# Vague Element Selection and Query Rewriting for XML Retrieval

Vojkan Mihajlović   Djoerd Hiemstra   Henk Ernst Blok
CTIT, University of Twente
P.O. Box 217, 7500AE Enschede, The Netherlands
{v.mihajlovic, d.hiemstra, h.e.blok}@utwente.nl

## ABSTRACT

In this paper we present the extension of our prototype three-level database system (TIJAH) developed for structured information retrieval. The extension is aimed at modeling vague search on XML elements. All three levels (conceptual, logical, and physical) of the TIJAH system are enhanced to support vague search concepts. The vague search is implemented as vague selection of XML elements using XML element name expansion lists and rewriting techniques. We test the performance of retrieval models using automatically generated expansion lists and compared them with models that use manual ones. The goal is to find the best approach for structured information retrieval with vague structural constraints on element names expressed in the query.

## 1. INTRODUCTION

Due to the fact that more and more documents on the web that come in the structured format, such as XML, information retrieval community begins to realize the importance of document component retrieval and structured querying. To exploit structured retrieval, information retrieval like query languages are enriched with the ability to state structural constraints in the query (e.g., [1, 4, 5, 19]). The question is how these constraints should be treated when answering a query: as strict constraints that must be satisfied or just as users suggestions on where to search for information.

Similarly, as the user gives only a number of terms as hints for searching within a document, XML elements specified within the query need not be considered as a strict requirement but as a hint for structural search. Therefore, when formulating a query the user can state that the search (support) element or answer (target) element should be treated as a hint or as a constraint in the retrieval process. We use the term *search element* for elements in which the user would like to find some information and the *target element* for elements that the user would like to see as an answer to his query.

The motivation for vague search is found in the discussion raised during the 2004 INitiative for Evaluation of XML retrieval (INEX) workshop and the tasks that are set for the year 2005[1]. INEX 2005 ad-hoc track introduced a number of subtasks for both content-and-structure (CAS) and content-only (CO). For the CAS task, the goal was to test whether the structural constraints should be followed strictly or not, and if not, to what degree they should be freely interpreted.

[1]http://inex.is.informatik.uni-duisburg.de:2005/.

The idea is that the structural constraints should be considered as hints, and by each scenario different degree of vagueness for structural constraints should be tested. Therefore, the CAS task is divided into four scenarios:

- SSCAS that assumes strict matching between the structural conditions stated in the query and path leading to the search and answer elements.
- SVCAS where the structural conditions of search (support) elements need not to be strictly satisfied.
- VSCAS in which the answer (target) elements can be interpreted vaguely, i.e., the answer element need not to be the one specified in the query.
- VVCAS where all the structural constraints can be interpreted vaguely.

On the other hand, for the CO task we address the subtask termed as content-only plus structure (CO+S or COS), formed by adding structural constraints to a set of terms in CO query. Here, structural constraints are also interpreted vaguely, but without explicit separation among different sub-tasks. These queries are used to check whether the structural information can help in the searching process and in what way.

To support these different types of vague search scenarios expressed in user queries

- we introduced vague element search as a concept
- we allow terms to cross the structural boundaries stated in the query

The vague element search (selection) can be treated similarly as a query expansion on terms in traditional IR. For example, if a user searches for the term ‘conclusion’, he might also be satisfied with terms ‘decision’, ‘determination’, ‘termination’, or ‘ending’ in the answer. In structured documents, if a user asks for ‘car’ elements, he would probably not mind getting ‘auto’ or ‘vehicle’ elements as an answer. Furthermore, he might also agree with the answers: ‘van’, ‘sports-car’, or ‘convertible’.

While the list of possible synonyms, hypernyms, and hyponyms for terms can be considered as relatively static over time (e.g., WordNet [15]) and the degree of similarity can be pre-specified, in the case of *element name expansion* the problem is more complex and dynamic. Besides the terms that have the same or similar meaning, like the ones given above, it can happen that element names follow different naming pattern. Thus, elements might have complex element names such as: ‘sports_car’, ‘vehicles_list’, etc.. Abbreviations could also be used, such as for section elements in INEX IEEE collection [10]: 'sec', 'ss1', 'ss2', 'ss3'.

Additionally, if a user asks for elements denoting one concept it might not be wrong if the answer is an element from a similar concept. Plenty of such examples can be identified in INEX; e.g., if a user asks for sections as answer elements, like in the INEX 2005 CAS query 235

```
 //article[about(.//abs, "data mining")]
              //sec[about(., "frequent itemsets")]
```

he might be satisfied with paragraphs, abstracts, or even short articles (summaries) given as an answer. Furthermore, the list of element names can be larger in semantically richer and heterogeneous XML collection and it can evolve over time with the introduction of the new XML collections.

The problem of element name matching is studied in the research area of schema matching and numerous techniques exist that try to resolve this problem (see [3, 18] for survey)[2]. However, we decided to simplify the vague element name search task and use the results from INEX 2004 assessments to find the expanded element names (see the following section for more details).

Throughout the paper we discuss the application of our TIJAH system for vague search in XML documents. Vague search is modeled using the concept of vague XML element selection and rewriting techniques. The TIJAH system [12, 13, 14] is developed as a transparent XML-IR three-level database system for structured information retrieval, consisting of conceptual, logical, and physical levels. The original TIJAH system can handle queries with the strict selection of XML elements, specified in the NEXI query language [19] and can reason about textual information. In this paper we extend the TIJAH system toward handling vague specification of XML elements in the query (similar to [5]).

Additionally, we employ two rewriting techniques at conceptual level that are used for extending the search on terms not only to the search elements deeper in the XML tree, but also to the higher-level elements that are used in the query formulation. We define two techniques for rewriting the original query as described in Section 2.

