Fine Tuning INEX

Alan Woodley
School of Software Engineering and Data Communications
Faculty of Information Technology
Queensland University of Technology
GPO Box 2434 Brisbane Q 4001 Australia
ap.woodley@student.qut.edu.au

Dr. Shlomo Geva
School of Software Engineering and Data Communications
Faculty of Information Technology
Queensland University of Technology
GPO Box 2434 Brisbane Q 4001 Australia
s.geva@qut.edu

ABSTRACT

Since 2002, INEX has been the benchmark for evaluating XML information retrieval (XML-IR) systems. INEX has based much of its evaluation methodology on that of existing workshops, albeit modified for the specific requirements of XML-IR. Due to some of the modifications, the time spent during evaluation phase of INEX takes a lot longer than comparable workshops. Here, we investigate ways to speed up the INEX evaluation process. We also investigate some structural changes and additional tasks that could be preformed at future INEX workshops.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software --- *performance evaluation*.

General Terms

Experimentation, Measurement

Keywords

Evaluation Methodology, System Pooling.

1. INTRODUCTION

In this position paper we propose and discuss several ideas that have been thrown around for quite some time at INEX workshops, on the mailing lists, and in private communications. In what follows we address some of the problems and proposed solutions and ideas in greater detail and some in less detail. Perhaps some of these can be discussed at the Glasgow workshop and adopted in future INEX collaborations.

Here we provided summaries of our proposals.

- Pooling submissions – Currently INEX uses a version of system pooling originally devised by Sparck Jones and Van Rijsbergen [3]. While system pooling has proven adequate, we propose a different pooling technique that may be superior. Our technique uses metasearch system to produce an assessment pool. Preliminary results indicate that metasearch pooling may be superior to the system pooling. We present a proposal to execute metasearch pooling at INEX 2005.
- Who contributes rare results? – Carrying on from the previous proposal, we know that under system pool results are taken from every submission, even those that will ultimately prove to be poor performing. The conjecture for including results from poor systems is that they may find rare (or even unique) relevant results. Here, we examine if this conjecture is true, and if INEX would be better served by not including results from poor performing systems in the pool (determined through committee ranking).
- Are inferred and additional results useful? - In INEX when a document contains at least one relevant result, the assessor must exhaustively score many other related elements in the document. To reduce assessment load some of these additional elements are automatically scored but the majority are assessed. The out-of-pool results are then added to the original pool. Unfortunately the process of exhaustive assessment is very time consuming. We argue that if the ranking of systems produced by an assessment set without exhaustive scoring is similar to the official INEX ranking, then assessment of out-of-pool results may not be necessary for all topics.
- Graded vs. Binary Assessment – In INEX, results are evaluated over 2 dimensions: exhaustiveness and specificity. In turn, these dimensions are scored over a range of 0-3. However, judging over these two dimensions is difficult and fraught with inconstancies between multiple assessors. However, binary relevance evaluation is much easier and quicker. In binary assessment each result is judged either relevant or irrelevant. We argue that if the ranking of systems produced by a binary assessment set is similar to the official INEX ranks, then two dimensional assessments may not be necessary.
- Miscellaneous issues:
  - Manual Runs – the pool of results may be enriched by through manual run submissions.
  - Re-using Topics – should topics from previous years be re-used? How?
  - XML Structure Changes to enrich the collection and the kind of tasks that can be performed by adding structure
  - Additional Tasks – What other tasks/sub-task could we undertake at INEX each year, to progress the state of XML-IR.

2. POOLING SUBMISSIONS

2.1 Background

In order to evaluate the systems we need a suitable baseline for comparison. In information retrieval we compare systems' results lists with a set of manual relevance assessments. In a sense, this allows us to compare system results lists with a results list from an 'ideal' system. This procedure allows us to produce standard recall and precision values for a system, and rank a set of systems according to these values. This approach has been followed since the early Cranfield experiments; however, several changes have been incorporated in order to scale to larger collections. Here we describe these changes, by comparing the methods used in the Cranfield experiments, to the methods employed by document retrieval experiments such as TREC, and finally the method used by INEX to handle structured information retrieval (Sections 4 and 5).

Early test collections (1960s, 1970s and early 1980s) such as Cranfield, were relatively small in size (less than 5MB). Since these collections contained a relatively small number of documents, human judgers were able to assess every document in the collection in relation its relevance to every query. With the emergence of TREC (1980s-current) much larger test collections (measuring in Gigabytes) became standard for laboratory information retrieval systems. Due to the large increase in collection size it was clear that the existing exhaustive method of evaluating every document in the collection was unfeasible. Therefore, a more scalable method of assessment was needed.

This challenge was handled by the use of system pooling, which was originally developed by Sparck-Jones and Van Rijsbergen [3]. The idea of system pooling is: for each topic, combine the top N results from each of the submission files. The results are merged, duplicates removed and are disassociated from their original submission. This becomes the system pool, and is sent to human judges for assessment. Results that are not in the system pool are automatically regarded as irrelevant. System pooling has proven to be an efficient means of evaluating systems, and has been used in several major international information retrieval workshops (for example, TREC, CLEF, NTCIR). Despite their proven worth, current evaluation methodologies have two shortcomings.

