

# A Scoring Method of XML Fragments Considering Query-Oriented Statistics

Atsushi Keyaki, Kenji Hatano

Faculty of Culture and Information Science,

Doshisha University

1-3 Tatara-Miyakodani, Kyotanabe,

Kyoto 610-0394, Japan

{keyaki, hatano}@ilab.doshisha.ac.jp

Jun Miyazaki

Graduate School of Information Science,

Nara Institute of Science and Technology

8916-5 Takayama, Ikoma,

Nara 630-0192, Japan

miyazaki@is.naist.jp

## Abstract

*In this paper, we propose a scoring method for searching XML fragments related with user's information need. In conventional XML fragment retrieval systems, the term-weighting schemes based on traditional information retrieval techniques are usually used directly for scoring retrieved XML fragments. However, we believe that retrieved XML fragments should be scored considering not only traditional retrieved-document-oriented statistics like the *tf-idf* scoring but also query-oriented ones such as constituent rate of query keywords and statistics of the query results, so that it remains possible that such techniques will help the conventional XML fragment retrieval systems to improve their relevances.*

*Our experimental results show that our method considering query conditions is more effective for searching XML fragments correctly than conventional approaches.*

## 1 Introduction

Extensible Markup Language (XML) [3] is becoming widely used as a standard document format in many application domains. In the near future, we believe that a greater number of documents will be produced in XML. Therefore, in a similar way to the development of Web search engines, XML search engines will become very important tools for users wishing to explore XML documents.

In the meantime, a search result of conventional XML search engines is usually a list of XML fragments<sup>1</sup>; that is, they are sorted in descending order of their scores like Web search engines. One of the differences between XML and Web search engines is the conditions of their queries. In the case of XML search engines, XML queries combine conditions on both content and logical structure such

<sup>1</sup>XML fragments are easily extracted from XML documents based on their markup. That is, they are sub-trees in the XML trees.

as Narrowed-Extended XPath I (NEXI) [13] and XQuery Full-Text queries [1], while Web queries are composed of only user's keywords in Web search engines. As a result, several approaches have been proposed to extend the well-established scoring in information retrieval techniques with the ability to score XML fragments.

Conventional approaches for XML search engines take statistics extracted from XML fragments into consideration to score them which satisfy query conditions [2]. The statistics can be extracted from XML fragments in a set of XML documents, so that we label them as "retrieved-document-oriented statistics." For example, in the context of the *tf-idf* scoring [12], element scoring pre-computes *tf* and *idf* factors for each distinct tag in input XML documents [5], while the path scoring pre-computes them for distinct paths [4, 8]. These scoring approaches led to improvements in the relevance of XML search engines; however, it is also an undeniable fact that their relevances themselves are still low for users. Therefore, we still have so much research work to deal with this problem.

Pondering on how to cope with the problem, we found that we overlook some important issues of extracted statistics. The statistics utilizing conventional scoring for XML fragments are invariant, because they are statically extracted from all XML fragments in a set of XML documents in advance. In contrast, the statistics extracted only from the XML fragments depending on each user's query can also be calculated. We label them *query-oriented statistics*. The query-oriented statistics also become important for scoring retrieved XML fragments. We believe that retrieved XML fragments should be scored using not only traditional retrieved-document-oriented statistics but also query-oriented ones. That is, both two types of statistics will improve their relevances.

In this paper, we propose a scoring method for searching XML fragments related with user's information need using both two types of statistics described above. We can consider that there are many kinds of query-oriented statistics; we use constituent rate of query keywords and occurrences

of terms in the query results as query-oriented statistics in this paper.

The remainder of this paper is organized as follows. In Section 2, we describe how to calculate the scores of XML fragments based on both retrieved-document-oriented and query-oriented statistics. In Section 3, we report our experimental results to verify our proposal and related work in Section 4. In the last section, finally, we conclude this paper and indicate some future works of this paper.

## 2 Scoring Method based on Query-Oriented Statistics

As we described in Section 1, our approach is a scoring method based on *the query-oriented statistics* to improve their relevances. That is to say, our approach does not directly utilize the statistics extracted from all XML fragments in a set of XML documents, but ones extracted only from the XML fragments depending on each user's query.

As a matter of fact, *the retrieved-document-oriented statistics* were focused in a research field of information retrieval in the past. However, *query-oriented statistics* were not so used in document search engines, because a user issues a query composed only of some keywords. In short, *the query-oriented statistics* were unhelpful for users.

