XML Fuzzy Ranking

Evangelos Kotsakis

Joint Research Center (CCR),TP267
Via Fermi 1, 21020 Ispra (VA), Italy
evangelos.kotsakis@jrc.it
http://www.jrc.it/

Abstract. This paper proposes a method of ranking XML documents with respect to an Information Retrieval query by means of fuzzy logic. The proposed method allows imprecise queries to be evaluated against an XML document collection and it provides a model of ranking XML documents. In addition the proposed method enables sophisticated ranking of documents by employing proximity measures and the concept of editing (Levenshtein) distance between terms or XML paths.

1 Introduction

Information Retrieval (IR) techniques have traditionally been applied to search large sets of textual data. The emerge of XML as a standard for data representation and exchange on the Internet poses new challenges to structured document retrieval. Integrating IR and XML search techniques could enable more sophisticated search on the structure as well as the content of the documents. Some of the recent XML-IR proposals [6, 8, 11, 4, 2] focus on indexing for improving the execution of simple IR-like queries. Other approaches [9, 15, 25] are based on soft computing techniques to model uncertainty. This paper is closer to the second approach and it uses fuzzy logic to rank the documents of an XML collection with respect an IR query. Ranking of structured documents is very important and may further boost the quality of the results. Ranking structured documents is mainly based on probabilistic models [24, 21, 11, 26] by mainly using the *idf* (inverse document frequency) and *tf*, within document frequency of terms.

Searching effectively large collections of XML documents requires a knowledge of the documents structure. For instance, using XQuery [5], it requires some knowledge on the XML schema (or DTD) [1, 3]. On the other hand searching XML documents from the Information Retrieval (IR) point of view requires little knowledge on the structure of the documents. Although knowledge on the XML schema could be desirable to reduce extensive search, it is not required. Another important aspect in retrieving XML documents is the relevance of the retrieved documents to the submitted query. In most cases, we are not interested in finding XML documents that exactly match the query but rather in XML documents, which are relevant to the query to some extent. Our tolerance to accept in the result set XML documents that do not precisely match the query is stemmed from the fact that the query itself is not precise. IR based queries cannot be precise mainly due to the lack of knowledge of the underlying document collection structure and the inherent difficulty to formulate a query that precisely reflects what we are looking for. On the other hand, the relevance is by nature a *fuzzy concept*. So it seems that the employment of fuzzy logic for estimating the relevance of the documents to a given query is more expressive and promising.

Modeling vagueness in information retrieval has been addressed by several researchers [14, 15, 18, 25, 9] and recently in the area of XML documents through the INEX workshops [28, 7]. Weighted boolean models [14] have been proposed to handle constraints imposed by the query terms. Fuzzy set theory has been proposed [15, 16, 18, 25] for modeling flexible information retrieval systems. Models for integrating IR and database systems is presented in [12]. In the area of structured documents [25] fuzzy aggregations have been employed to facilitate retrieval. A survey on information retrieval techniques based on soft computing (mainly fuzzy sets and neural networks) is presented in [18]. In [9] the logical structure among objects have been employed to model structured documents in which the logical structure among objects is captured by means of knowledge augmentation. While in classical retrieval the quality is estimated by means of *idf* (inverse document frequency) and *tf* (within document term frequency), [27] proposes a third dimension called *accessibility* that captures the structure of the document. Vague queries and inexact data matching in databases is proposed in [22] and fuzzy Hamming distance that extends the Hamming concept for measuring dissimilarity between vector objects is proposed in [17].

This paper proposes a fuzzy logic based technique for ranking XML documents against a given query. The employment of fuzzy logic enables more sophisticated ranking by exploiting the imprecise formulation of the queries. To achieve this, a database is used for indexing purposes and for storing all the *facts* that constitute the XML document collection. Fuzzy measurements based on editing distance are used to handle uncertainty and imprecision. The rest of the paper is organized as follows: Section 2 discusses the IR query format. Fuzzy ranking is discussed in section 3. The realization of the database structure used to facilitate the ranking is presented in section 4. The fuzzy relevance of a document against to a subquery is discussed in the section 5. Some implementation notes are presented in the section 6 and the section 7 summarizes the contributions and concludes the paper.

