Proceedings of the International MultiConference of Engineers and Computer Scientists 2012 Vol I,
IMECS 2012, March 14 - 16, 2012, Hong Kong

A Comparative Study of MySQL Functions for XML Element Retrieval

Chuleerat Jaruskulchai, *Member, IAENG,* and Tanakorn Wichaiwong, *Member, IAENG*

***Abstract*—Due to the ever increasing information available electronically, their size is growing rapidly. Since, XML documents have further information; document representation of these might be changed. Term weight technique is a mechanism to retrieve documents. In XML element retrieval applications, the weighting algorithm plays an important role and it greatly affects the precision and recall results of the retrieval systems. MySQL's full text search algorithm is widely applied into retrieval flat document. However, for the semi-structure document hasn't evaluation. In this paper, we have to investigate the weighting functions of MySQL and performed a comparative study of weighting schemes processing. Our objective of the study was to find out the appropriate features to achieve the effectiveness of XML element retrieval. The experiment results show the Natural Language use of vector space model function performs better than other methods measured by INEX evaluations.**

***Index Terms*—MySQL Full Text Search, XML Retrieval, Ranking Strategies.**

I. Introduction

THE weighting functions of information retrieval [1], [2] have been widely studied. Many researchers have been studied base on different weighting functions. We developed XML [3] information retrieval system by using MySQL [4]. For this purpose, our study addressing on the comparison of various term weighting base on MySQL's features and contribution of weighting schemes area of understanding features that influence the automatic indexing potential of terms.

This paper is organized as follows; Section 2 reviews related works. Section 3 explains our experiments, conclusions and further work are drawn in Section 4.

II. Related Works

A. MySQL Full Text Search Overview

MySQL Full Text Search (FTS) [4] has multiple table types; **MyISAM** has support for an FTS index. It is implemented differently than other FTS systems. Instead of storing the inverted lists in a tightly packed binary format, it uses a normal two-level B-Tree. The first level contains records of the form (word, count), where the count is the number of documents in which the word appears. The second level contains records of the form (weight, rowID), where weight is a floating point value signifying the relative importance of the word in the document pointed to by rowID. The full text search support in MySQL uses the following constructs.

- **MATCH** takes a comma-separated list that names the columns to be searched.
- **AGAINST** takes a string to search for and an optional modifier that indicates what type of search to perform. The search string must be a literal string rather than a variable or a column name

MySQL has three types of weighting functions:

- **Natural Language Searches**: By default, the MATCH function performs a natural language search for a string against a text collection. A collection is a set of columns included in FULLTEXT indices. The search string is given as the argument to AGAINST function. For each row in the table, MATCH function returns a relevance value; that is, a similarity measure between the search string and the context. When MATCH function is used in a WHERE clause, as in the example shown earlier, the rows returned are automatically sorted with the highest relevance first. Relevance is computed based on the number of words in the row, the number of unique words in that row, the total number of words in the collection, and the number of documents that contain a word. The first query sorts the results by relevance whereas the second does not. However, the second query performs a full table scan and the first does not. The first may be faster if the search matches few rows; otherwise, the second may be faster because it would read many rows anyway. For example, although the word "mysql database" is present in every row in the column "details" of "inex08" table shown earlier as below;

  SELECT xPath, MATCH (details) AGAINST ('mysql database' IN NATURAL LANGUAGE MODE) AS score FROM inex08 WHERE MATCH (details) AGAINST ('mysql database' IN NATURAL LANGUAGE MODE);

- **Boolean Searches**: MySQL can perform boolean searches using the IN BOOLEAN MODE modifier. With this modifier, certain characters have special meaning at the beginning or end of words in the search string. In the following query, the + and - operators indicate that a word is required to be present or absent,

Manuscript received October 10, 2011; revised December 01, 2011. This work was supported in part by the Graduate School of Kasersart University.