The first shortcoming is that the judges' decisions are inherently subjective. The notion of 'relevance' is at the very least a fuzzy concept, and people are bound to disagree on what constitutes a relevant result. Therefore, if two people are given the same set of results to judge, it is very unlikely that they will make exactly the same decision for every result in the set. The problem is even worse if relevance is judged on a graded, rather than binary scale. Incidentally, this is not a problem limited to pooling, and it could also occur with exhaustive assessment. However, research by Voorhees [4] concluded that while judges may disagree, the impact of their disagreement on systems ranking is not significant. The second shortcoming is that pooling inherently misses some relevant results. This is because all results ranked below the pool depth are automatically regarded as irrelevant. Research by Zobel [5] concluded that a system pool will only find about 70% of the relevant results in a collection; but, once again the impact of system ranking was not significant. However, it does raise the question of whether other, possibly more efficient or more effective pooling methods, could be used instead of system pooling.

2.2 Proposed INEX 2005 Experiment

Our proposal continues the work of Cormack et. al. [1] and Sanderson and Joho [2]. At present, the INEX Ad-hoc track uses a modified version of the Cranfield methodology that includes system pooling. The following six steps are undertaken annually:

1. Participants contribute topics (end user queries) and a subset of topics is selected for evaluation.

2. The topics are distributed to participants who run their search engines and produce a ranked list of results for each topic. The top 1500 ranked results for each topic are combined into a single submission file. Participates are allowed to send between 1 and 3 submissions, per task to INEX.

3. The top results from each submission are pooled together, disassociated from their originating submissions and duplicates are eliminated. We call this the s*ystem pool* (S*)* and say that it contains K_s results. We call the number of results taken from each result the *pool depth* (D_s) and it is currently set to 100.

4. The results in S are individually judged by the original topic contributors, who act as end users manually assessing the relevance of the results in terms of exhaustiveness and specificity. When judges find a document with a relevant result they must search the document for other relevant results, thus the size of S increases to $K_{s+i.}$ We shall refer to the results added to the pool as *inferred* results. We refer to the decisions made by the judges as *assessments*.

5. Using the assessment set and a standard evaluation module (inex_eval), the participating search engines are ranked in terms of performance (recall/precision) using several metrics.

6. Results are returned to participants who in turn write up and present their systems and discuss it at the workshop.

We propose replacing steps 3, 4 with the following.

3a. Produce a results pool (S) from the top N_s results from each submission in the usual manner. The pool depth N_s has to be determined in a certain manner and this is discussed later. We call this the s*ystem pool (S)* and say that it contains K_s results.

3b. In addition to the system pool, use a metasearch system to produce a merged ranked results list from all the submissions. From the list, select the top K_s results as a *metasearch pool* (M).

3c. Merge M and S (removing duplicates) to produce the *combined pool* (C) that contains K_c results.

Table 1: Metasearch Pool vs. System Pool

Assessments-Task	System			Metasearch		
	Average Precision/Topic	Average Recall/Topic	Average Unassessed/Topic	Average Precision/Topic	Average Recall/Topic	Average Unassessed/Topic
I-CO	0.131	0.471	125	0.146	0.507	383
II-CO	0.132	0.451	125	0.175	0.487	381
I-VCAS	0.208	0.440	10	0.224	0.460	211
II-VCAS	0.170	0.435	10	0.241	0.448	215

4. The results in C are judged by the original topic contributors, as if it was a traditional system pool. Again, inferred results are added to C, increasing its size to K_{s+i}

Submission evaluations are performed using the assessments in exactly the same manner as they were in previous years; so the assessors need not be aware of the source of the pool and scoring procedures need not change. The only problem that could arise is an increase in workload – only if the number of results in the combined pool is very large, since during judgments this would require much more work than the status-quo approach. However, by carefully controlling N_s, the pool depth, this can be avoided. We know that the size of the combined pool equal to the set union of the metasearch and system pools. In order to keep their weighting in the combined pool equal, we take the same number of results from both. However, we won't be able to predict the size of the combined pool since that will depend on the overlap between the system and metasearch pools. If the overlap is large, the size of the metasearch pool will be close to $K_{s,}$ the size of the system (and metasearch) pool. However, if the overlap is small, the size of the combined pool will be close to $2K_{s,}$ double the size of the system pool. The value of K_s can be easily chosen by experimentation since the process is an automated one. We may choose a value to limit the assessment workload rather than choose an arbitrary value.

After assessment is complete we will be able to determine which pool (M or S) has the higher level of recall. This will tell us which pooling method is the superior. If the metasearch pooling is superior then we could continue to use it in future INEX Workshops.

2.3 Preliminary Experiment

Before using metasearch pooling at INEX one would want to verify the validity of the approach. Therefore, we conducted a preliminary experiment to compare the performance of the proposed metasearch pool with the existing system pool. We conducted the experiment using the INEX 2004 submissions and both set of INEX 2004 assessments sets, and followed the proposed steps 3a and 3b in Section 2.2. The pool depth was set to 50 for the system pool to give us approximately 50% of the results that would be in a system pool of depth 100. In theory any metasearch method could be used to derive the metasearch pool, but we used the Borda Count approach. The Borda Count only requires a ranked list of results from constituent systems (that is - no relevance score per result) and it does not require any training. Evaluation of the pools was conducted as follows: For each pool, we calculated the total recall and precision values for each of the topics; then, we averaged the values across all topics. These averages are presented in Table 1, along with the average number of results not assessed. To produce the metasearch we used a pool depth of 500 results. We tested several pool depths (between 250 and 1500 results), but found that they all perform similarly. These results indicate that the metasearch pool is slightly superior to the system pool.