In this paper we aim to test whether we can automatically derive expansion lists that can be used to improve the vague search and to compare them them with rewriting techniques. For that we need a test collection with a real set of vague queries. Although queries specified in INEX ad-hoc CAS and COS tasks are declared as vague, it is not clear whether every structural condition is vague in them. Furthermore, due to the content of the collection, i.e., homogeneous collection of scientific articles, the XML markup is mostly used to represent document structure (article, title, abstract, sections paragraphs, etc.). Therefore, the vagueness that can be expressed in INEX queries is mostly 'structural' vagueness (similarity between different structural elements of a document), and not the 'semantic' one (similarity between different concepts in different documents), which would be better suitable for our experimental setup. Even though INEX 2005 CAS and COS queries are not ideal for what we are aiming at and due to that we lack other suitable collections, we decided to make the fresh test our approach on INEX test collection.

[2]Note that the schema matching approaches are concerned with matching the exact relations among elements besides element name matching but due to the complexity of the problem we start our research by trying to understand the elementary problems such as element name matching.

The following section explains the extensions introduced in the TIJAH system to model vague XML element specification and rewriting techniques. Experimental setup is presented in Section 3. We conclude with the discussion of experiments performed using INEX 2004 and 2005 collections in Section 4 and with conclusions and future directions in Section 5.

## 2. VAGUENESS IN USER QUERIES

This section details the motivation and the implementation of vague search in our three-level database framework. We explain the extensions at each level, conceptual, logical, and physical, aimed for vague search on elements and for rewriting the queries.

### 2.1 Vague search in NEXI

Instead of extending our conceptual parser for rewriting content-and-structure (CAS) and content-only plus structure (COS) queries into variants with strict or vague specification of target and support elements and as the vague element specification should be expressed by the (expert) user we derive our own vague queries. Our vague queries are denoted with SS, VS, SV, VV in front of the CAS and COS query types, e.g., SVCAS, VSCAS, and VVCAS (SSCAS is equal to CAS in our case). Besides the vague selection of elements we model vagueness using query rewriting techniques.

#### *2.1.1 Vague element selection*

To express vague element selection we decided to extend the NEXI grammar with one extra symbol '$\sim$'. The 'tilde' symbol is used in front of the element name in the query specification, denoting that the element name does not have to be strictly matched in the query evaluation. As we decided to simplify the vague element name search task, instead of more advanced schema matching techniques used to find the expanded element names, we use the results from INEX 2004 assessments. The list of expanded element names is defined based on the list of element names assessed as relevant in INEX 2004 assessments process. The lists that we use, termed *element name expansion lists* are the following:

- One manually specifies set of lists with the default score 0.55 (based on 2004 experiments) denoted as *manual*55, given in Table 1[3]. The list is formed by selecting the most frequent highly and marginaly exhaustive elements and adding the most frequently used INEX query elements, such as *sec* (section), *p* (paragraph), *abs* (abstract), to the expansion lists of each element where they are not present and for which they seem to be a reasonable expansion element name.

- Seven automatically generated lists out of assessments with exhaustivity ($E$) and specificity ($S$) greater or equal to marginally (1), fairly (2), or highly (3) exhaustive or specific: *hh* ($E > 2, S > 2$), *hf* ($E > 2, S > 1$), *fh* ($E > 1, S > 2$), *ff* ($E > 1, S > 1$), *fm* ($E > 1, S > 0$), *mf* ($E > 0, S > 1$), *mm* ($E > 0, S > 0$).

[3]Other elements in INEX IEEE collection are not included in Table 1 as they were not present as target elements in the 2004 topic set.

**Table 1: Manual element name expansion list based on INEX 2004 assessments.**

| El. name | Expanded element names |
|---|---|
| abs | abs, fm, kwd, vt, p, sec, article, bdy, ref |
| article | article, bdy, sec, abs, fm, bm, bib, bibl, bb, p, ref |
| atl | atl, st, fgc |
| bb | bb, bm, bibl, bib, atl, art |
| bdy | bdy, article, sec, abs, p, ref |
| bib | bib, bm, bb, atl, art |
| fig | fig, sec, st, p, fgc, st, atl |
| fm | fm, sec, abs, kwd, vt, p, article, bdy, ref |
| kwd | kwd, abs, fm, st, fgc, atl |
| p | p, vt, abs, sec, fm, article, bdy, st |
| sec | sec, abs, fm, vt, p, article, bdy, bm, app |
| st | st, atl, fgc |
| tig | tig, bb |
| vt | vt, p, sec, bm, fig |

The default score is based on a number of relevant elements of that specific name, normalized by a total number of relevant elements[4], for all distinct target elements. The lists containd from xx to xx original element names (with the mean of xx), and each list contained on average xx expanded element names.

### 2.1.2 *Query rewriting techniques*

We also model vague node selection using two query rewriting techniques that we used previous years for INEX [12, 14]. These rewriting techniques treat structural constraints as strict but add a new *about* clause that searches for the same terms in as in the original query but in different elements. The rewriting is done at conceptual level.

In the first rewriting approach (rw I), all terms that are in different *about* clauses in the same predicate expression, and are not at the top level (i.e., not in `about(., term)` expression), are added to an extra top-level *about* clause in the same predicate expression.

The second approach (rw II), is an extension of the first one, where not only the terms from non top-level *abouts* are added to the new *about*, but also all the terms from the other predicate[5], if there exists any, are added to the top-level *about* in each predicate.

## 2.2 Introducing complex selection operator for vague node selection

The logical level of the TIJAH system is based on Score Region Algebra - SRA (see [13] for more details). The data model consists of a set of regions, each defined by region start ($s$) and region end ($e$) positions, region type ($t$), region name ($n$), and region score ($p$). The basic operators on regions are given in Table 2, where $r_1 \prec r_2 \equiv r_1.s > r_2.s \wedge r_1.e < r_2.e$. The first four define the selection of regions based on: region name and type - $\sigma_{n=name,t=type}(R)$, numeric value assigned to a region - $\sigma_{\diamond num}(R_1)$, and containment relation among regions - $R_1 \sqsupset R_2$ and $R_1 \sqsubset R_2$. Operator $R_1 \sqsupset R_2$ selects regions from $R_1$ that contain regions from $R_2$, and $R_1 \sqsubset R_2$ selects regions from $R_1$ that are contained in the regions from $R_2$.