C. Jaruskulchai received her Bachelor of Education from Chulalongkorn University, and she received a Master's degree in Applied Science with a specialty in Computer Science from the National Institute of Development Administration in 1978. She received a Ph.D. from George Washington University, Washington, D.C., in 1998. Currently, she served as the Chair of the Ph.D. program in Computer Science in the Department of Computer Science, Kasetsart University, Bangkok. e-mail: fscichj@ku.ac.th

Wichaiwong received his B.BA. degree in Business Computing from North Eastern University, Khonkean. He received a Master's degree in Computer Science from Kasetsart University, Bangkok, in 2008. He is currently pursuing the Ph.D. degree in Kasetsart University, Bangkok. His current research interests include the area of information retrieval, XML retrieval and XML query language and database. e-mail: g5184041@ku.ac.th

respectively, for a match to occur. In implementing this feature, MySQL uses what is sometimes referred to as implied boolean logic, in which

1) $+$ stands for AND
2) $-$ stands for NOT
3) [no-operator] stands for OR

For example, the query retrieves all the rows that contain the word "mysql" but that do not contain the word "database" is present in every row in the column "details" of "inex08" table shown earlier as below;

SELECT xPath FROM inex08 WHERE MATCH (details) AGAINST ('+mysql -database' IN BOOLEAN MODE);

- **Query Expansion**: FTS supports query expansion (and in particular, its variant "blind query expansion"). This is generally useful when a search phrase is too short, which often means that the user is relying on implied knowledge that the full-text search engine lacks. For example, a user searching for "database" may really mean that "MySQL", "Oracle", "DB2", and "RDBMS" all are phrases that should match "databases" and should be returned, too. This is implied knowledge. Blind query expansion (also known as automatic relevance feedback) is enabled by adding WITH QUERY EXPANSION following the search phrase. It works by performing the search twice, where the search phrase for the second search is the original search phrase concatenated with the few most highly relevant documents from the first search. Thus, if one of these documents contains the word "databases" and the word "MySQL", the second search finds the documents that contain the word "MySQL" even if they do not contain the word "database". Because blind query expansion tends to increase noise significantly by returning no relevant documents, it is meaningful to use only when a search phrase is rather short.

  SELECT xPath FROM inex08 WHERE MATCH (details) AGAINST ('database' WITH QUERY EXPANSION);

The element scoring of its context using vector space model of MySQL-FTS as following:

$$LeafScore(e,Q) = \sum_{t \in Q} Q_t * W_t * L_t \tag{1}$$

$$L_t = \log[\frac{tf_t + 1}{len(e)}] * \frac{U}{(1 + 0.0115 * U)} \tag{2}$$

$$W_t = \log\langle \frac{N - e_t}{e_t} \rangle \tag{3}$$

$$Q_t = Qtf_t \tag{4}$$

Note that;
$LeafScore(e,Q)$ measures the relevance of element e to a query Q.
W_t is the inverse element frequency weight of a term t.
tf_t is the frequency of a term t occurring in an element e.
$len(e)$ is the length of an element e.

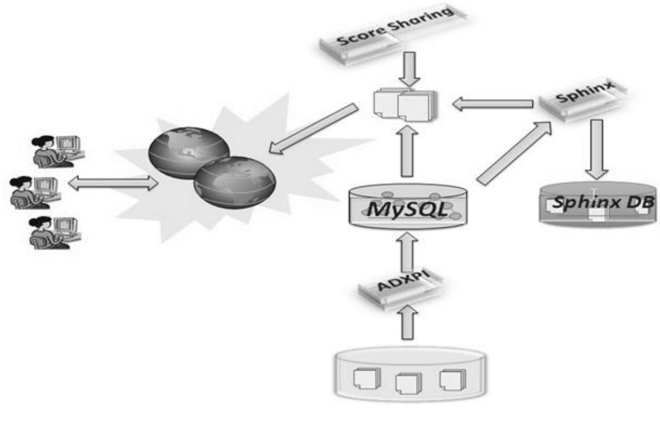

Fig. 1. XML Information Retrieval (XMLIR) System Overview

U is the number of unique terms in element e.
N is the total number of an element in the collection.
e_t is the total element of a term t occur.
Qtf_t is the frequency of a term t occurring in a query Q.