However, it must be noted that the assessment set is possibly/probably biased towards the system pool. This is because there were some results selected by the metasearch pool, which were not included in the assessments. Since these results were not assessed by a human judge they were automatically scored as irrelevant, even though in reality they could be relevant. Of course, the only way to know if these results are in fact relevant is to assess them, in the manner that we propose for INEX 2005. At the very least, our preliminary experiment has shown that the metasearch pool is as good as the system pool, with the possibility of out-performing it.

3. WHO CONTRIBUTES RARE RELEVANT RESULTS?

In the pooling method used in INEX results are added to the pool regardless of their originating system, even though some poor performing systems contribute very few relevant results (either at the element or document level) to the pool. There are two justifications for including results from poor performing systems in the pool. First, it keeps the pool unbiased, and removes the possibility of a 'self-fulfilling prophecy', whereby systems perform poorly because their retrieved results are not assessed. Second, even poor performing systems *may* find rare relevant results that are useful when added to the pool. But do we know for certain that poor performing systems find unique relevant results? If not, and if we can somehow identify poor system without completing the detailed manual assessment, should we not include their results in the pool (and include more results from better systems)?

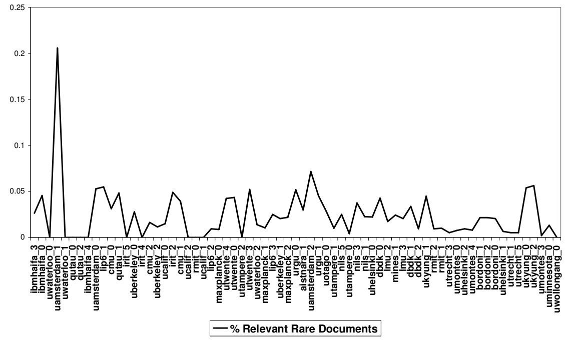

Figure 1: Percentage of Relevant Rare Results – CO – I

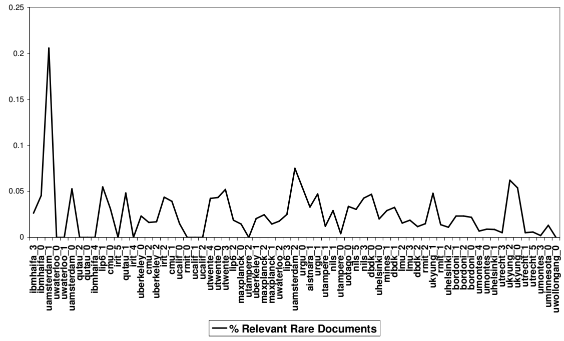

Figure 2: Percentage of Relevant Rare Results – CO - II

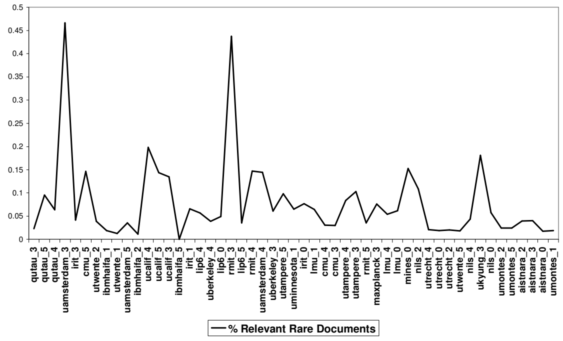

Figure 3: Percentage of Relevant Rare Results – VCAS - I

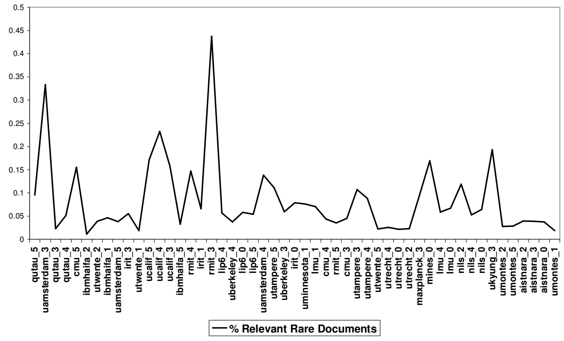

Figure 4: Percentage of Relevant Rare Results – VCAS - II

Here, we tested whether or not poor performing systems contribute a significant number of unique or rare relevant documents to the system pool. Originally we had planned to investigate the amount of unique results located by systems. However, the notion of "*uniqueness*" is clouded by the hierarchal nature of XML document, since we can not consider a result unique if its ancestor has been found by other system. For instance, imagine that a result **article[1]/sec[3]/p[5]** was only found by a one system (and therefore unique). However, if its parent node (**article[1]/p[3]**) was found by several or more systems can we then say that it is truly unique? We would say no, since the parent node obviously contains the child node. Furthermore, in INEX assessors when a relevant parent node is located assessors must judge each child node, arguably making the inclusion of the child node in the original pool moot. Hence, we concluded that for a system to have a unique result, neither the element nor any of its ancestors can a found by another system. And since the root ancestor node of all elements is the article node, in practice, this meant we where investigating the amount of unique articles/documents located by systems. However, after executing initial tests we realized that very few systems found unique document, therefore we extended our investigation to locate the amount of rare documents located by systems.