The $\sqsupset_p$ operator is used for computing scores based on the containment relation among two regions ($R_1$ and $R_2$) and retrieval models specified using function $f_{\sqsupset}(r_1, R_2)$ (see Section 3). The two operators, $\blacktriangleright$ and $\blacktriangleleft$, specify score propagation to the containing or contained regions respectively. Operators $\sqcap_p$ and $\sqcup_p$ specify score combination in an AND and OR like combination of regions at the logical level.

The vague node selection at the conceptual level (NEXI) is translated into complex vague node selection operator at the logical level. However, the vague node selection operator in score region algebra has more expressive power than the simple NEXI extension at the conceptual level. It allows much finer specification of search and answer elements than a simple vague '∼' node name specification. The vague node selection operator in SRA is defined as a union of all XML element regions that match the names of the 'expanded name regions' within the element name expansion list. By default all 'expanded regions' are down-weighted by a predefined factor. The definition of the operator is given in the last row of Table 2.

In the definition of $\sigma^{expansion(class)}_{n=name,t=type}(R_1)$, $expansion(class)$ is a set that contains the expansions for all the region names in one expansion class, where expansion list for each region $name$ is denoted as $expansion(class, name)$ (with cardinality $n$):

$$expansion(class, name) := \{(ex\_n_1, ex\_w_1), (ex\_n_2, ex\_w_2), \\ ..., (ex\_n_n, ex\_w_n)\}$$

Here $ex\_n_i$ is a expanded element name and $ex\_w_i$ is a real number in the range $[0, 1]$ denoting the down-weight factor. The operator $\sigma^{expansion(class)}_{n=name,t=type}(R_1)$ assigns name ($ex\_n$) and score ($ex\_w$) values to the region name ($n$) and score ($p$) based on the name and score values in the expansion list $expansion(class, name)$.

The simple selection operator in basic score region algebra operator set $\sigma_{n=name,t=type}(R)$ can be considered as a complex selection operator where the $expansion(class, name)$ set contains only the $name$ element with $ex\_w = 1.0$. Note that the complex selection operator can also be expressed using the basic SRA selection operator and scaling operator ($R \circledast w$ denotes that the score of regions in the region set $R$ should be multiplied by $w$, i.e., $p := p \cdot w$) as follows:

$$\sigma^{expansion(class)}_{n=name,t=type}(R) := \\ (\sigma_{n=ex\_n_1,t=type}(R) \circledast ex\_w_1) \sqcup_p (\sigma_{n=ex\_n_2,t=type}(R) \circledast ex\_w_2) \\ \sqcup_p \; ... \; \sqcup_p (\sigma_{n=ex\_n_n,t=type}(R) \circledast ex\_w_n) \qquad (1)$$

As our NEXI extension does not allow explicit specification of the 'expanded regions' list we pre-defined the 'expanded regions' set and pre-specified the default value for *weight*. For such a purpose we used manually predefined lists in Table 1 and seven automatically generated lists from INEX 2004 assessments and combine them with the INEX equivalence classes given in Table 3 [10]. In this way we also kept the framework fairly simple.

The strict (SS) runs discussed in Section 4 use equivalence classes defined for INEX IEEE collection [10], depicted in Table 3 and termed *eq_class*, as these represent the default setup in INEX. For the vague selection we used the fusion of equivalence classes and our INEX 2004 expansion element name lists given in Table 1 and in seven automatically extracted lists. This is done in such way that every expanded element name in these lists that has the equivalent name in

[4]We had to manually edit the `vt` (vitae) expansion list using the *manual*55 run as the distribution of assessments for `vt` elements significantly degraded the performance of our 2004 runs.

[5]Note that the NEXI syntax allows only two predicates with the *about* clauses to be specified in one query [19].

Table 2: Score region algebra operators.

| Operator | Operator definition |
|---|---|
| $\sigma_{n=name,t=type}(R)$ | $\{r \mid r \in R \wedge r.n = name \wedge r.t = type\}$ |
| $\sigma_{\diamond num}(R_1)$ | $\{r_1 \mid r_1 \in R_1 \wedge \exists r_2 \in C \wedge r_2.t = \text{term} \wedge r_2 \prec r_1 \wedge r_2.n \diamond num\}$, where $\diamond \in \{=, <, >, \leq, \geq\}$ |
| $R_1 \sqsupset R_2$ | $\{r_1 \mid r_1 \in R_1 \wedge \exists r_2 \in R_2 \wedge r_2 \prec r_1\}$ |
| $R_1 \sqsubset R_2$ | $\{r_1 \mid r_1 \in R_1 \wedge \exists r_2 \in R_2 \wedge r_1 \prec r_2\}$ |
| $R_1 \sqsupset_p R_2$ | $\{(r_1.s, r_1.e, r_1.n, r_1.t, f_{\sqsupset}(r_1, R_2)) \mid r_1 \in R_1 \wedge r_1.t = \text{node}\}$ |
| $R_1 \blacktriangleright R_2$ | $\{(r_1.s, r_1.e, r_1.n, r_1.t, f_{\blacktriangleright}(r_1, R_2)) \mid r_1 \in R_1 \wedge r_1.t = \text{node}\}$ |
| $R_1 \blacktriangleleft R_2$ | $\{(r_1.s, r_1.e, r_1.n, r_1.t, f_{\blacktriangleleft}(r_1, R_2)) \mid r_1 \in R_1 \wedge r_1.t = \text{node}\}$ |
| $R_1 \sqcap_p R_2$ | $\{(r_1.s, r_1.e, r_1.n, r_1.t, p_1 \otimes p_2) \mid r_1 \in R_1 \wedge r_2 \in R_2 \wedge (r_1.s, r_1.e, r_1.n, r_1.t) = (r_2.s, r_2.e, r_2.n, r_2.t)\}$ |
| $R_1 \sqcup_p R_2$ | $\{(r.s, r.e, r.n, r.t, p_1 \oplus p_2) \mid r \in R_1 \vee r \in R_2\}$ |
| $\sigma^{expansion(class)}_{n=name,t=type}(R_1)$ | $\{(r_1.s, r_1.e, r_1.n, r_1.t, r.p) \mid r_1 \in R_1 \wedge r_1.t = type \wedge (r_1.n, r.p) \in expansion(class, name)\}$ |

Table 3: Equivalence classes for INEX IEEE collection.

| El. name | Equivalent names |
|---|---|
| `h` | `h, h1, h1a, h2, h2a, h3, h4` |
| `list` | `list, dl, l1, l2, l3, l4, l5, l6, l7, l8, la, lb, lc, ld, le, numeric-list, numeric-rbrace, bullet-list` |
| `p` | `p, ilrj, p1, p2, p3, ip1, ip2, ip3, ip4, ip5, item-none` |
| `sec` | `sec, ss1, ss2, ss3` |

the *eq_class name* part is also expanded with the *eq_class* equivalent names for *name*. These expansions are termed *manual*55 and *xx* for other seven lists, where $x \in \{h, f, m\}$, as explained in the previous section.