B. The Score Propagation

Score Propagation [5], [6], [7] is used to rank elements based on leaf-node indexing. Scoring is propagated upward to ancestors. The resulting relevance score for each element is a weighted accumulation of ranking scores of an element's children. This strategy was presented by the Gardens Point XML-IR (GPX) [5], [6], [7], a propagation method that was proposed as a bottom-up scheme (BUS) [8]. For example, for each element with only one relevant child element, the child should be ranked higher. Otherwise, this element ranks higher than their child. The assignment of a numeric score to a document given a query can be represented as follows:

$$score(e,q) = D(m) * \sum_{e,c} score(e_c, q) \tag{5}$$

Note that;
$D(m)$ is the smoothing parameter set as follows.
If e has one child, then $D(m)$ = 0.49.
Otherwise, $D(m)$ = 0.99

C. XML System Overview

Our system uses a relational database as a storage back end and query processing methods are based on Full-Text Search. In the following, we discuss the schema setup using MySQL [4] engine generally available release: 5.1.51.

In Fig. 1, depicts the overview of XML retrieval system. For the initial step, we consider a simplified XML data model, but disregarding any kind of markup including comment, link and attributes. The main components of the XML retrieval system are including;

- The Absolute Document XPath Indexing (ADXPI) [9], when new documents are entered, the indexer parses and analyzes the tag and content data to build the list of leaf nodes.
- The score sharing method, which allows assigning the parent scores by sharing score from leaf node to their parents by a top down scheme approach.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

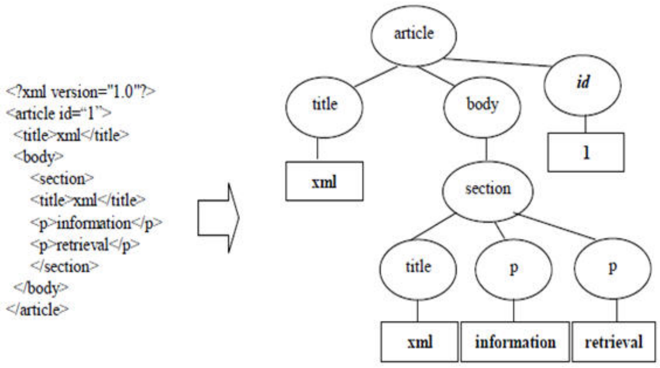

Fig. 2. The Example of XML Element Tree

D. The Score Sharing Function

In a previous study [10], we compute the scores of all elements in the collection that contain query terms. We consider the scores of elements by accounting for their relevant descendants. The scores of retrieved elements are now shared between the leaf node and their parents in the document XML tree according to the following scheme.

$$Score(PNode) \leftarrow Score(PNode) + \langle LeafScore * \beta^n \rangle \quad (6)$$

Note that;
$PNode$ is a current parent node.
β is a tuning parameter.
$IF\langle 0-1 \rangle\ THEN$ preference is given to the leaf node over the parents.
$OTHERWISE$, preference is given to the parents.
n is the distance between the current parent node and the leaf node.

E. The ADXPI Indexer

According to previous studies [9], a single inverted file can hold the entire reference list, while a suitable indexing of terms can support the fast retrieval of term-inverted lists. To control for overlap and reduce the cost of joined on relational database; we use the ADXPI scheme to transform each leaf element level into a document level. For instance, take a document named ”x1” as shown in Fig.2.

Fig. 2, depicts the example of the XML element trees then we can build an index by ADXPI expression identifies a leaf node that has text contain within the document, relative to document and their parents are following:

- x1/article[1]/title[1]: ”xml”
- x1/article[1]/body[1]: ”xml”
- x1/article[1]/body[1]/section[1]: ”retrieval”
- x1/article[1]/body[1]/section[1]/title[1]: ”xml”
- x1/article[1]/body[1]/section[1]/p[1]: ”information”
- x1/article[1]/body[1]/section[1]/p[2]: ”retrieval”

III. Experiment Setup

In this section, we present and discuss the results based on the INEX collection. We also present the results of an empirical sensitivity analysis of various parameter performed on a Wikipedia collection. This experiment was performed on Intel Pentium i5 4 * 2.79 GHz with 6 GB of memory, Microsoft Windows 7 Ultimate 64-bit Operating System and Microsoft Visual C♯.NET 2008.