Out process was as follows: for each topic, we examined each system's top 100 results, examined their document name, and determined which documents were located by 5 or fewer systems. The number 5 was chosen as an estimate to what consists a rare document. This became our rare documents set (R). When formulating U, we only examined the top 100 results from each system because that corresponds to the pool depth used to derive the INEX system pool. We then classified each document in R as either relevant if it had a non-zero exhaustiveness or specificity value, and irrelevant otherwise. We conducted our experiments using both the CO and VCAS tasks and both 2004 assessments sets. Figure 1 – 4 are the plots, and for each system show the percentage of relevant rare documents. The systems are sorted according to each system's official INEX rank with the highest scoring systems on the left. As the results indicate there doesn't appear to be a correlation between a system's performance and the number of relevant rare documents. This indicates that it is valid to pool results from poor performing systems.

Table 2: CO Rank Correlations – Official vs. Out-Of-Pool

Assessment/Correlation	Aggregate	Strict	Generalized	SO	E3S32	E3S321	S3E32	S3E321
I-Spearman-rho	0.996	0.997	0.989	0.986	0.996	0.997	0.996	0.993
II-Spearman	0.995	0.996	0.990	0.990	0.996	0.997	0.996	0.989
I-Kendall-tau	0.965	0.968	0.937	0.936	0.962	0.977	0.964	0.953
II-Kendall-tau	0.960	0.960	0.947	0.948	0.960	0.973	0.965	0.942

Table 3: 2004 VCAS Rank Correlations – Official vs. Out-Of-Pool

Assessment/Correlation	Aggregate	Strict	Generalized	SO	E3S32	E3S321	S3E32	S3E321
I-Spearman-rho	0.989	0.983	0.995	0.992	0.994	0.994	0.984	0.987
II-Spearman	0.990	0.993	0.996	0.989	0.996	0.992	0.987	0.987
I-Kendall-tau	0.942	0.923	0.969	0.950	0.961	0.961	0.923	0.927
II-Kendall-tau	0.950	0.951	0.970	0.939	0.962	0.956	0.933	0.936