Therefore, the *eq_class* selection on section elements can be expressed as $\sigma^{expansion(eq\_class)}_{n=\text{'sec'},t=node}(R)$, and vague node selection ∼**sec**, using highly exhaustive and highly specific elements, can be transformed into the next SRA operation $\sigma^{expansion(hh)}_{n=\text{'sec'},t=node}(R)$. In such a way we can transparently define the set of expanded nodes and their respective weights and use them for vague node selection in a vague element name selection retrieval scenarios.

## 2.3 The implementation of vague selection operator

At the physical level, since we are working with the known INEX IEEE data collection, and as we used static INEX equivalence element name lists and expansion element name list based on INEX 2004 assessments, we decided to replicate the lists and store them as tables at the physical level, i.e., in MonetDB [2]. Thus, we have eight tables with *(entity_name, expansion_name, expansion_weight)* for *manual*55 and *xx* lists, and one *(entity_name, equivalent_name)*[6] for *eq_class* list. The complex selection operator is then implemented as an additional MIL (MonetDB Interpreter Language) function to functions implementing other operators [14], based on the definition given in the previous section, that uses data from these tables.

For example, the vague *name* selection operator on region table $R$ and the 'expansion regions' table $S$ for the *uni_class* element names, in the relational algebra can be defined as:

$$\pi_{r.s,r.e,r.n,r.t,s.weight}(\sigma_{s.n=name}(S) \bowtie_{s.n=r.n} (\sigma_{r.t=node}(R)))$$

## 3. EXPERIMENTAL SETUP

Below, after introducing the retrieval models instantiated in score region algebra operators and metrics reported in the paper, we illustrate our approaches for INEX CAS and COS (sub)tasks.

[6]In the experiments we do not store weights for equivalent region names as we assume that their default weight is 1.0.

### 3.1 Retrieval models

We base the instantiation of retrieval models on language models [6] since they showed good performance in our previous experimental runs [14]. For the relevance score computation on regions we use Equation 2, where *Root* is the root region of the collection and $size(r) := r.e - r.s - 1$.

$$f^{\text{LM}}_{\sqsupset}(r_1, R_2) = r_1.p(\lambda \frac{\sum_{r_2 \in R_2 | r_2 \prec r_1} r_2.p}{size(r_1)} + (1-\lambda)\frac{|R_2|}{size(Root)}) \quad (2)$$

For upwards score propagation and downwards score propagation we employ Equation 3 and Equation 4.

$$f_{\blacktriangleright}(r_1, R_2) = r_1.p \cdot \sum_{r_2 \in R_2 | r_1 \prec r_2} r_2.p \quad (3)$$

$$f_{\blacktriangleleft}(r_1, R_2) = r_1.p \cdot \sum_{r_2 \in R_2 | r_2 \prec r_1} r_2.p \quad (4)$$

Abstract operators $\otimes$ and $\oplus$ in score combination operators, $\sqcap_p$ and $\sqcup_p$, are implemented as product and sum.

### 3.2 Metrics

For the evaluation of our 2004 and 2005 runs we use some of the official INEX 2004 and 2005 metrics, and precision at selected recall points. For the 2004 runs we use *inex_eval*, with both strict and generalized quantization, and we report a set-based overlap (aka O-overlap out of four overlap types distinguished in [16]). The *inex_eval* is based on the concept termed expected search length [17], and it uses three levels of exhaustivity and specificity: marginal, fair, and high [11]. Additionally, following the idea that even the simple metrics can give enough evidence for the evaluation of XML retrieval [7, 16], we report precision at three recall points: 10, 25, and 50.

The official INEX metrics for 2005 ad-hoc track are based on extended Cumulative Gain (xCG) metrics [9]. The official metrics are: normalized xCG (nxCG), effort-precision/gain-recall (ep/gr), and extended Q and R [8]. We report the evaluation results of our 2005 runs using nxCG at recall points 10, 25, and 50, as it can be compared to the precision at the low recall points. nxCG actually measures the gain a user has accumulated up to the specific rank, compared to the gain he could have accumulated if the ranking was ideal. The evaluation can be done either with the generalized or with the strict quantization. We also report the ep/gr MAP. ep/gr measures the user effort in inspecting the retrieved elements with respect to his effort in case the ranking was ideal, which resembles the *inex_eval* measure. We use only VVCAS and COS.Thorough assessments as we wanted to test the approaches on the same assessments set and without going into discussion over the overlap issue.

## 3.3 Vague CAS and COS queries

Since we decided to extend the NEXI syntax with the vague selection we had to manually rewrite the queries for each CAS and COS scenario except the SSCAS and SSCOS. For example, the (SS)CAS query 225:

```
//article[about(.//fm//atl, "digital libraries")]
          //sec[about(.,"information retrieval")]
```

is rewritten into three variants:

- SVCAS:
```
//article[about(.//∼fm//∼atl, "digital libraries")]
       //sec[about(.,"information retrieval")]
```
- VSCAS:
```
//article[about(.//fm//atl, "digital libraries")]
       //∼sec[about(.,"information retrieval")]
```
- VVCAS:
```
//article[about(.//∼fm//∼atl, "digital libraries")]
       //∼sec[about(.,"information retrieval")]
```

We do not consider the 'article' element as a vague element in case it is not the target element or it is not the element in which the *about* search should be performed, as in these cases the 'article' element just serves as a focusing element for deeper search in the XML tree.