However, we believe that *the query-oriented statistics* should be focused in XML search engines, because queries issued into XML search engines are composed of not only keywords, but also structural constraints, i.e., XPath expressions, of XML fragments that the user wants to retrieve. Issuing the structural constraints as a part of the query, we can consider many kinds of new statistics such as the number of XML fragments with the same abbreviated structural constraint containing query keywords, constituent rate of query keywords, and so on. We assert that such new statistics are also helpful for searching XML fragments relevantly, so that we have to propose a new scoring method in XML search engines by using these statistics.

Incidentally, our XML search engine first extracts all XML fragments whose root node is an intermediate node of XML documents, calculates term-scores based on the retrieved-document-oriented statistics. When a user issues a query, our XML search engine calculates query-keyword-scores based on the query-oriented statistics and scores of XML fragments related with the query. This is different from the approaches of conventional document retrieval systems, because our search engine has to calculate query-keyword-scores after issuing a user's query into the engine.

Our search engine can retrieve XML fragments related to user's query more relevantly than conventional ones because we consider both two types of statistics, though it takes a lot of time. The most important thing that we first have to solve is not an optimization problem but the relevance of retrieved XML fragments.

In order to employ our approach for taking the query-oriented statistics to relevantly retrieve XML fragment, we have to calculate term-scores and query-keyword-scores. In this section, we explain these scoring algorithms in detail.

### 2.1 Term-Score of XML Fragments

The term-score is one of the indicators of containing query keywords. In short, the more an XML fragment contains a lot of important query keywords, the larger its term-score becomes. In this sense, such an XML fragment can be captured as maximally informative one for a user.

Though this type of score has already been proposed in this research area, we chose the *tf-ipf* scoring algorithm [8] to calculate the term-scores. The *tf-ipf* scoring is one of the path-based scorings which expand the versatility of the *tf-idf* scoring [12]. XML fragments extracted from an original XML document are identified their structural constraints, so that they are classified according to the abbreviated syntax of their XPath expressions. Assuming that the XML fragments with the same abbreviated structural constraints have the same properties, we can quantify the importance of terms in XML fragments with the same properties as *tf-ipf* scores. That is to say, a *tf-ipf* score of a certain term in an XML fragment becomes large if the term appears in it many times and does not appear in others with the same abbreviated structural constraints at the same time.

Let  $T$  be the set of query keywords,  $s$  be an XML fragment in a search result,  $tf_f(s, t)$  be the number of occurrences of term  $t \in T$  in  $s$ , and  $ipf_f(s, t)$  be the natural logarithm of quotient of the number of XML fragments which have the same structural constraints as  $s$  and the number of such answer XML fragments containing term  $t$ . For example, for given the query `//article//sec[about(., t1 t2) ]`,  $tf_f(s, t_i)$  and  $ipf_f(s, t_i)$  ( $i = 1, 2$ ) are defined as follows:

$$tf_f(s, t_i) = \frac{n(s, t_i)}{l(s)}, \quad ipf_f(s, t_i) = 1 + \log \frac{M(s)}{m(s, t_i)} \quad (1)$$

where  $n(s, t_i)$  is the number of occurrences of  $t_i$  in  $s$ ,  $l(s)$  is the length of  $s$  (total number of terms in  $s$ ),  $M(s)$  is the number of XML fragments which satisfy  $s$ 's full structural constraints (in this case, `/article/body/sec`, `/article/body/list/sec`, and all others satisfied with structural constraints of the query), and  $m(s, t_i)$  is the number of such fragments containing  $t_i$ . Then, a term-score of the XML fragment containing query keyword  $t_i$  is defined as the following equation:

$$S_f(s, t_i) = tf_f(s, t_i) \cdot ipf_f(s, t_i) \quad (2)$$

### 2.2 Query-Keyword-Scores of XML Fragments

As was mentioned in the previous section, it is highly possible that XML fragments are appropriate for a search

```

<?xml version="1.0" encoding="UTF-8"?>
<article>
  <name id="3747">Bill Gates</name>
  <body>
    <p>
      <emph3>William Henry Gates III</emph3>
      (born October 28, 1955), commonly known
      as <emph3>Bill Gates</emph3>, is the co-
      founder, chairman and chief software
      architect of Microsoft Corporation, ...
    </p>
    ...
    <section>
      <title>Early life</title>
      <p>
        Gates was born in Seattle, Washington,
        to William H. Gates, Sr., a prominent
        lawyer, and Mary Maxwell Gates. Gates
        was born ...
      </p>
      <p>
        Gates, with an estimated I.Q. of 160,
        excelled in elementary school,
        particularly in mathematics and the
        sciences. His parents enrolled him ...
      </p>
      ...
    </section>
    ...
  </body>
</article>