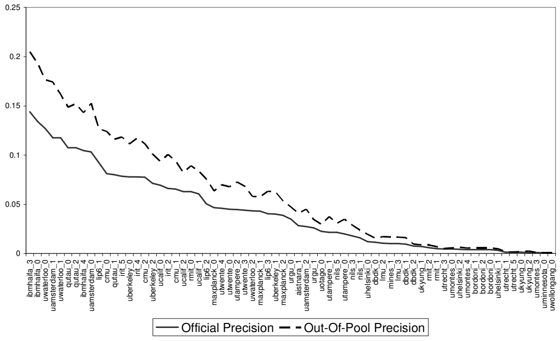

Figure 5: Official MAP vs. Out-Of-Pool MAP(Aggr) – CO – I

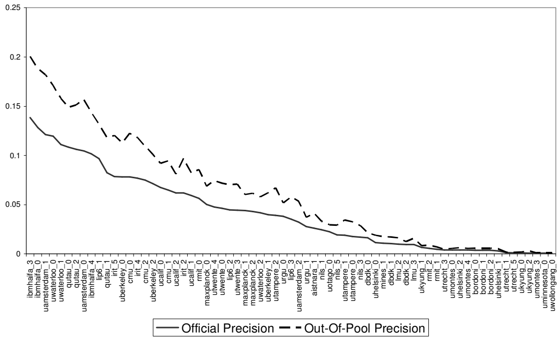

Figure 6: Official MAP vs. Out-Of-Pool MAP(Aggr) – CO - II

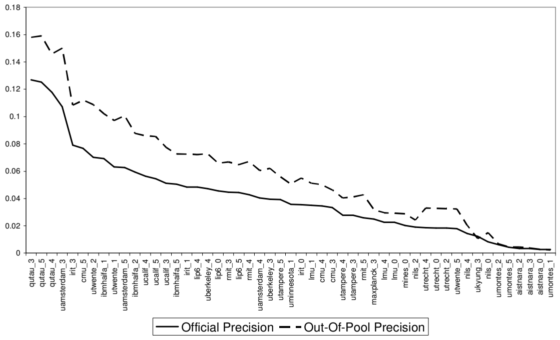

Figure 7: Official MAP vs. Out-Of-Pool MAP(Aggr) - VCAS - I

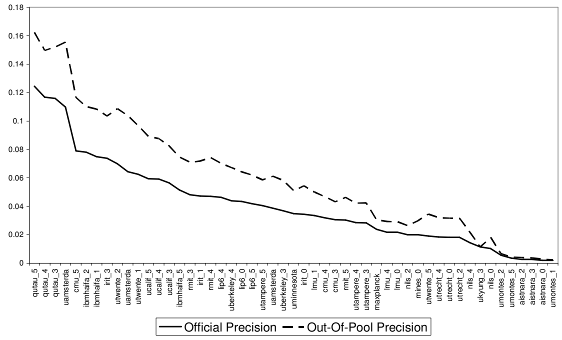

Figure 8: Official MAP vs Out-Of-Pool MAP(Aggr) - VCAS - II

4. ARE OUT-OF-POOL RESULTS USEFUL?

During the INEX evaluation phase, if a document contains one or more relevant results, the assessor must examine all the other elements in the document, and individually assess each for relevance. Assessors can also add incidental results that are picked up by inspection. These results are then added to the original pool as 'out-of-pool results'. The justification for this is two-fold. First, the results may have been identified by systems, but at below position 100 thus escaping the system pool. Second, it will help to 'future proof' the evaluation set since judges may locate results beyond the capabilities of current search engines, which may be found by future, more sophisticated search engines. We do not dispute the validity of these motivations; however, the process has one major drawback - it is very time consuming. We already know that the INEX evaluation process takes a lot longer than comparable workshops. We believe that by removing the evaluation of out-of-pool assessment, thereby having judges assess only returned results, we could greatly reduce the time required for assessment. However, there is a risk involved in not assessing all AAAelements: the rank of systems may significantly change when inferred results are not included in the assessment pool. Here, we investigate this hypothesis, by producing a rank of systems inferred results are not included in the assessment pool. Here, we investigate this hypothesis, by producing a rank of systems using the original pool, without the inclusion of 'out-of-pool' results. We then compare this systems ranking to the official systems ranking. We argue that if the system ranks are similar, then out-of-pool assessment may not be necessary.

We conducted our experiments in the following manner. First, we parsed the INEX 2004 assessments and removed all the out-of-pool results. This allowed us to have a set of results consisting of the original pool. Then we executed the inex_eval using the original pool assessments, and produced a ranked list of systems. Tables 2 and 3 present the correlation between the two. Table 2 is the correlation between ranks for the INEX 2004 CO Task and Table 3 is the correlation between ranks for the INEX 2004 VCAS Task. We compared systems using both evaluation sets and metrics used in INEX 2004. We used two correlation measures: Spearman-rho and Kendall-tau. Figures 9-12 plot the Aggregate Mean Average Precision (MAP) for participants in the CO and 2004 VCAS tasks, using both sets of assessments. The two systems rankings are very similar. This indicates that the assessment of out-of-pool results is not vital for accurately discriminating between XML-IR systems; this raises the possibility of having judges only assess results in the original Ad-hoc pool. However, even if we choose to keep out-of-pool results in the Ad-hoc task, current or future tasks/tracks may choose to eliminate the process without significantly impacting on the ranking of systems.

5. GRADED VS BINARY RELEVANCE ASSESSMENTS

The objective of XML-IR is two-fold. First, systems must find XML elements (results) that match the subject area specified in a user query. Second, systems must choose the most appropriately sized elements to return to the user, and rank accordingly. To correspond with this dual retrieval objective, INEX has extended the notion of relevance to cover two dimensions - exhaustivity and specificity**.** Each dimension is judged as one of four values from zero to three, where zero is judged as irrelevant. Also, an element cannot have a zero score in one dimension and a non-zero score in another. This produces nine possible levels of relevancy, plus a single non-relevant level. In contrast, most document-level evaluation methods classify documents as relevant or non-relevant.

In theory, INEX's use of two dimensions, and graded scaled makes sense, since we assume that as one propagates up an XML tree, the values for the two dimensions will change. The observation is that since ancestor nodes contain a larger amount of information, they tend to be more exhaustive than descendants. Conversely, relevant descendant nodes tend to be more specific than their ancestors, as they contain less irrelevant information. The graded INEX evaluation process is very time consuming and prone to great disagreement between multiple judges. However, it should be much easier and quicker to judge result relevancy on a binary scale (that is - as either relevant or irrelevant). Here, we investigate this hypothesis by producing a ranked list of systems evaluated using binary assessment, and comparing it with the official INEX systems rank. We propose that if the two systems rankings are similar, then quantized assessment may not be necessary.

We conducted our experiment in the following manner. First, we parsed the INEX 2004 assessments and changed the value of every non-zero score exhaustiveness or specificity score to 3/3. This allowed us to simulate binary relevance. Then we executed the INEX evaluation module (inex_eval) using the binary assessments, and produced a ranked list of systems. Tables 4 and 5 presents the correlation between the two systems ranks. Table 4 is the correlation between ranks for the INEX 2004 CO Task and Table 5 is the correlation between ranks for the INEX 2004 VCAS Task. We compared systems using both evaluation sets and metrics used in INEX 2004. We used two correlation measures: Spearman-rho and Kendall-tau. Figures 9-12 plot the Mean Average Precision (MAP) for participants in the CO and 2004 VCAS tasks, using both sets of assessments. The results show that the two systems are similar, but significantly different. This indicates that graded assessment is important for accurately discriminating between the performances of XML-IR systems. This validates INEX's choice of using graded results for its Ad-hoc task. However, since the systems ranks are reasonably similar, particularly for the Generalized and SO metrics, it raises the possibility of using binary relevance in situations were time is a major constraint (such as the interactive track).