## 3.4 Query rewriting

As we explained in Section 2, we use two techniques for query rewriting. According to the first one we add the terms from the *about* clause that are not in the top-level element to the top level element. For example, for INEX 2005 topic 240:

```
//article[about(.//(abs|kwd), quality control measure)]
           //sec[about(.//p, software quality)]
```

the rewritten query using rw I is:

```
//article[about(.//(abs|kwd), quality control measure)
           and about(., quality control measure)]
           //sec[about(.//p, software quality) and
           about(., software quality)]
```

Similarly, for the second rewriting technique (rw II), as it is an extension of the rw I technique, we use the rw I rewriting rule and also interchange terms from the other *about* clauses. Thus, for the same topic we have:

```
//article[about(.//(abs|kwd), quality control measure)
           and about(., quality control measure) and
           about(., software quality)]
           //sec[about(.//p, software quality) and
           about(., software quality) and
           about(., quality control measure)]
```

As can be seen in the next section we run these queries in isolation or in combination with the vague element selection queries.

## 4. DISCUSSION

In this section we discuss the results of experimenting with the vague element selection and rewriting techniques on INEX 2004 and 2005 collections. We start with estimating the best expansion lists and continue with rw I and rw II experiments and their comparison and combination with the vague selection. The result values given in bold in Tables 4 to 12 represent the highest scores (precision or MAP) in the column.

## 4.1 Estimating the best element name expansion lists

In this set of experiments we test whether automatically generated runs are comparable with the manual run and which of them is the best. As can be seen in Tables 4 to 6, except for the generalized *inex_eval* MAP, all automatic runs are comparable and outperform the manual one in many cases, especially when looking at 2004 runs and precision at 25 and 50 for 2005 runs. Although we did not put much effort in specifying the expansion lists for our manual run, we think it is a good representative for what element names the expert user would accept in the results lists

Out of automatic runs, the one that uses all relevant elements in INEX 2004 assessments set (*mm*) seems to be the most effective. This is particularly the case for the early precision and overall MAP when using generalized quantization. Furthermore, this run shows constantly good results across different measures. This can be viewed as an indicator that the user really appreciates the wider set of element names in the expansion lists and that the only problem is how to estimate better their importance, i.e., downweighting factor. Therefore, we selected *mm* and *manual55* runs for our further experiments.

## 4.2 Comparing vague element selection and query rewriting

Here we test if we can improve the effectiveness when replacing strict queries with vague ones and when using rewriting techniques. Tables 7 to 9 show that both the rewriting techniques and the vague element search help. The improvements are significant and they can go up to more than 100% (e.g., for the "*manual55*, VV" run with generalized quantization presented in Table 9). Looking at the rewriting techniques, the rw II shows overall better scores, especially for the early precision as can be seen in Table 8, and it has higher MAP values.

Clearly, the vague element selection has higher MAP values than the rewriting techniques, but in all CAS experiments it has lower precision at low recall points. This can indicate that the rewriting techniques might be used as a precision tool, while the vague element selection can be considered as a recall tool. Looking at different vague scenarios, namely SV, VS, and VV, and except for some early precision scores (see Table 8), VV runs seem to have the best performance. Therefore, "*mm*, VV" and "*manual55*, VV" runs are used in combination with the rewriting techniques for further experiments.

## 4.3 Combining vague element selection and query rewriting

The third set of experiments confirms our assumption about the rewriting techniques as a precision and vague element search as a recall tool. As can be seen in Tables 10 to 12 in most of the cases the combination of the rw I and rw II rewriting techniques and manual and automatic vague element search improves early precision. However, not in all cases we managed to save the MAP values, especially for the rw I combinations as can be seen in Table 12.

## 5. CONCLUSIONS AND FUTURE WORK

Throughout the paper we show that the TIJAH database system is flexible enough to incorporate new advanced search techniques, such as vague element selection and query rewriting. We have shown that rewriting techniques and vague element selection are viable solutions for vague search in XML documents. While query rewriting techniques are more suitable for obtaining higher precision at low recall points, vague

**Table 4: INEX 2004 CAS experiments with different expansion classes evaluated using *inex_eval* and precision at different recall points.**

| | Strict | | | | Generalized | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Exp. class** | **Pr@10** | **Pr@25** | **Pr@50** | **MAP** | **Pr@10** | **Pr@25** | **Pr@50** | **MAP** | **Overlap** |
| ff | **0.0192** | 0.0169 | **0.0138** | 0.08133 | 0.0769 | 0.0615 | 0.0477 | 0.05787 | 44% |
| fh | **0.0192** | **0.0185** | **0.0138** | 0.08200 | 0.0846 | **0.0646** | **0.0485** | 0.05573 | 43% |
| fm | **0.0192** | **0.0185** | **0.0138** | 0.08270 | 0.0769 | 0.0631 | 0.0477 | 0.05772 | 45% |
| hf | **0.0192** | 0.0169 | **0.0138** | 0.08214 | **0.0885** | 0.0615 | **0.0485** | 0.05839 | 44% |
| hh | **0.0192** | **0.0185** | **0.0138** | 0.08064 | 0.0808 | 0.0631 | **0.0485** | 0.05563 | 43% |
| mf | **0.0192** | **0.0185** | **0.0138** | 0.08157 | 0.0846 | 0.0631 | 0.0469 | 0.05712 | 44% |
| mm | **0.0192** | **0.0185** | **0.0138** | **0.08330** | 0.0846 | 0.0615 | 0.0462 | 0.05798 | 45% |
| manual55 | 0.0154 | 0.0108 | 0.0100 | 0.08202 | 0.0692 | 0.0462 | 0.0362 | **0.06230** | 60% |

