI, now and next. In *Proc INEX 2004*, (pp. 41-53).

[6] van Zwol, R., Baas, J., van Oostendorp, H., & Wiering, F. (2005). Query formulation for XML retrieval with bricks. In *Proc INEX-MW 2005, 2nd Ed*, (pp. 80-88).