2 Query Format

XML Querying mechanisms that facilitate Information Retrieval (IR) should allow the formulation of simple queries that require no knowledge of the underlying structure of the XML document collection. However, the query method should be also flexible enough to allow more complex queries that take into consideration the structure of the underlying documents. In addition the query mechanism should also allow the user to focus the search on particular XML elements that might be more interesting. So, element prioritization that reflects the importance of the element in context and in relation with other elements is equally important.

The proposed IR query model is relatively simple and it has been derived from Xpath [13] by adopting only the basic expressions for addressing parts of XML documents. Only few of the Xpath location paths and core function are supported. The objective is to keep the querying mechanism as simple as possible.

A query Q consists of one or more sub-queries q_i, each one being a simplified Xpath expression. The sub-queries can be connected with each other using ordinary conjunction and disjunction operators. Each sub-query is associated with a weight, which is called *relative importance*. The relative importance R is used as a tool that allows the user to express its preferences on particular sub-queries. This is a user defined quantity. Let n be the number of sub-queries that forms the query Q, then R and Q could be expressed as vectors.

$$Q = (q_1, q_2, \ldots, q_n)$$

$$R = (r_1, r_2, \ldots, r_n)$$

The relative importance R is defined such as

$$\sum_{i=0}^{n} r_i = 1$$

Each sub-query consists of two main components: the first one is for expressing the path and the second one for expressing a literal term found at the end of the path. For example the sub-query `/article/*/name[Kevin]` consists of the path expression `/article/*/name` and the literal term “Kevin”, which is in fact part of the atomic value of the element “name”. Special symbols such as * (matching any element) and // (descendent axis) could also be used to express a path.

Example 1. Consider the following XML file:

```
<article>
    <title> Knowledge-Based Automation Software</title>
    <author>
        <name> Kevin Smith</name>
        <affiliation> Member of IEEE</affiliation>
    </author>
    <abstract>
        This paper describes a knowledge-based approach to automate
        a software design method for concurrent systems. Production
        rules provide the mechanism for codifying a set of heuristics.
    </abstract>
</article>
```

A typical IR query and its associated “relative importance” vector could be formed as follows:

$$Q = \begin{pmatrix} q_1: & //author/*[smith]\wedge \\ q_2: & //abstract[knowledge]\wedge \\ q_3: & /article/title[knowledge] \end{pmatrix}$$

$$R = (0.4, 0.2, 0.4)$$

The query Q above consists of three sub-queries q_1, q_2 and q_3 whose relative importance are $0.4, 0.2$ and 0.4 respectively. The term *"knowledge"* found in the *"title"* element would yield a score twice as big as if it was found in the *"abstract"* element. This shows that the user is interested more about the term *knowledge* found in *title* rather in *abstract.* Using the relative importance vector, the user can focus the search on particular paths of the collection and retrieve those XML documents that are more relevant according to the user preferences.

3 Fuzzy Ranking

Suppose $D = (d_1, d_2, \ldots, d_m)$ is the vector representing the XML documents of the collection and $Q = (q_1, q_2, \ldots, q_n)$ is the vector representing the query. The problem is to find those XML documents that best satisfy the query or in other words they are relevant to the query. The relevance of the document d_i with respect to the sub-query q_j is measured by a rating, which is called *Degree of Satisfaction* DS_{ij}. The degree of satisfaction DS_i of the document d_i is defined by the following vector:

$$DS_i = (DS_{i1}, DS_{i2}, \ldots, DS_{in}) \tag{1}$$

Let $R = (r_1, r_2, \ldots, r_n)$ be the relative importance vector, then the relative degree of satisfaction DS_i^r of the document d_i is defined as

$$DS_i^r = (r_1 \cdot DS_{i1}, r_2 \cdot DS_{i2}, \ldots, r_n \cdot DS_{in}) \tag{2}$$