A. Data sets

The document collections are following; the INEX-IEEE document collection [11] contains total of 16,819 articles from 24 IEEE Computer Society journals, covering the period of 1995-2005 and totalling 764 megabytes in size and 11 million elements in its canonical form. The Wikipedia XML Corpus of the English Wikipedia in early 2006 [12] that contains 659,338 Wikipedia articles and the total size is 4.6 GB without images and 52 million elements. On average an article contains 161.35 XML nodes, where the average depth of a node in the XML tree of the document is 6.72.

B. INEX Evaluations

As for INEX-IEEE effectiveness [11], we refer to the relative and absolute precision values as well as the non-interpolated mean average precision (MAP), which displays absolute precision as a function of absolute recall using official relevance assessments provided by INEX. Furthermore, the following, more sophisticated and XML-specific metrics were newly introduced for the INEX-IEEE benchmark: nxCG: The normalized extended Cumulated Gain (CG) metrics are an extension of the Cumulated Gain metrics that consider the dependency of XML elements (e.g., overlap and near-misses) within an evaluation.

As for INEX-Wikipedia effectiveness [13], we refer to the main ranking of INEX competition based on iP[0.01] instead of the overall measure MAiP, allowing us to emphasize precision at low recall levels. Our experiment targets Content Only (CO) Task only as well as systems that accept CO queries. Note that CO queries are terms enclosed in the <title> tag. Then, only the Focused Task remains in the INEX during the period 2005-2008. Thus, the system is evaluated only using Focused Task according to the *inex-eval* and $EvaJ$ tools provided by INEX.

C. Experiment Results and Discussion

In this section, we tuned parameters using INEX-2005 Adhoc track evaluation scripts distributed by the INEX organizers. Our tuning approach was such that the sums of all relevance scores are maximized. The total number of leaf node is 2500 and the β parameter is set to 0.10 [10], which is used to compute the sharing score function.

In order to evaluate the sensitivity of the evaluation, we have used the entire MySQL features are including Natural Language Searches (NLS), Boolean Searches (BLS), Query Expansion (QE), and Natural Language Searches with Query Expansion (NLSQE). As such, we report the effectiveness of our system on INEX collections as follows:

The performance of different feature and ranking models is evaluated. Tables I, II, III and Fig. 3, 4, 5 on INEX-Wikipedia collection and Tables IV, V and Fig. 6, 7 show the comparison of effectiveness to GPX system more details are following:

The run NLS obtained the highest scores for INEX-Wikipedia on 2006 topics is 0.4518, 0.2124 for BLS, 0.3163 for QE, and 0.3389 for NLSQE at iP[0.01] respectively. The