```

**Figure 1. A Sample XML Document in Wikipedia Corpus**

result if the XML fragments have large term-scores. The term-scores solely rely on original XML documents and do not consider query conditions, because they are calculated by using the statistics extracted from all XML fragments in a set of XML documents in advance. Therefore, only using the retrieval-document-oriented statistics is unable to function to calculate similarities between the XML fragments and user's query exactly. This is why we assert that the query-keyword-scores calculated by using the query-oriented statistics are needed in XML fragment retrieval. We would like to give large scores to XML fragments which are closely-associated with query conditions, query keywords, and structural constraints in user's query. In this sense, such XML fragments will be more exhaustive for a user.

More concretely, let us consider the XML document given in Figure 1. This example is quoted from the Wikipedia Corpus [6]. If a NEXI query like `//article//p[about(., "Gates")]` is issued to this document, the XML fragments

- $s_1$ : `/article[1]/body[1]/p[1]`,
- $s_2$ : `/article[1]/body[1]/section[1]/p[1]`,
- $s_3$ : `/article[1]/body[1]/section[1]/p[2]`,

would be returned as a search result. In existing approaches, the term-scores of XML fragment  $s_1$  and  $s_{2,3}$  are different each other, because their abbreviated structural constraints are different from the perspective of both content and logical structure of original XML documents. From the standpoint of query condition, however, they should be identified because these XML fragments are satisfied with both content and logical structure of the query. In short, we would like to give the same scores to the XML fragments satisfied with all conditions of the query. Therefore, we can account for this by considering both retrieved-document-oriented and query-oriented statistics and defining the query-keyword-scores as a function of those conditions as well as the term-scores. This idea might be basically the same in traditional document search [12]; however, it is an initial proposal in the research field of XML search engines. We believe that it helps to improve the relevances of our XML search engine.

Now, we define the query-keyword-score as  $S_q(t_i)$ . In the same manner as  $S_f(s, t_i)$ , given the query `//article//sec[about(., t1 t2)]`,  $tf_q(t_i)$  and  $iaf_q(t_i)^2$  are defined as follows:

$$tf_q(t_i) = w(t_i), \quad iaf_q(t_i) = 1 + \log \frac{V(p)}{v(p, t_i)} \quad (3)$$

where  $w(t_i)$  is the number of occurrences of  $t_i$  in the query keywords,  $V(p)$  is the number of XML fragments satisfying abbreviated structural constraints of  $p$  (in this case, `//article//sec`), and  $v(p, t_i)$  is the number of XML fragments satisfying the condition where  $p$  contains term  $t_i$ . In order to calculate  $iaf_q(t_i)$ , we also assume independence between paths in the query and combine  $iaf_q(t_i)$  for individual paths. Therefore, the query-keyword-score of XML fragment containing query keyword  $t_i$  is defined as the following equation:

$$S_q(t_i) = tf_q(t_i) \cdot iaf_q(t_i) \quad (4)$$

### 2.3 Combining Scores

We finally define the combination of  $S_f(s, t_i)$  and  $S_q(t_i)$ . In order to combine these two values, a mathematical function is usually used. There are many mathematical functions which can combine multiple values [11]; however, we try to use the function of their product which is

<sup>2</sup>The *iaf* is derived from the inverse answer frequency. That is, it is defined as the reciprocal of the number of XML fragments containing  $t_i$ .

the simplest one in the same spirit of traditional document retrieval systems.

Note that the XML fragment should also contain all kinds of query keywords in itself, so that the XML fragment containing all query keywords should be given a large term-score. Therefore, we have to take the indicator reflecting the number of query keywords in the XML fragment into the term-score as follows:

$$S'_f(s, t_i) = \frac{j(s)}{i} \cdot tf_f(s, t_i) \cdot ipf_f(s, t_i) \quad (5)$$

where  $i$  is the number of query keywords, and  $j(s)$  is the number of query keywords in  $s$ . The final scores of an XML fragment  $s$  associated with a query is thus defined using equation (4) and (5) as follows:

$$S(s) = \sum_{t_i \in T} S'_f(s, t_i) \cdot S_q(t_i) \quad (6)$$

## 2.4 Query Processing

We have experimentally developed an XML search engine using an RDBMS based on XRel [14]. With XRel, XML documents are typically divided into five relational tables: *document*, *path*, *element*, *token*, and *attribute*. Since our system is based on an RDBMS, an XPath-style NEXI query must be translated into an SQL before the execution. For the sake of simplicity, we only show an examples of a part of a query for calculating *tf-iaf* score  $S_q(t_i)$  in Figure 2.