**Table 5: INEX 2005 CAS experiments with different expansion classes evaluated using nxCG at different recall points and ep/gr.**

| | Strict | | | | Generalized | | | |
|---|---|---|---|---|---|---|---|---|
| **Exp. class** | **nxCG[10]** | **nxCG[25]** | **nxCG[50]** | **MAP** | **nxCG[10]** | **nxCG[25]** | **nxCG[50]** | **MAP** |
| ff | 0.1333 | 0.1578 | 0.1511 | 0.01066 | 0.2711 | 0.2702 | 0.2534 | 0.06895 |
| fh | **0.1444** | 0.1578 | **0.1556** | 0.01081 | 0.2736 | 0.2736 | 0.2537 | 0.06666 |
| fm | 0.0778 | 0.1578 | 0.1511 | 0.01037 | 0.2723 | 0.2699 | **0.2543** | 0.07027 |
| hf | 0.0889 | **0.1622** | 0.1511 | 0.01067 | 0.2760 | **0.2742** | 0.2517 | 0.06870 |
| hh | **0.1444** | **0.1622** | **0.1556** | 0.01073 | 0.2767 | 0.2726 | 0.2520 | 0.06693 |
| mf | **0.1444** | 0.1578 | 0.1533 | **0.01094** | 0.2687 | 0.2685 | 0.2492 | 0.06824 |
| mm | **0.1444** | 0.1578 | **0.1556** | 0.01085 | **0.2811** | 0.2728 | 0.2529 | 0.07062 |
| manual55 | **0.1444** | 0.1444 | 0.1467 | 0.01056 | 0.2545 | 0.2553 | 0.2428 | **0.07296** |

**Table 6: INEX 2005 COS experiments with different expansion classes evaluated using nxCG at different recall points and ep/gr.**

| | Strict | | | | Generalized | | | |
|---|---|---|---|---|---|---|---|---|
| **Exp. class** | **nxCG[10]** | **nxCG[25]** | **nxCG[50]** | **MAP** | **nxCG[10]** | **nxCG[25]** | **nxCG[50]** | **MAP** |
| ff | 0.0765 | **0.0877** | 0.0865 | **0.00219** | 0.2822 | 0.2564 | 0.2261 | 0.05907 |
| fh | 0.0765 | 0.0783 | 0.0830 | 0.00218 | 0.2765 | 0.2492 | 0.2202 | 0.05956 |
| fm | 0.0765 | 0.0854 | **0.0877** | 0.00180 | 0.2855 | 0.2541 | 0.2213 | 0.05783 |
| hf | 0.0765 | 0.0854 | **0.0877** | 0.00191 | 0.2869 | 0.2522 | 0.2197 | 0.05750 |
| hh | 0.0765 | 0.0830 | 0.0842 | 0.00218 | 0.2881 | 0.2495 | 0.2230 | 0.05831 |
| mf | 0.0765 | 0.0759 | 0.0830 | 0.00218 | 0.2849 | 0.2479 | 0.2238 | 0.05998 |
| mm | 0.0765 | 0.0830 | 0.0854 | 0.00218 | **0.2912** | 0.2580 | 0.2258 | 0.06060 |
| manual55 | **0.0824** | 0.0759 | 0.0689 | 0.00218 | 0.2851 | **0.2585** | **0.2315** | **0.06872** |

**Table 7: INEX 2004 CAS experiments with different vague scenarios and rewriting techniques evaluated using *inex_eval* and precision at different recall points.**

| | Strict | | | | Generalized | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Exp. class** | **Pr@10** | **Pr@25** | **Pr@50** | **MAP** | **Pr@10** | **Pr@25** | **Pr@50** | **MAP** | **Overlap** |
| eq_class | 0.0192 | 0.0185 | 0.0138 | 0.07117 | 0.0692 | 0.0615 | 0.0462 | 0.03746 | 25% |
| rw I | **0.0269** | **0.0215** | **0.0162** | 0.07241 | **0.0846** | **0.0692** | 0.0469 | 0.03968 | 26% |
| rw II | **0.0269** | **0.0215** | **0.0162** | 0.07909 | **0.0846** | **0.0692** | 0.0469 | 0.04485 | 27% |
| mm, SV | 0.0192 | 0.0185 | 0.0138 | 0.07154 | 0.0808 | 0.0615 | **0.0492** | 0.03781 | 24% |
| manual55, SV | 0.0192 | 0.0185 | 0.0138 | 0.07131 | 0.0769 | 0.0615 | **0.0492** | 0.03716 | 24% |
| mm, VS | 0.0192 | 0.0185 | 0.0138 | 0.08078 | 0.0654 | 0.0554 | 0.0408 | 0.05730 | 45% |
| manual55, VS | 0.0077 | 0.0077 | 0.0085 | 0.07709 | 0.0385 | 0.0354 | 0.0292 | **0.06233** | 59% |
| mm, VV | 0.0192 | 0.0185 | 0.0138 | **0.08330** | **0.0846** | 0.0615 | 0.0462 | 0.05798 | 45% |
| manual55, VV | 0.0154 | 0.0108 | 0.0100 | 0.08202 | 0.0692 | 0.0462 | 0.0362 | 0.06230 | 60% |

**Table 8: INEX 2005 CAS experiments with different vague scenarios and rewriting techniques evaluated using nxCG at different recall points and ep/gr.**

| | Strict | | | | Generalized | | | |
|---|---|---|---|---|---|---|---|---|
| **Exp. class** | **nxCG[10]** | **nxCG[25]** | **nxCG[50]** | **MAP** | **nxCG[10]** | **nxCG[25]** | **nxCG[50]** | **MAP** |
| eq_class | 0.1000 | 0.1022 | 0.0867 | 0.00581 | 0.2799 | 0.2851 | 0.2644 | 0.05033 |
| rw I | 0.1000 | 0.1200 | 0.1022 | 0.00609 | 0.2687 | 0.2834 | 0.2645 | 0.04670 |
| rw II | **0.1889** | 0.1289 | 0.1022 | 0.00777 | 0.3030 | **0.2977** | **0.2679** | 0.05476 |
| mm, SV | 0.0889 | 0.1067 | 0.0911 | 0.00609 | 0.2865 | 0.2882 | 0.2626 | 0.05219 |
| manual55, SV | 0.1000 | 0.1022 | 0.0844 | 0.00609 | **0.3066** | 0.2853 | 0.2419 | 0.05291 |
| mm, VS | 0.1333 | 0.1533 | 0.1511 | 0.01012 | 0.2672 | 0.2658 | 0.2524 | 0.06749 |
| manual55, VS | 0.1444 | 0.1222 | 0.1400 | 0.00975 | 0.2316 | 0.2417 | 0.2391 | 0.06720 |
| mm, VV | 0.1444 | **0.1578** | **0.1556** | **0.01085** | 0.2811 | 0.2728 | 0.2529 | 0.07062 |
| manual55, VV | 0.1444 | 0.1444 | 0.1467 | 0.01056 | 0.2545 | 0.2553 | 0.2428 | **0.07296** |