Which is in fact comprised of the performance of the document d_i on the n sub-queries taking into consideration the relative importance of the sub-query. Let $DS_{ij}^r = r_j \cdot DS_{ij}$ then the matrix in equation 3 gives the relative degree of satisfaction DS^r for the whole XML collection with respect to the query Q and the relative importance vector R.

$$DS^r = \begin{bmatrix} DS_{11}^r & DS_{12}^r & \ldots & DS_{1j}^r & \ldots & DS_{1n}^r \\ DS_{21}^r & DS_{22}^r & \ldots & DS_{2j}^r & \ldots & DS_{2n}^r \\ \vdots & \vdots & \ldots & \vdots & \ldots & \vdots \\ DS_{i1}^r & DS_{i2}^r & \ldots & DS_{ij}^r & \ldots & DS_{in}^r \\ \vdots & \vdots & \ldots & \vdots & \ldots & \vdots \\ DS_{m1}^r & DS_{m2}^r & \ldots & DS_{mj}^r & \ldots & DS_{mn}^r \end{bmatrix} \tag{3}$$

DS_{ij}^r is a real number within the interval $[0, 1]$ and represents the satisfaction of the document d_i with respect to the sub-query q_j. The estimation of DS_{ij} is discussed in section 5. This section explains how to rank the documents employing fuzzy logic by using the matrix in equation 3.

Suppose that the sub-queries $q_j, \forall j \in \{1 \ldots n\}$ are connected with each other with disjunction, conjunction and negation operators. Each sub-query q_j may be seen as a fuzzy condition. In practice the definitions of the operators AND, OR and NOT may vary, but one popular definition is shown in Table 1.

Table 1. Fuzzy operators

Operator	Symbol	Definition
AND	$\wedge$	minimum value
OR	$\vee$	maximum value
NOT	$\neg$	negation (complement to 1)

Let w_i be the ranking weight of the document d_i with respect to the query Q, then w_i is given by the following tuple:

$$w_i = [DS_i^r, (Q, \vee, \wedge, \neg)] \tag{4}$$

Where DS_i^r is the document d_i relative satisfaction vector given by the equation 2 and $(Q, \vee, \wedge, \neg)$ is the query vector with the fuzzy connectors. The following example show how the value of w_i is calculated.

Example 2. Suppose we have three XML documents d_1, d_2 and d_3 and a query Q with three sub-queries q_1, q_2 and q_3, which are connected as $Q = q_1 \wedge (q_2 \vee q_3)$. Suppose also that the relative importance vector is $R = (0.31, 0.27, 0.42)$ and the degree of satisfaction DS matrix is given by

$$DS = \begin{bmatrix} 0.2 & 0.3 & 0.1 \\ 0.4 & 0.5 & 0.35 \\ 0.6 & 0.15 & 0.25 \end{bmatrix}$$

From the above, the relative degree of satisfaction DS^r is given by

$$DS^r = \begin{bmatrix} 0.31 \cdot 0.2 & 0.27 \cdot 0.3 & 0.42 \cdot 0.1 \\ 0.31 \cdot 0.4 & 0.27 \cdot 0.5 & 0.42 \cdot 0.35 \\ 0.31 \cdot 0.6 & 0.27 \cdot 0.15 & 0.42 \cdot 0.25 \end{bmatrix} = \begin{bmatrix} 0.0868 & 0.081 & 0.042 \\ 0.1736 & 0.135 & 0.147 \\ 0.2604 & 0.0405 & 0.105 \end{bmatrix}$$

By taking into account the definition in table 1, the ranking weights w_1, w_2 and w_3 of the documents d_1, d_2 and d_3 respectively are estimated as follows:

$$W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} min(0.0868, max(0.081, 0.042)) \\ min(0.1736, max(0.135, 0.147)) \\ min(0.2604, max(0.0405, 0.105)) \end{bmatrix} = \begin{bmatrix} 0.081 \\ 0.147 \\ 0.105 \end{bmatrix}$$

Therefore, the highest ranked document to the query Q is document d_2, then it comes d_3 and d_1.