The term weights using a variant of *tf-ipf* are pre-computed and stored in attribute *tod* of *token* table.  $tf_q(t_i)$  in equation (3) is calculated by sub-query (A) in Figure 2. On the other hand,  $iaf_q(t_i)$  is computed by sub-queries (B) and (C). Finally, *tf-iaf* score  $S_q(t_i)$  is obtained by calculating  $\text{SUM}(r0.score * (1 + \text{LN}(c0.count/c1.count)))$  in the first SELECT clause.

Seeing Figure 2, the translated SQLs tend to be complicated. If the number of XML documents increases, the performance becomes degraded because such a complicated SQL must deal with a large number of data. Therefore, we need to optimize to handle such a large number of data, even using a sophisticated RDBMS.

One good solution is to utilize the *materialized view*, which keeps pre-computed views in storage to avoid re-creating the same ones for every query. The materialized view is usually supported by RDBMSs and used as an optimization technique for OLAP applications which consists of many aggregation queries.

For example, translated SQL queries include sub-queries which are enclosed with dotted boxes like in Figure 2. The *tf-iaf* scores of relevant documents are calculated by the tables which are temporarily created by these sub-queries; however, they contain joins which are potentially high cost

```

-- NEXI: //article[about(., "genetic algorithm")]
SELECT d.filepath, r0.elementid, r0.dewey, SUM(r0.score * (1 +
LN(c0.count/c1.count))) AS rsv
FROM document d
SELECT e.docid, e.elementid, e.dewey, e.st, e.ed,
SUM(t.tod) AS score
FROM element e, path p, token t
WHERE p.pathexp LIKE '#%/article'
AND p.pathid = e.pathid
AND e.elementid = t.elementid
AND e.docid = t.docid
AND t.token IN ('algorithm', 'genetic')
GROUP BY e.docid, e.elementid, e.dewey, e.st, e.ed) r0, (
SELECT CAST(COUNT(*) AS numeric) AS count
FROM (
SELECT e.docid, e.elementid, e.dewey
FROM element e, path p
WHERE p.pathexp LIKE '#%/article'
AND p.pathid = e.pathid
GROUP BY e.docid, e.elementid, e.dewey) sc
) c0, (
SELECT CAST(COUNT(*) AS numeric) AS count
FROM (
SELECT e.docid, e.elementid, e.dewey
FROM element e, path p, token t
WHERE p.pathexp LIKE '#%/article'
AND p.pathid = e.pathid
AND e.docid = t.docid
AND e.elementid = t.elementid
AND t.token IN ('algorithm', 'genetic')
GROUP BY e.docid, e.elementid, e.dewey) sc
) c1
WHERE d.docid = r0.docid
GROUP BY d.filepath, r0.elementid, r0.dewey
ORDER BY rsv DESC;

```

Figure 2. A translated SQL query

operations. Fortunately, the structures of the sub-queries are commonly used in any queries. If the materialized view can be applied for the temporarily created tables generated by the sub-queries, instead of re-calculating sub-queries, the cost of queries reduces.

As in Figure 2, the sub-queries (A), (B), and (C) should be materialized. However, it has only to create two materialized views, (A) and (B), because the sub-query (C) is completely contained by view (A). The storage cost for storing materialized views affects the systems, though the size of each materialized view is not larger than that of base tables. However, the optimized queries using materialized views can improve the performance.

## 3 Experiments

In this section, we conducted experiments for the sake of the effectiveness of our proposal on our XML search engine. In order to conduct the experiments, we used the INEX 2008 test collection, which is the most famous test collection in the research area of XML fragment retrieval. The INEX 2008 test collection consists of the Wikipedia corpus, about 100 topics (queries), and their assessments. The number of the CO and CAS topics which were assessed by INEX participants are 31 and 39, respectively. The top-

ics have two types; one is content-only (CO), the other is content-and-structure (CAS). The former is based on the content condition, that is, only query keywords are provided. On the other hand, the latter is based on the content-and-structure, so that both query keywords and their abbreviated structural constraints are provided as user's query. In this paper, however, our proposal, query-oriented statistics, improves their relevances of CAS-type queries, so that we do not conduct the experiment for CO-type ones. Using this test collection, we can evaluate the effectiveness of XML search engines in perspective of their relevances.