**Table 9: INEX 2005 COS experiments with different vague scenarios and rewriting techniques evaluated using nxCG at different recall points and ep/gr.**

| | Strict | | | | Generalized | | | |
|---|---|---|---|---|---|---|---|---|
| **Exp. class** | **nxCG[10]** | **nxCG[25]** | **nxCG[50]** | **MAP** | **nxCG[10]** | **nxCG[25]** | **nxCG[50]** | **MAP** |
| eq_class | 0.0471 | 0.0559 | 0.0559 | 0.00153 | 0.2677 | 0.2258 | 0.1787 | 0.03205 |
| rw I | 0.0588 | 0.0748 | 0.0595 | 0.00161 | 0.2715 | 0.2430 | 0.1894 | 0.03323 |
| rw II | 0.0588 | 0.0677 | 0.0571 | 0.00158 | 0.2872 | 0.2467 | 0.1898 | 0.03409 |
| mm, SV | 0.0471 | 0.0559 | 0.0559 | 0.00153 | 0.2772 | 0.2333 | 0.1951 | 0.03657 |
| manual55, SV | 0.0471 | 0.0559 | 0.0559 | 0.00152 | 0.2727 | 0.2349 | 0.1972 | 0.03650 |
| mm, VS | 0.0765 | **0.0854** | **0.0854** | 0.00213 | 0.2827 | 0.2499 | 0.2042 | 0.04283 |
| manual55, VS | **0.0824** | 0.0807 | 0.0689 | 0.00215 | 0.2751 | 0.2410 | 0.2060 | 0.04587 |
| mm, VV | 0.0765 | 0.0830 | **0.0854** | **0.00218** | **0.2912** | 0.2580 | 0.2258 | 0.06060 |
| manual55, VV | **0.0824** | 0.0759 | 0.0689 | **0.00218** | 0.2851 | **0.2585** | **0.2315** | **0.06872** |

**Table 10: INEX 2004 CAS experiments on combining vague search and rewriting techniques evaluated using *inex_eval* and precision at different recall points.**

| | Strict | | | | Generalized | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Exp. class** | **Pr@10** | **Pr@25** | **Pr@50** | **MAP** | **Pr@10** | **Pr@25** | **Pr@50** | **MAP** | **Overlap** |
| rw I | **0.0269** | **0.0215** | **0.0162** | 0.07241 | **0.0846** | **0.0692** | **0.0469** | 0.03968 | 26% |
| rw II | **0.0269** | **0.0215** | **0.0162** | 0.07909 | **0.0846** | **0.0692** | **0.0469** | 0.04485 | 27% |
| mm, VV | 0.0192 | 0.0185 | 0.0138 | 0.08330 | **0.0846** | 0.0615 | 0.0462 | 0.05798 | 45% |
| manual55, VV | 0.0154 | 0.0108 | 0.0100 | 0.08202 | 0.0692 | 0.0462 | 0.0362 | 0.06230 | 60% |
| mm, VV + rw I | **0.0269** | **0.0215** | **0.0162** | 0.08062 | **0.0846** | 0.0677 | 0.0462 | 0.05993 | 46% |
| manual55, VV + rw I | 0.0115 | 0.0092 | 0.0092 | 0.07411 | 0.0423 | 0.0308 | 0.0269 | 0.06563 | 61% |
| mm, VV + rw II | **0.0269** | **0.0215** | **0.0162** | **0.08494** | **0.0846** | 0.0677 | 0.0462 | 0.06571 | 52% |
| manual55, VV + rw II | 0.0115 | 0.0092 | 0.0092 | 0.07958 | 0.0423 | 0.0308 | 0.0269 | **0.07372** | 60% |

**Table 11: INEX 2005 CAS experiments on combining vague search and rewriting techniques evaluated using nxCG at different recall points and ep/gr.**

| | Strict | | | | Generalized | | | |
|---|---|---|---|---|---|---|---|---|
| **Exp. class** | **nxCG[10]** | **nxCG[25]** | **nxCG[50]** | **MAP** | **nxCG[10]** | **nxCG[25]** | **nxCG[50]** | **MAP** |
| rw I | 0.1000 | 0.1200 | 0.1022 | 0.00609 | 0.2687 | 0.2834 | 0.2645 | 0.04670 |
| rw II | 0.1889 | 0.1289 | 0.1022 | 0.00777 | 0.3030 | **0.2977** | **0.2679** | 0.05476 |
| mm, VV | 0.1444 | 0.1578 | 0.1556 | 0.01085 | 0.2811 | 0.2728 | 0.2529 | 0.07062 |
| manual55, VV | 0.1444 | 0.1444 | 0.1467 | 0.01056 | 0.2545 | 0.2553 | 0.2428 | **0.07296** |
| mm, VV + rw I | 0.1778 | **0.1711** | **0.1622** | 0.00904 | 0.2734 | 0.2641 | 0.2603 | 0.05899 |
| manual55, VV + rw I | 0.1667 | 0.1622 | 0.1489 | 0.00926 | 0.2427 | 0.2691 | 0.2469 | 0.06872 |
| mm, VV + rw II | **0.2000** | 0.1378 | 0.1089 | **0.01129** | **0.3092** | 0.2815 | 0.2366 | 0.05760 |
| manual55, VV + rw II | 0.1889 | 0.1333 | 0.1111 | 0.01113 | 0.3005 | 0.2943 | 0.2556 | 0.06896 |