4 Database Structure

Before discussing the estimation of the Degree of Satisfaction DS_{ij}, it helps to understand how the XML documents have been stored in a relational database and how the XML facts (terms, paths, documents etc.) have been structured and the posting information is used to derive weights for both path expressions and literal terms.

The following entities are used to index an XML document collection:

$$\begin{aligned} paths\left(\underline{path\_id}, path\_name\right) \\ terms\left(\underline{term\_id}, term\_name, IDF\right) \\ docs\left(\underline{doc\_id}, doc\_name\right) \end{aligned}$$

A *path_name* is a string of tags separated by slash (i.e. /article/author/name). The *"paths"* entity contains only those paths whose leafs are simple elements, that is, at the end of the path there is always a simple content elements (i.e. #PCDATA)

The *"terms"* entity contains every single term (excluding the stop words), which is not a tag, that is, it appears within the simple content of an element. An associated IDF (Inverse Document Frequency) value is also associated with each term of the XML document collection.

The *"docs"* entity contains information about each XML document of the collection. Relationships between the above three entities have been also defined in a way that reflects the structure of the XML document of the collection.

The above structure has been realized in a relational database. Figure 1 shows the schema of the database.

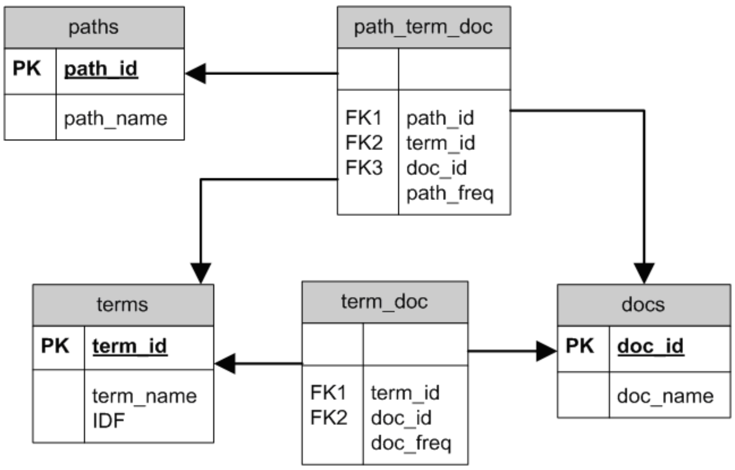

Fig. 1. Physical implementation of the database structure used to store an XML document collection

path_freq is the frequency of the term in the path and *doc_freq* is the frequency of the term in the XML document.

5 Degree of Satisfaction (DS)

The degree of satisfaction DS_{ij} represents the relevance of the document d_i with respect to the sub-query q_j. The sub-query q_j consists of two parts; the first one is the path expression $path(q_j)$ and the second one is the literal term expression $term(q_j)$ at the end of the path. The relevance of the document d_i to the sub-query q_j depends on the degree that q_j matches a path-term pair of d_i.

Suppose that the document d_i has k paths, $p_1, p_2, \ldots p_k$ and l terms, $t_1, t_2, \ldots, t_l$. By calculating the Levenshtein (or editing) distance between the $path(q_j)$ and each p_s for $s = 1 \ldots k$, it yields a fuzzy number δ_{ij}