Table 4: CO Rank Correlations – Official vs. Binary

Assessment/Correlation	Aggregate	Strict	Generalized	SO	E3S32	E3S321	S3E32	S3E321
I-Spearman-rho	0.957	0.882	0.988	0.973	0.928	0.917	0.893	0.909
II-Spearman	0.950	0.862	0.985	0.970	0.937	0.932	0.902	0.928
I-Kendall-tau	0.875	0.788	0.940	0.901	0.837	0.820	0.789	0.812
II-Kendall-tau	0.862	0.788	0.933	0.893	0.850	0.837	0.790	0.819

Table 5: VCAS Rank Correlation – Official vs. Binarv

Assessment/Correlation	Aggregate	Strict	Generalized	SO	E3S32	E3S321	S3E32	S3E321
I-Spearman-rho	0.961	0.914	0.996	0.983	0.900	0.901	0.960	0.969
II-Spearman-rho	0.957	0.888	0.986	0.986	0.877	0.874	0.952	0.962
I-Kendall-tau	0.875	0.811	0.963	0.921	0.796	0.803	0.867	0.889
II-Kendall-tau	0.862	0.778	0.925	0.926	0.765	0.760	0.858	0.879

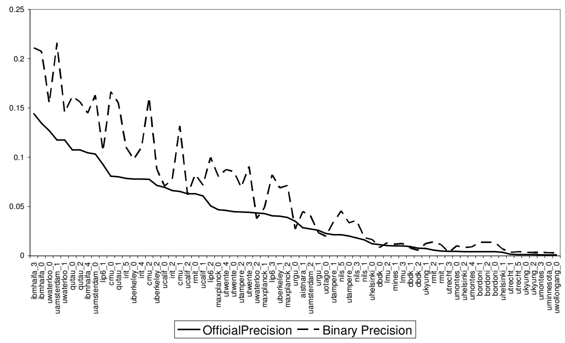

Figure 9: Official MAP vs. Binary MAP – CO –I

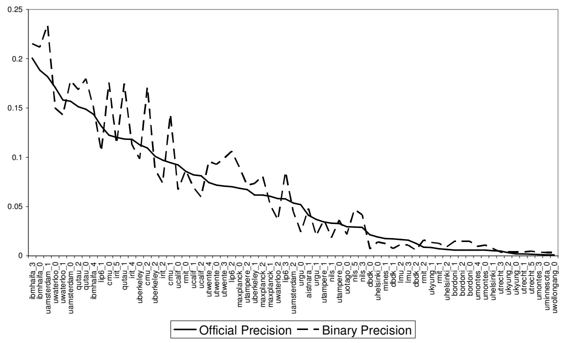

Figure 10: Official MAP vs. Binary MAP – CO -II

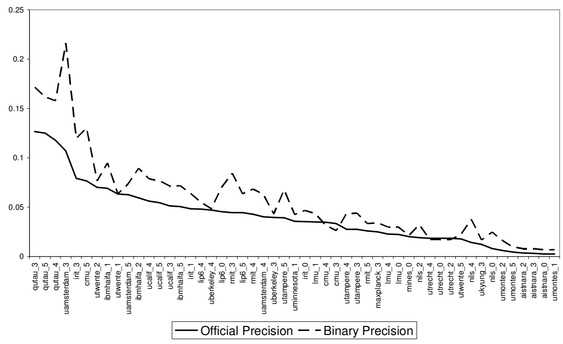

Figure 11: Official MAP vs. Binary MAP – VCAS -I

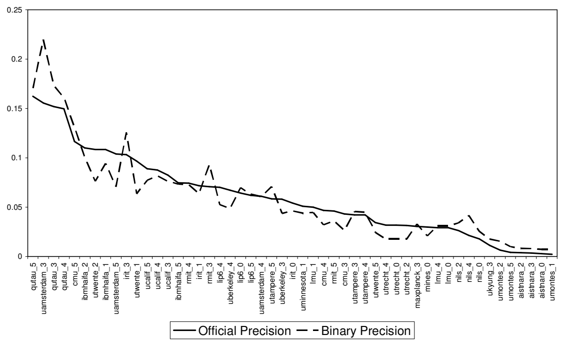

Figure 12: Official MAP vs. Binary MAP – VCAS -II

6. MANUAL RUNS

Even if we are successful in expanding the proportion of relevant results in the results pool through metasearch, it is still limited by the ability of the search engines to automatically find results. It is possible to increase the size of the results pool by including the results of manual runs to the pool. Manual run results can be performed either through a semi-automated relevance feedback process, or through more elaborate manual intervention of assembling relevant result sets. In Semi-automated mode participants evaluate results and provide relevance scores for the top N results. The search engine then automatically utilizes the feedback to modify the search strategy (for example - by adding/removing keywords), by changing the ranking strategy (for example - by re-ranking results through a change in the scoring parameters), or both. In the more elaborate manual mode users can change the query in any way desired through iterative use of the search engine and by manually eliminating irrelevant results and re-ordering results (that is - manual ranking).

There are two ways in which evaluation of systems can then take place. The evaluation of automatic runs can be carried out as in previous INEX workshops. The important contribution of manual runs is in providing a result pool that is closer to the 'absolute' results pool, or a baseline, against which to compare automatic results. Then there is also benefit in comparing the average performance of automatic runs with the average performance of manual runs. This average performance can be computed on the best N performing systems, or by comparison that is based on a metasearch pool that is obtained as described in section 2. There is another comparison that could be made – between manual runs - but there is a problem: the quality of the manual submissions depends critically on the competence of the persons who use the search engines. Although it is possible to conduct controlled experiments that will attempt to eliminate this problem, there seems to be no simple enough way other then by involving numerous users, each of whom will be required to use several different systems. This seems infeasible under the INEX mode of operation and resource constraints.