**Table 12: INEX 2005 COS experiments on combining vague search and rewriting techniques evaluated using nxCG at different recall points and ep/gr.**

| | Strict | | | | Generalized | | | |
|---|---|---|---|---|---|---|---|---|
| **Exp. class** | **nxCG[10]** | **nxCG[25]** | **nxCG[50]** | **MAP** | **nxCG[10]** | **nxCG[25]** | **nxCG[50]** | **MAP** |
| rw I | 0.0588 | 0.0748 | 0.0595 | 0.00161 | 0.2715 | 0.2430 | 0.1894 | 0.03323 |
| rw II | 0.0588 | 0.0677 | 0.0571 | 0.00158 | 0.2872 | 0.2467 | 0.1898 | 0.03409 |
| mm, VV | 0.0765 | 0.0830 | 0.0854 | 0.00218 | 0.2912 | 0.2580 | 0.2258 | 0.06060 |
| manual55, VV | 0.0824 | 0.0759 | 0.0689 | 0.00218 | 0.2851 | 0.2585 | 0.2315 | 0.06872 |
| mm, VV + rw I | **0.0882** | 0.0901 | 0.0818 | 0.00215 | 0.2929 | 0.2686 | 0.2262 | 0.06168 |
| manual55, VV + rw I | 0.0824 | 0.0930 | **0.0871** | 0.00216 | **0.3040** | 0.2676 | **0.2395** | **0.07168** |
| mm, VV + rw II | 0.0765 | 0.0759 | 0.0748 | 0.00213 | 0.2873 | 0.2492 | 0.2168 | 0.05878 |
| manual55, VV + rw II | 0.0824 | **0.0954** | 0.0836 | **0.00219** | 0.2956 | **0.2688** | 0.2262 | 0.06891 |

element selection yields higher average precision. Furthermore, we show that automatically generated runs give comparable results to manually generated ones. Finally, the combination of vague selection and rewriting techniques approaches can boost the early precision, but it can also have negative influence on mean average precision.

The continuation of the work presented in this paper include the experimental evaluation of different scenarios for search in structured documents: the vague element search with different assignments of non-uniform down-weighting factors, e.g., using the value of exhaustivity and specificity in the assessments to derive more accurate down-weighting factors, or better chosen manual element name expansion lists, and their combination with rewriting techniques. For that, and for more realistic experimental setup we would need a new large (heterogeneous) collection of structured documents, with more semantical information esspecially in element names(e.g., Wikipedia, Lonely Planet).

## 6. ACKNOWLEDGMENTS

Many thanks to the Netherlands Organisation for Scientific Research (NWO) for funding the research described in this paper (grant number 612.061.210).

## 7. REFERENCES

[1] S. Amer-Yahia, C. Botev, and J. Shanmugasundaram. TeXQuery: A Full-Text Search Extension to XQuery. In *Proceedings of the 13th WWW Conference*, 2004.

[2] P. Boncz. *Monet: a Next Generation Database Kernel for Query Intensive Applications*. PhD thesis, CWI, 2002.

[3] A. Doan and A.Y. Halevy. Semantic Integration Research in the Database Community. *AI Magazine*, 26:83–94, 2005.

[4] D. Florescu and I. Manolescu. Integrating Keyword Search into XML Query Processing. In *Proceedings of the 9th International WWW Conference*, 2000.

[5] N. Fuhr and K. Großjohann. XIRQL: An XML Query Language Based on Information Retrieval Concepts. *ACM Transactions on Information Systems*, 22(2):313–356, 2004.

[6] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, Twente, The Netherlands, 2001.

[7] D. Hiemstra and V. Mihajlović. The simplest evaluation measures for xml information retrieval that could possibly work. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, 2005.

[8] G. Kazai and M. Lalmas. INEX 2005 Evaluation Metrics. In *Proceedings of the Fourth Initiative on the Evaluation of XML Retrieval (INEX), to appear*, 2006.

[9] G. Kazai, M. Lalmas, and A.P. de Vries. The Overlap Problem in Content-oriented XML Retrieval Evaluation. In *Proceedings of the 27th ACM SIGIR Conference*, 2004.

[10] G. Kazai, M. Lalmas, and S. Malik. INEX'03 Guidelines for Topic Developments. In *Proceedings of the Second INEX Worksop*, ERCIM Workshop Proceedings, 2004.

[11] Gabriella Kazai. Report of the inex 2003 metrics working group. In *Proceedings of the 2nd INEX Workshop*, ERCIM Workshop Proceedings, 2004.

[12] J. List, V. Mihajlović, A. de Vries, G. Ramirez, and D. Hiemstra. The TIJAH XML-IR System at INEX 2003. In *Proceedings of the 2nd INEX Workshop*, ERCIM Workshop Proceedings, 2004.

[13] V. Mihajlović, H.E. Blok, D. Hiemstra, and P.M.G. Apers. Score Region Algebra: Building a Transparend XML-IR Database. In *Proceedings of the ACM CIKM Conference*, 2005.

[14] V. Mihajlović, G. Ramírez, A.P. de Vries, D. Hiemstra, and H.E. Blok. TIJAH at INEX 2004: Modeling Phrases and Relevance Feedback. In *Proceedings of the Third INEX Workshop*, volume 3493 of *Lecture Notes in Computer Science*, 2005.

[15] G.A. Miller, C. Fellbaum, R. Tengi, S. Wolff, P. Wakefield, H. Langone, and B. Haskell. WordNet: A Lexical Database for the English Language.

[16] J. Pehcevski and A. Thom. HiXEval: Highlighting XML Retrieval Evaluation. In *Proceedings of the Fourth INEX Workshop, to appear*, 2006.

[17] V. Raghavan, P. Bollmann, and G. Jung. A critical investigation of recall and precision. *ACM Transactions on Information Systems*, 7(3), 1989.

[18] E. Rahm and P.A. Bernstein. A Survey of Approaches to Automatic Schema Matching. *The VLDB Journal - The International Journal on Very Large Databases*, 10:334–350, 2001.

[19] A. Trotman and R. A. O'Keefe. The Simplest Query Language That Could Possibly Work. In *Proceedings of the Second INEX Workshop*, ERCIM Publications, 2004.