$$\delta_{ij} = \{\frac{v_1}{p_1}, \frac{v_2}{p_2}, \ldots, \frac{v_k}{p_k}\}$$

The Levenshtein distance between paths is calculated by using the path tags as the editing granule. That is, */article/author* and */article/author/name* have Levenshtein distance equal to 1; it needs to delete one tag (i.e. "name") of the second to get the former. In case path expressions contain the descendant axis ("//") operator, the estimated distance is the minimum Levenshtein distance taken from all possible matches. For instance, in the XML document in the Example 1, the paths *//abstract* and *//name* match the paths */article/abstract* and */article/author/name* respectively whose distance is 3 (1 deletion and 2 insertions; substitution cost is double the insertion/deletion cost) and thus the distance between *//abstract* and *//name* is 3. In the case we had more than one matches, the minimum distance of all possible matches determines the distance between the paths. The value v_x for $x = 1 \ldots k$ is the membership value and it is given by

$$v_x = e^{-x} \tag{5}$$

Where x is the Levenshtein distance between the $path(q_j)$ and p_x. The exponential function in equation 5 yields 1 (perfect match) when the Levenshtein distance $x = 0$. On the other hand when the distance gets larger, it tends to be zero. Figure 2 shows a graphical representation of the exponential membership function based on the editing distance.

The concept of editing distance is again applied to the comparison of the terms $t_1, t_2, \ldots, t_l$ of the document d_i with $term(q_j)$, but in this case the editing granule is a single character rather a tag. The comparison yields a fuzzy number μ_{ij}, which is given by

$$\mu_{ij} = \{\frac{u_1}{t_1}, \frac{u_2}{t_2}, \ldots, \frac{u_l}{t_l}\}$$

Again u_x is given by an exponential function in equation 5 where x is the editing distance between the term t_x and $term(q_j)$.

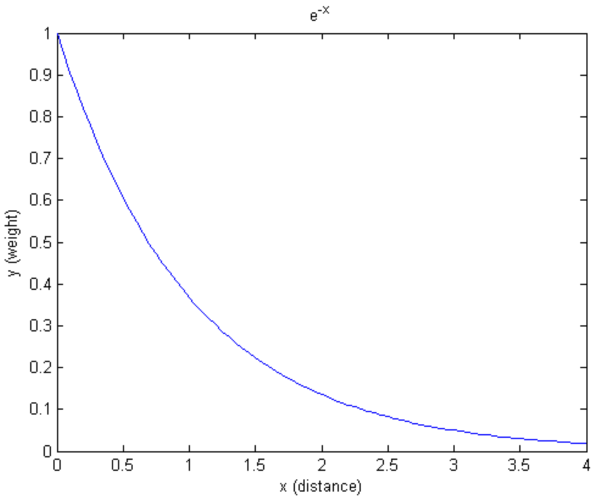

Fig. 2. $y = e^{-x}$ is the the fuzzy membership function representing the degree of *fuzzy* matching between two terms or paths given their Levenshtein distance x

Let M_i be the binary matrix i.e., a matrix each of whose elements is 0 or 1, that represents the binding of terms and paths in the document d_i. An element m_{xy} of the binary matrix is 1 when the path p_x contains the literal term t_y.

The degree of satisfaction DS_{ij} of the document d_i with respect to the sub-query q_j is given as

$$DS_{ij} = max\{m_{xy}min(v_x, u_y), \forall x, y \in d_i\} \tag{6}$$

Where $x = 1 \ldots k$ represents the paths in d_i and $y = 1 \ldots l$ represents the terms in d_i and m_{xy} is the corresponding entry of the binary matrix M_i. The above definition guarantees that if the sub-query q_j matches exactly a path-term pair of the document d_i the DS_{ij} is 1, otherwise if the matching is inexact, it will be a number within the interval $[0, 1]$.