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

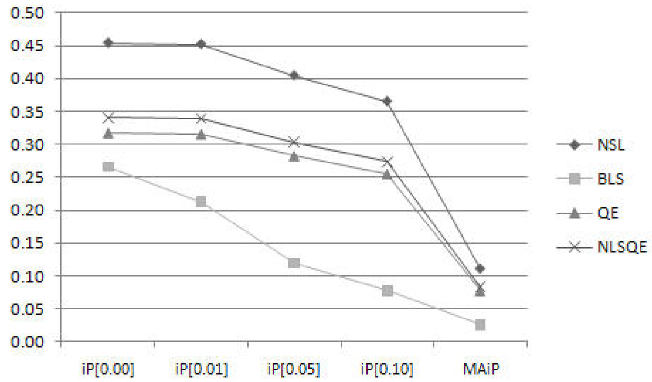

Fig. 3. The Effectiveness on INEX-2006 Focused Task

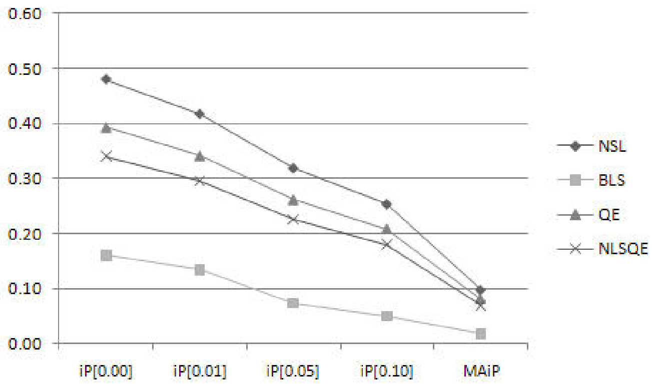

Fig. 4. The Effectiveness on INEX-2007 Focused Task

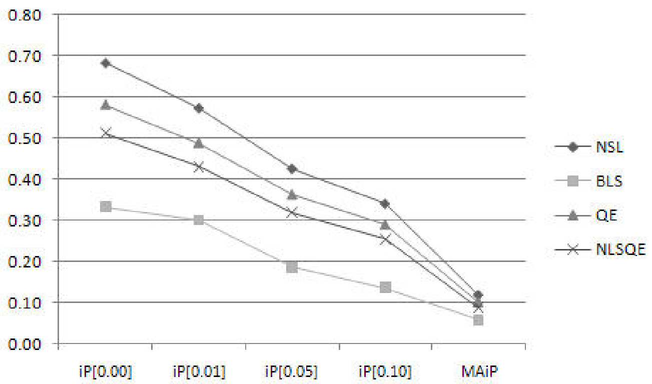

Fig. 5. The Effectiveness on INEX-2008 Focused Task

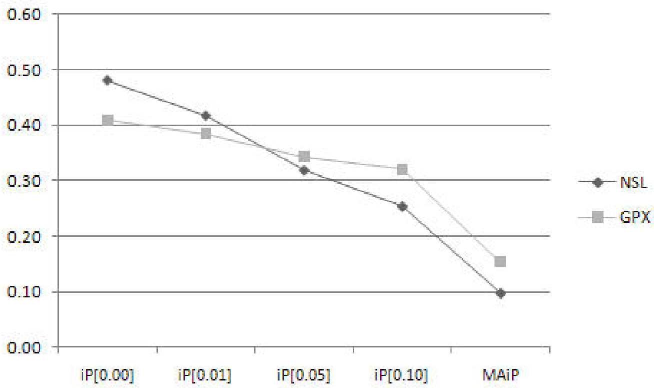

Fig. 6. Compare To GPX on The Effectiveness on INEX-2007

TABLE I
THE EFFECTIVENESS ON INEX-2006 FOCUSED TASK

RunID	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
NSL	0.4539	**0.4518**	0.4044	0.3651	0.1109
BLS	0.2657	**0.2124**	0.1193	0.0773	0.0257
QE	0.3177	**0.3163**	0.2831	0.2556	0.0776
NLSQE	0.3404	**0.3389**	0.3033	0.2738	0.0832

TABLE II
THE EFFECTIVENESS ON INEX-2007 FOCUSED TASK

RunID	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
NSL	0.4800	**0.4169**	0.3186	0.2539	0.0987
BLS	0.1605	**0.1349**	0.0733	0.0500	0.0179
QE	0.3936	**0.3419**	0.2613	0.2082	0.0809
NLSQE	0.3408	**0.2960**	0.2262	0.1803	0.0701

TABLE III
THE EFFECTIVENESS ON INEX-2008 FOCUSED TASK

RunID	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
NSL	0.6838	**0.5740**	0.4262	0.3411	0.1187
BLS	0.3338	**0.3008**	0.1859	0.1371	0.0584
QE	0.5812	**0.4879**	0.3623	0.2899	0.1009
NLSQE	0.5129	**0.4305**	0.3197	0.2558	0.0890

TABLE IV
COMPARE TO GPX IN THE INEX-2007 AD HOC TRACK

RunID	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
NLS	0.4800	**0.4169**	0.3186	0.2539	0.0987
GPX	0.4086	**0.3842**	0.3433	0.3208	0.1541

TABLE V
COMPARE TO GPX IN THE INEX-2008 AD HOC TRACK

RunID	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
NLS	0.6838	**0.5740**	0.4262	0.3411	0.1187
GPX	0.6818	**0.6344**	0.5693	0.5178	0.2587

run NLS obtained the highest scores for INEX-Wikipedia on 2007 topics is 0.4169, 0.1349 for BLS, 0.3419 for QE, and 0.2960 for NLSQE at iP[0.01] respectively. The run NLS obtained the highest scores for INEX-Wikipedia on 2008 topics is 0.5740, 0.3008 for BLS, 0.4879 for QE, and 0.4305 for NLSQE at iP[0.01] respectively.