Figure 3 shows an experimental result, recall-precision curves, using focused task<sup>3</sup> of the INEX 2008 test collection. This figure indicates that our approach based on the query-oriented statistics (marked as *qo* in this figure) is better than the conventional *tf-ipf* scoring (marked as *rdo* in the figure) in the relevances. Especially, precisions of our approach at recall level 0.0 to 0.4 are higher. This fact is also found by the table of mean average interpolated precision (MAiP) [9] in the figure — the relevances have been improved by over 10 percents. This is because our approach could provide large scores to the relevant XML fragments satisfied with user's query using the term- and query-keyword-scores.

Table 1 compares the proposed method with three other competitive systems. Only four systems including ours were evaluated both CAS and CO queries. *iP[X]* in the table indicates interpolated precision at recall level *X*. Our proposed method provides higher precision values when the recall level is less or equal to .01. On the other hand, our system is inferior to others when the recall level is over .01. However, we assert that our system can be regarded as a better one in practical use, because most of users look at only a small number of top ranked results.

We claim that not only the term-scores but also the query-keyword-scores can improve the ranking of the XML fragments which are maximally and exhaustively informative for users. This is because the CAS topics express users' information needs which are specified by query structural constraints. The scores of the resulting fine-grained XML fragments are refined and emphasized by the query-oriented statistics.

As described above, we can say that our newly-introduced statistics can improve the effectiveness of XML search engines in perspective of relevances.

## 4 Related Work

The application of information retrieval techniques in searching XML fragments has become an area of research

<sup>3</sup>In INEX definition, it is important for users to see as much relevant text as possible with as little irrelevant text as possible. This assumption is the basis of methods for evaluating the effectiveness of information retrieval systems based on recall and precision in the research field of information retrieval.

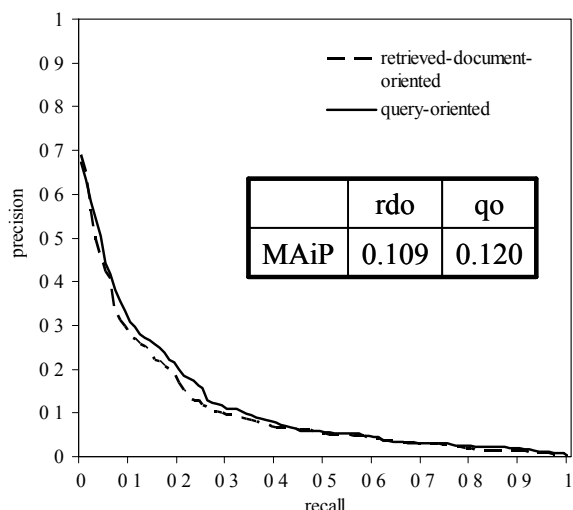


Figure 3. Recall-Precision (All CAS topics)

in recent years. Especially, the participants in the INEX project have proposed many types of scoring algorithms for XML search engines [2]. Over the years, it has become clear that the relevance has been increased by refining the level of granularity at which the document structure is taken into account in pre-computing individual term weights both in the vector space and the probabilistic models. However, statistics related to query conditions have not been explored to the extent that we are proposing in this paper.

Fuhr et al proposed a method for propagating scores of XML fragments leaf-to-root along the XML document tree [7]. Although XIRQL, their proposed language, enables queries with a mix of conditions on both structure and keywords, only keywords are scored using conditions on document structure. Other scoring methods also use conditions on document structure to apply length normalization between query paths and data paths [4], to compute term weights based on element tags or paths [5, 8]. It was reported that these methods were useful for searching XML fragments [10]; however, such methods did not consider the query-oriented statistics extracted from the XML fragments satisfied with user's query.

In this paper, therefore, we showed that both the retrieved-document-oriented and the query-oriented statistics, in addition to the existing methods, improve their relevances.