Example 3. Suppose a document d has the following paths $p_1 = /article/author/name$, $p_2 = /article/title$ and the terms $t_1 = santos$, $t_2 = know$ and $t_3 = information$. A sub-query q could be $/article/author[santo]$ with $path(q) = /article/author$ and $term(q) = santo$. Let's assume that the binary matrix M is as follows:

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

The matrix above shows that the term t_1 is under the path p_1, while the terms t_2 and t_3 are under the path p_2. Suppose that the cost of the Levenshtein editing operations are 1 for deletion and insertion and 2 for substitution, then δ and μ are as follows:

$$\delta = \{\frac{e^{-1}}{p_1}, \frac{e^{-2}}{p_2}\}$$

$$\mu = \{\frac{e^{-1}}{t_1}, \frac{e^{-5}}{t_2}, \frac{e^{-10}}{t_3}\}$$

The degree of satisfaction DS of the document d with respect to the sub-query q is given by

$$DS = max\{1 \cdot min(e^{-1}, e^{-1}), 1 \cdot min(e^{-2}, e^{-5}), 1 \cdot min(e^{-2}, e^{-10})\} = e^{-1}.$$

6 System Implementation

A prototype of the system has been developed using mainly open source software components. The database has been realized using the MySQL DBMS. All tools for processing the XML documents have been implemented in Perl [19]. For parsing the XML documents a module that is based on *Expat* [20] has been used. For manipulating the XML elements, the LibXML module has been utilized. LibXML supports a major standard for XML processing known as the Document Object Model (DOM) [23]. The web interface for submitting the queries has been realized using PHP [10]. For testing the system, the INEX 2002 [28, 7] XML document collection has been used.

7 Conclusions

This paper proposes a method of ranking XML documents, which is based on fuzzy logic. The advantages of the proposed method are the following:

– Allows the formulation of imprecise queries
– The use of fuzzy logic provides a quantitative tool for measuring the inexact matching between paths or other XML terms
– Easy to implement it
– The editing distance between paths provides an effective way to quantify the relevance of the paths according to their similarity against the query terms

Employing fuzzy logic seems to be a natural way to handle imprecision in evaluating XML-IR based queries. The combination of fuzzy measurements with probabilistic measurements may further improve the ranking quality.

References

1. Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler and François Yergeau (eds): Extensible Markup Language (XML) 1.0 (Third Edition). W3C Recommendation, 04 February 2004. http://www.w3.org/TR/REC-xml/
2. Evangelos Kotsakis: XSD: A Hierarchical Access Method for Indexing XML Schemata. Knowledge and Information Systems **4**(2):168–201 (2002), Springer-Verlag.