7. RE-USING TOPICS

Past topics can be very useful for at least two reasons. The most obvious reason is of course the reduced assessment load. The second reason is the ability to quantify the improvement in search engine technology over time. Re-used topics should lead to result pools that include additional relevant results. Of course there is no need to re-assess the entire result pool of a re-used topic. Previously assessed results can be assigned the known scores. This leaves a smaller residual result pool for assessment. There is always a risk, when re-using topics, that search engines that were designed with the use of past assessments, are over-fitted to those assessments. However, this can be tested by comparing the performance of systems over re-used topics with the respective performance over new topics. This evaluation can reveal whether this is a problem that is specific to some systems, to all systems, or perhaps to none. If we discover that over-fitting does not occur at significant levels then we can re-use topics with confidence in future workshops.

8. XML Structure Changes

There are several generic XML DTD changes that we would like to propose and that we believe will enrich the INEX collection, the type of tasks that may be pursued, and possibly improve the results that can be obtained. The changes may also assist in result assessment and run evaluation processes. These proposed changes are discussed below

8.1 Text Segmentation.

Segmentation can always be applied to text type elements. For instance, we may wrap sentences within XML tags <s> ... </s>. This can be useful in several ways. Question answering tasks that require highly specific responses can benefit from the ability to pinpoint relevant sentences. Furthermore, in evaluation it may be useful to be able to assess individual sentences as relevant. For instance, sometimes only a small part of a long paragraph is relevant and at present there is no way to assess at below the paragraph level.

8.2 Part of Speech Tagging (POS)

Part of Speech (POS) tags can be added with fairly high accuracy to the collection. State of the art POS taggers are claimed to operate with accuracy of better than 95%. Apart from being useful in supporting NLP functionality POS tagging can be very useful in facilitating very simple selectivity in term searching and can probably assist in improving overall accuracy – merely by adding some elementary semantics to index terms. By adding POS tags attributes of sentences in the collection all participants at INEX will be able to use such information – or ignore it – and it should foster greater interest in using POS techniques in IR.

8.3 XPointers and XLinks

A standard mechanism of referencing, namely XPointer and XLink, exist in XML but are not used within the INEX collection. It is possible to convert the collection to support XLinks and XPointers and we propose to pre-process the INEX collection and augment it as follows:

- Replace references within the article body, pointing to bibliography entries in the References section, by XPointers. Furthermore, insert XPointers in each bibliography entry pointing back to each element in the article that references that bibliography entry.
- Replace references within the article body, pointing to figures in the article, by XPointers. Furthermore, insert XPointers in each Figure pointing back to each element in the article that references that figure.
- Some bibliography entries in the INEX collection refer to other articles within the INEX collection itself. Insert

XLinks in each bibliography entry that makes such a reference.

These additions can simplify processing of common operations such as composing a response for a query from multiple relevant components that are interlinked. For instance, it would be possible to easily support queries that in the past were excluded at INEX, such as “Get figures of the CORBA architecture together with the relevant text that explains it”.

8.4 Element Size attributes

We could augment each element with a set of size attributes. These attributes could facilitate various operations in searching and in ranking results, as well as in assessing results. The following size elements could be considered:

- Number of children (C).
- Number of descendents (D)
- Number of Sentences (S)
- Number of words (W)

For instance, <sec C="7" D=”43” S=”234” W=”1432”> ... </sec> indicating that the section has 7 children elements and 43 descendents, 234 sentences, and 1432 words in total. This information can be omitted from very small elements as it is unlikely to be useful.

This information can be used, for instance, in determining the size ratio of relevant to irrelevant parts of a given XML component. It could be used in evaluating the specificity of a result element for the purpose of ranking and also for the purpose of automating an assessment tool for INEX.

9. ADDITIONAL TASKS

The performance of systems over the INEX Ad-hoc task varies greatly. Some of the performance differences can be attributed to specific system characteristics. It is usually impossible to assess precisely which properties of a given system are responsible for its performance (or lack thereof…). Contributing factors can be superior indexing structures, the use of insightful heuristics, rigorous analysis, a user model that is faithful to some general traits of assessors, and so on. Many of these are embedded deeply and implicitly within search engines and therefore the ranking of entire systems is the only way for us, as a community, to assess the merits of individual approaches. We would like to explore particular aspects of search engines in isolation. In the following we propose a few tasks that might help us achieve this. Importantly, none of these tasks require any additional assessments – evaluations can be fully automated using the standard ad-hoc track assessments.

9.1 Query Expansion sub-task

As a pre-text to our suggestion, we would like to briefly look at the Natural Language Query (NLQ) task. One of the sub-tasks is the translation of a description element into a NEXI title expression. The idea is to evaluate all the NLQ approaches using one or more baseline search engines. In this manner, any performance variation can be attributed to superior translation of a query into a NEXI specification, rather than to any inherent property of a particular search engine. It is possible to isolate the NLQ contribution from the contribution of the implementation of searching and ranking.