## 5 Conclusion

In this paper, we proposed an more relevant XML fragment retrieval method using the combination of retrieved-document-oriented statistics and query-oriented ones. Our

**Table 1. Comparisons with other systems in focused task**

| Team                           | iP[.00] | iP[.01] | iP[.05] | iP[.10] | MAiP   |
|--------------------------------|---------|---------|---------|---------|--------|
| Doshisha Univ.                 | 0.6717  | 0.6321  | 0.4417  | 0.3068  | 0.1201 |
| Renmin Univ. of China          | 0.5969  | 0.5969  | 0.5815  | 0.5439  | 0.2486 |
| Queensland Univ. of Technology | 0.6232  | 0.6220  | 0.5521  | 0.4617  | 0.2134 |
| Univ. of Amsterdam             | 0.6232  | 0.6220  | 0.5521  | 0.4617  | 0.2134 |

experimental evaluations showed that we could retrieve more relevant XML fragments using our approach than other competitive systems as well as conventional ones. In particular, the query-keyword-scores could make a significant contribution to improve their relevances, and the term-scores could also improve precisions, in particular, at lower recall level, which brings great advantage in practical use.

In the future, we would like to address the following issues:

- As we described Section 3, our approach cannot be applied to CO topics, because query-keyword-scores are calculated from the query-oriented statistics. Therefore, we have to study the meanings of “informative XML fragments” in the research field of XML fragment retrieval, and have to propose an approach to improve their relevances for CO topics.

We are currently formalize to define fine-grained XML fragments for CO topics, and are going to propose a novel information retrieval model for XML fragment retrieval.

- Our new scoring algorithm is suitable for searching relevant XML fragments; however, it takes a long time to retrieve a search result compared with a conventional Web search engines when user issues a twig query. Therefore, we have to develop an efficient query processing mechanism for it in our XML search engine.
- In order to develop an XML search engine, it is also important for users to propose not only scoring algorithm but also a user interface for issuing XML queries and browsing search results. This is because the user interface helps users to understand search results efficiently. We are currently developing such a user interface on top of our XML search engine.

## References

- [1] S. Amer-Yahia, C. Botev, J. Dorre, and J. Shanmugasundaram. XQuery Full-Text extensions explained. *IBM Systems Journal*, 45(2):335–352, Dec. 2006.
- [2] S. Amer-Yahia and M. Lalmas. XML Search: Languages, INEX and Scoring. *SIGMOD Record*, 35(4):16–23, Dec. 2006.
- [3] T. Bray, J. Paoli, M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible Markup Language (XML) 1.0 (Fifth Edition). <http://www.w3.org/TR/xml>, Sep. 2006. W3C Recommendation 26 November 2008.
- [4] D. Carmel, Y. S. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer. Searching XML Documents via XML Fragments. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 151–158, Jul./Aug. 2003.
- [5] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv. XSearch: A Semantic Search Engine for XML. In *Proceedings of 29th International Conference on Very Large Data Bases*, pages 45–56, Sep. 2003.
- [6] L. Denoyer and P. Gallinari. The Wikipaedia XML Corpus. *SIGIR Forum*, 40(1):64–69, June 2006.
- [7] N. Fuhr and K. Großjohann. XIRQL: An XML Query Language based on Information Retrieval Concepts. *ACM Transactions on Information Systems*, 22(2):313–356, Apr. 2004.
- [8] T. Grabs and H.-J. Schek. PowerDB-XML: A Platform for Data-Centric and Document-Centric XML Processing. In *Proceedings of the First International XML Database Symposium*, volume 2824 of *Lecture Notes on Computer Science*, pages 100–117. Springer, Sep. 2003.
- [9] J. Kamps, J. Pehcevski, G. Kazai, M. Lalmas, and S. Robertson. INEX 2007 Evaluation Measures. In *Focused Access to XML Documents*, volume 4862 of *Lecture Notes in Computer Science*, pages 24–33. Springer-Verlag, August 2008.
- [10] G. Kazai and M. Lalmas. INEX 2005 Evaluation Metrics. In *Advances in XML Information Retrieval and Evaluation*, volume 3977 of *Lecture Notes on Computer Science*, pages 16–29. Springer-Verlag, Jun. 2006.
- [11] M. Montague and J. A. Aslam. Condorcet fusion for improved retrieval. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages pp.538–548. ACM, November 2002.
- [12] G. Salton and C. Buckley. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513–523, 1988.
- [13] A. Trotman and B. Sigurbjörnsson. Narrowed Extended XPath I (NEXI). In *Advances in XML Information Retrieval*, volume 3493 of *Lecture Notes in Computer Science*, pages 16–40. Springer-Verlag, May 2005.
- [14] M. Yoshikawa, T. Amagasa, T. Shimura, and S. Uemura. XRel: A Path-based Approach to Storage and Retrieval of XML Documents using Relational Databases. *ACM Transactions on Internet Technology*, 1(1):110–141, Aug. 2001.