IV. Conclusion

Due to the ever increasing information available electronically, their size is growing rapidly. The widespread use of XML documents in digital libraries led to the development of information retrieval (IR) methods specifically designed for XML collections. Most traditional IR systems are limited to whole document retrieval; however, since XML documents separate content and structure, XML-IR systems are able to retrieve the relevant portions of documents. Therefore, users who utilize an XML-IR system could potentially receive highly relevant and highly relevant and precise material.

In this paper, we have to investigate the weighting functions of MySQL and performed a comparative study of weighting schemes processing. Our objective of the study was to find out the appropriate features to achieve the effectiveness of XML element retrieval. Our experiment shows

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

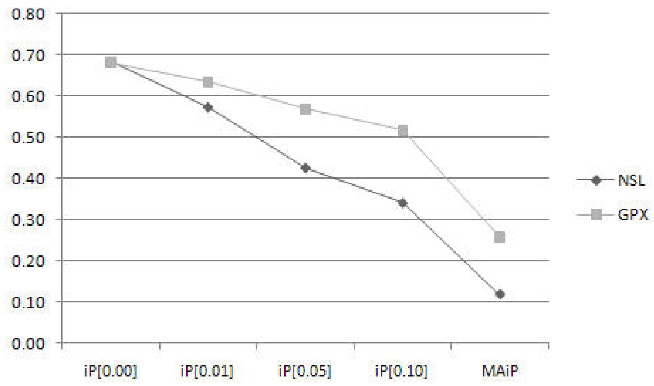

Fig. 7. Compare To GPX on The Effectiveness on INEX-2008

that the Natural Language search function used is the TF-IDF performs better than other methods measured by INEX evaluations on iP[0.01] and MAiP.

In our future work, we plan to study how to make inferences regarding structural aspects based on Content and Structure (CAS) queries.

REFERENCES

[1] Salton, G. and McGill, M.J., Introduction to Modern Information Retrieval. R. Donnelley and Sons Company, USA. (1983).
[2] Ricardo, B. and R. Berthier. Modern Information Retrieval, Addison Wesley Longman Publishing Co. Inc. (1999).
[3] Bray, T. et al, Markup Language (XML) 1.1 (Second Edition). Available Source: http://www.w3.org/TR/xml11/.
[4] Hinz, S. et al, MySQL Full-Text Search Functions, Available Source: http://dev.mysql.com/.
[5] Geva, S., GPX - Gardens Point XML Information Retrieval INEX 2004, Springer, Lecture Notes in Computer Science LNCS, pp. 211-223. (2005).
[6] Geva, S., GPX - Gardens Point XML-IR at INEX 2005, Springer, Lecture Notes in Computer Science LNCS, pp. 240-253. (2006).
[7] Geva, S., GPX - Gardens Point XML-IR at INEX 2006, Springer, Lecture Notes in Computer Science LNCS, ISBN 978-3-540-73887-9, pp. 137-150. (2007).
[8] Shin, D. et al, BUS: an effective indexing and retrieval scheme in structured documents, in: Proceedings of the third ACM conference on Digital libraries, pp. 235-243. (1998).
[9] Wichaiwong T. and Jaruskulchai C., XML Retrieval More Efficient Using ADXPI Indexing Scheme, The 4th International Symposium on Mining and Web, Biopolis, Singapore, (2011).
[10] Wichaiwong T. and Jaruskulchai C., A Simple Approach to Optimize XML Retrieval, The 6th International Conference on Next Generation Web Services Practices, Goa, India, November 23-25, (2010).
[11] Schenkel, R. et al, INitiative for the Evaluation of XML Retrieval (INEX). Available Source: https://inex.mmci.uni-saarland.de/. (Current 2011).
[12] Denoyer L. and Gallinari P., The Wikipedia XML Corpus. SIGIR Forum, pp. 64-69. (2006).
[13] Kamps, J. Pehcevski, J. Kazai, G. Lalmas, M. and Robertson, S., INEX 2007 evaluation measures. In 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007, Dagstuhl Castle, Germany, Selected Papers, (2008).

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)