We would like to propose the extension of this approach to query expansion. One of the critical success factors in query evaluation is query expansion. Most queries are expanded by the addition of terms, and in the case of CAS queries by the addition of tags, to the original query. In order to isolate the contribution of query expansion, from the contribution of the searching and ranking processes that follow, we propose the following task. In similar manner to the NLQ task, given a set of INEX topics, produce a set of expanded topics. Each NEXI topic in the original set is expanded – transformed into a new NEXI expression. The submission thus consists of a set of new topics rather than the submission of results from retrieval runs. All the expanded sets of topics will then be run on one or more of the better performing search engines and ranked through their indirect performance with the baseline search engines. The relative ranking can thus be attributed to query expansion rather than to the underlying search engines.

9.2 Ranking-Only sub-task

A natural extension to this approach is a ranking-only sub-task. Given a topic and a bag of results (that is - an unordered set, possibly derived using a metasearch technique as discussed in Section 2), the task is to rank the results. The task here is solely the ordering of results. It requires scoring and ranking only and therefore does not require the implementation of a search engine. This task may provide greater insights into which ranking and scoring strategies work better, in isolation from query expansion and indexing/searching strategies. It is not obvious how to separate the functions, but if it is possible than this would be a worthwhile investigation.

9.3 Ontology Mining

Currently, most search engines accept a list of terms, or reduce a natural language sentence to a list of terms by ‘cleaning up’ noise words. Search engines typically use query expansion techniques (for example - addition of synonyms or related terms) to explicitly augment the implicit correlation between query terms. It is difficult to do this in a user-free context since different users may benefit from expanded queries in different ways, depending on individual interests and contexts. Furthermore, query expansion may be context dependent in itself. The same query terms may be closely associated with one set of terms in one context and with a completely different set of terms in relation to another context. The WordNet ontology is perhaps the best known example; however, it is very generally language oriented rather than collection specific.

Identifying sets of terms for distinct contexts is a difficult problem. The term 'Ontology' is understood in this context to mean a thesaurus that can identify the use of related terms in different specific contexts. Unlike an ordinary thesaurus, which is language based, very general, and not context sensitive, ontology has higher granularity and is context sensitive. In this task we would like to study techniques for mining ontologies from XML document collections. The aim is to automatically construct and maintain ontologies that capture the possible semantic information in XML documents, including term taxonomy, 'interesting topics', frequent terms and phrases, associations between terms and phrases, and so on.

Of course it is impractical to generate domain ontologies manually. So the trick is to take a large collection and to perform data mining operations to discover associations, co-occurrences, similar uses, and so on. This is not new - there is a lot of research in ontology mining. However, the XML collection potentially offers us much richer semantics to create associations with. Rather than merely word proximity we now have terms appearing together in <keywords> elements, or in <abstract> , <author>, <biography>, <theorem>, and so on. It should be possible then to take advantage of this rich semantics in mining ontologies. But how can we identify and quantify any potential improvement?

We propose to study Ontology Mining in XML collections in the INEX context. The task that we propose is closely related to the task described in section 9.1, Query Expansion.

Given the INEX collection of 18 Journals and Magazines:

- Automatically generate a comprehensive ontology from the XML collection
- Given a set of topics (queries) expand the queries with related terms derived from the ontology – that is, produce an augmented set of queries

The idea is to evaluate ontology mining systems through their utility in query expansion - we use a set of standard search engines to evaluate the original and the expanded queries and we measure the improvement (if any). The baseline measurement is the performance of the standard search engines with the original queries. The expanded queries are also executed by the same baseline search engines. If the ontology is accurate, and if there is an advantage to query expansion by obtaining more comprehensive and accurate results, then we can rank approaches to ontology mining by the amount of improvement.

10. SUMMARY

Here we presented several ideas that that could be incorporated into INEX. Some proposals relate to new tasks or extended functionality and others to different assessment procedures. We believe that it is possible to obtain an increase in evaluation efficiency by trading off evaluation effectiveness. Regardless, we feel that the proposal will lead to spirited discussion and debate at the INEX Workshop on Element Retrieval Methodology and with respect to IR in XML in general.

11. ACKNOWLEDGMENTS

We wish to acknowledge the direct and indirect contribution of all the participants of the INEX workshops and the various INEX mailing lists for raising and discussing many of the issues that we address in this document.

12. REFERENCES

[1] G. V . Cormack, C. R. Palmer, and C. L. A. Clark, "Efficient Construction of Large Test Collections", In *Proceedings of The 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* ACM Press, Melbourne, Australia, 1998

[2] M. Sanderson, H. Joho, "Forming Test Collections with no System Pooling", In *Proceedings of The 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Sheffield, Great Britain, 2004, pp. 33-40.

[3] Sparck-Jones, and C. J. Van Rijsbergen, "Report on the Design Study for the 'Ideal' Information Retrieval Test Collection", *British Library Research and Development Report 5428*, Computer Laboratory, University of Cambridge, 1975.

[4] E. M. Voorhees, "Variations in Relevance Judgements and the Measurement of Retrieval Effectiveness", In *Proceedings of The 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Melbourne, Australia, 1998, pp. 315-323.

[5] J. Zobel, "How Reliable are the Results of Large Scale Information Retrieval Experiments", In *Proceedings of The 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Melbourne, Australia. 1998.