3. David C. Fallside and Priscilla Walmsley (eds): XML Schema Part 0: Primer Second Edition. W3C Recommendation, 28 October 2004. http://www.w3.org/TR/xmlschema-0/
4. Holger Meyer, Ilvio Bruder, Andreas Heuer, Gunnar Weber: The Xircus Search Engine. Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX), Schloss Dagstuhl, Germany, December 9-11, 2002, pp. 119–124
5. Scott Boag, Don Chamberlin, Mary F. Fernández, Daniela Florescu, Jonathan Robie and Jérôme Siméon (eds): XQuery 1.0: An XML Query Language. W3C Candidate Recommendation, 3 November 2005. http://www.w3.org/TR/xquery/
6. Evangelos Kotsakis. Structured Information Retrieval in XML documents. In Proceedings of the seventeenth ACM Symposium on Applied Computing (SAC 2002), pp. 663–667, Madrid, Spain, March 10-14, 2002.
7. Initiative for the Evaluation of XML retrieval, INEX 2002, DELOS Network of Excellence for Digital Libraries, http://qmir.dcs.qmul.ac.uk/inex/index.html
8. Carlo Combi, Barbara Oliboni and Rosalba Rossato: Querying XML Documents by Using Association Rules, 16th International Workshop on Database and Expert Systems Applications (DEXA'05) (2005) pp. 1020–1024, Copenhagen, Denmark.
9. Mounia Lalmas, Thomas Rölleke: Four-Valued Knowledge Augmentation for Structured Document Retrieval. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **11**(1): 67-86 (2003)
10. PHP Documentation Group: PHP-Hypertext Preprocessor, http://www.php.net/
11. Felix Weigel, Holger Meuss, Klaus U. Schulz and Francois Bry: Content and Structure in Indexing and Ranking XML, Proceedings of the 7th International Workshop on the Web and Databases (WebDB)(2004), Paris, France.
12. Norbert Fuhr: Models for Integrated Information Retrieval and Database Systems, IEEE Bulletin of the Technical Committe on Data Enginnering, Vol. 19 No. 1, March 1996 pp 3–13
13. James Clark and Steve DeRose (eds): XML Path Language (XPath) Version 1.0. W3C Recommendation, 16 November 1999. http://www.w3.org/TR/xpath
14. G. Pasi: A logical formulation of the Boolean model and of weighted Boolean models, Workshop on Logical and Uncertainty Models for Information Systems (LUMIS 99), University College London, 5-6 July 1999.
15. G. Bordogna, G. Pasi: Modeling Vagueness in Information Retrieval, in LNCS-1980 M. Agosti, F. Crestani and G. Pasi eds, "Lectures on Information Retrieval: Third European Summer-School, ESSIR 2000, Varenna, Italy, September 11-15, 2000, Springer Verlag, 2001.
16. Ronald R. Yager, Henrik Legind Larsen: Retrieving Information by Fuzzification of Queries. J. Intell. Inf. Syst. **2**(4): 421-441 (1993)
17. Abraham Bookstein, Shmuel Tomi Klein and Timo Raita: Fuzzy Hamming Distance: A New Dissimilarity Measure, In Lecture Notes in Computer Science Volume 2089 / 2001 A. Amir, G.M. Landau (Eds.): Combinatorial Pattern Matching: 12th Annual Symposium, CPM 2001 Jerusalem, Israel, July 1-4, 2001.
18. F. Crestani and G. Pasi: Soft Information Retrieval: Applications of Fuzzy Set Theory and Neural Networks, in Neuro-fuzzy tools and techniques, N.Kasabov Editor, Physica-Verlag , Springer-Verlag Group, 1999, pp. 287-313.
19. Larry Wall, Tom Christiansen and Jon Orwant: Programming Perl, 3rd Edition, July 2000, O'Reilly, ISBN: 0-596-00027-8
20. The Expat XML Parser, by James Clark http://expat.sourceforge.net/

21. Jens E. Wolff, Holger Flörke and Armin B. Cremers: XPRES: a Ranking Approach to Retrieval on Structured Documents. Technical Report JAI-TR-99-12, Institute of Computer Science III (1999)
22. Raquel Kolitski Stasiu, Carlos A. Heuser and Roberto da Silva: Estimating Recall and Precision for Vague Queries in Databases, In Lecture Notes in Computer Science Volume 3520, Oscar Pastor, Joo Falco e Cunha (eds: Advanced Information Systems Engineering: 17th International Conference, CAiSE 2005, Porto, Portugal, June 13-17, 2005, Springer Berlin
23. Document Object Model (DOM), http://www.w3.org/DOM/
24. Norbert Fuhr: A Probabilistic Framework for Vague Queries and Imprecise Information in Databases. Proceedings of VLDB 90, pp. 696-707, 1990.
25. Gabriella Kazai, Mounia Lalmas, Thomas Rölleke: A Model for the Representation and Focussed Retrieval of Structured Documents Based on Fuzzy Aggregation, String Processing and Information Retrieval (SPIRE) 2001: 123-135
26. Katsuya Masuda: A Ranking model of proximal and structural text retrieval based on region algebra. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL'03)- Volume 2, Pages: 50 – 57, Sapporo, Japan (2003)
27. Thomas Rölleke, Mounia Lalmas, Gabriella Kazai, Ian Ruthven, and Stefan Quicker: The Accessibility Dimension for Structured Document Retrieval, Lecture Notes in Computer Science Volume 2291/2002,Springer Berlin
28. Gabriella Kazai, Norbert Gövert, Mounia Lalmas, Norbert Fuhr: The INEX Evaluation Initiative, Lecture Notes in Computer Science Volume 2818/2003, Pages: 279 - 293, Springer Berlin