Content-Oriented Relevance Feedback in XML-IR Using the Garnata Information Retrieval System

Luis M. de Campos, Juan M. Fernández-Luna,
Juan F. Huete, and Carlos Martín-Dancausa

Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S.I. Informática y de Telecomunicación, CITIC-UGR
Universidad de Granada, 18071 – Granada, Spain
{lci,jmfluna,jhg,cmdanca}@decsai.ugr.es

Abstract. Relevance Feedback (RF) is a technique allowing to enrich an initial query according to the user feedback in order to get results closer to the user's information need. This paper presents a new RF method for keyword queries (content queries). It is based on the re-weighting of the original query terms plus the addition of new query terms from the content of elements jugded as relevant or non-relevant by the user. This RF method is integrated in our search engine, Garnata, and evaluated with the INEX 2007 collection.

1 Introduction

XML (Extensible Mark-up Language) is a general-purpose specification for creating custom mark-up languages which specifies lexical grammars and parsing requirements. Nowadays, this language is used, in most cases, to represent a large diversity of structured documents in digital libraries, intranet, or the Web, so Information Retrieval (IR) is being adapted to handle collections of XML documents as new retrieval models have to be designed and implemented.

Keyword-based queries [1,2,3] are used in lots of XML search engines with collections of documents with unknown or highly heterogeneous structure. This type of queries, also known as content queries (CO queries), is only composed of a sequence of keywords, in oposition to the content and structure (CAS) queries, where taking the most of the internal organization of the documents, structural hints about what XML elements to retrieve and where to look for are included in the query. For CO queries, the IR system is in charge of making the decision of which parts of the documents (XML elements) are the most appropriate (relevant) for the query according only to the keywords contained in the query.

In order to improve the quality of the results, and to offer the user results closer to her/his information need, Relevance Feedback (RF) is used to integrate relevance information provided by a user in the original query. Its objective is the re-weighting of the original query terms and the expansion of the initial

T. Andreasen et al. (Eds.): FQAS 2009, LNAI 5822, pp. 617–628, 2009.

query using the information provided by the user, who has jugded relevant or non-relevant the ranked list of elements returned as output of the original query. There are three different types of RF in XML retrieval:

– Content Only (CO) - Content Only (CO): The original query and the expanded query are only composed of terms.
– Content Only (CO) - Content and Structure (CAS): The original query is only composed of terms and the expanded query generated by the RF module contains terms and structural restrictions.
– Content and Structure (CAS) - Content and Structure (CAS): Both queries are composed of terms and structural restrictions.

The granularity of the elements judged by the users as relevant or non-relevant is different depending on the type of search engine. If the search engine works with documents in plain text (traditional IR), the elements retrieved by the system are documents. This means that the element judged by the user can contain a lot of information that is not interesting to the query, and it is more difficult for her/him to make a decision about weather it is relevant or not. However if the search engine works with XML documents implementing a structured IR system, the retrieved elements are structural units, like paragraphs, sections, and so on, which contain, in general, more specific information for the query. Then it should be easier for the user to determine what is relevant or not. This is the case of our search engine, Garnata [4]. This piece of software is based on Probabilistic Graphical Models, more precisely on Influence Diagram and the corresponding underlying Bayesian networks [14,15].

This paper is devoted to the development of an RF module for Garnata, and is focused on the simplest case of XML RF, namely CO-CO. In general terms, the RF module would work as follows: The user introduces a CO query in the search engine and obtains a ranked list of results (elements of the XML documents). Some of them (the first n elements) are evaluated by the user, indicating whether they are relevant or non-relevant. Then, the RF module, using the element's content judged by the user and the original query, will generate an CO expanded query containing only terms, but with the special feature that all the terms will have associated different weights to indicate if they are more or less important to the query. After that, this expanded query can be fed into the Garnata search engine generating new results.

In order to describe with more detail this techinque, this paper is organized as follows: In Section 2 we will briefly review some works about XML RF that are directly related with our proposal, as well as the probabilistic retrieval model underlying our structured IR system Garnata. In Section 3, our CO-CO RF technique is explained. In Section 4, we shall describe the architecture of the RF module and how it is implemented. The experimental evaluation of the proposed model is presented in Section 5. Finally, Section 6 contains the concluding remarks and some proposals for future work.

2 Preliminaries

2.1 Related Work

In XML-IR, researchers has focussed on RF as a mean of improving retrieval effectiveness. Lots of works have been based on the expansion of the original queries using the Rocchio's algorithm [6], which consists of extracting the most expressive terms from the elements judged relevant by the user. Each element is seen as a retrieved unit, and elements are therefore considered to be independent from each other, even though they appear in the same document. Ruthven and Lalmas [8] have studied different RF techniques as automatic techniques, in which the system modifies the user's query, and interactive techniques, in which the user has control over the query modification. They also consider specific interfaces to RF systems and characteristics of searchers that can affect the use and success of RF systems.

Most of the papers we have found, are concentrated on query expansion based on the retrieved elements with known relevance:

– The work presented in [9] is based on an extension of the vector space model. The major advance achieved is the inclusion of a flexible capability, which allows the system to retrieve at a desired level of granularity (i.e., at the element level).
– In [10], it is described a component ranking algorithm for XML retrieval and shows how to apply known RF algorithms from traditional IR on top of it to achieve RF for XML.
– In [11], the authors investigate the effectiveness of blind (”pseudo”) feedback based on top ranking XML elements.

Pan et al. [12] show an approach for extracting user information needs by RF, maintaining more intelligent personal ontologies, clarifying uncertainties, re-weighting atomic conditions, expanding query, and automatically generating a refined query for the XML retrieval system.

2.2 The Search Engine: Garnata

The search engine to retrieve the relevant material for the user is Garnata [4], an IR System, specially designed to work with structured documents in XML. This system is based on the Context-based Influence Diagram model (CID model) [13], which is supported by Influence Diagrams and Bayesian networks [14,15]. These are probabilistic graphical models specially designed for decision problems in uncertain environments.

The main functionality of Garnata is, given a query, to compute the expected utility of retrieving each element or structural unit of the documents in a XML collection, and then to give a ranking of those units in decreasing order of expected utility. The query Q is formulated in natural language, e.g. “XML format”, where all the terms $t \in Q$ are relevant and have the same importance in the query (this means that $p(t|Q) = 1\ \forall t \in Q$). After that, the system computes the probability of relevance of each structural unit U, $p(U|Q)$. With this information, the search engine makes the decision of retrieving a unit or not also considering the utility of the different structural units. Then, the system offers an ordered list of units by expected utility.

3 The Content-Oriented Relevant Feedback Method

Our content oriented RF method using Garnata aims at identifying XML elements, results of a content query Q submitted to the structured IR System, that are relevant and non-relevant, based on the decision of the user, to re-weight the terms contained in the original query and to add new terms to it. The proposed methodology is based on the idea that by evaluating a set of retrieved elements, the user might obtain new pieces of evidence that may help to discriminate which XML elements are relevant to his/her information need. The proposed RF technique is able to measure the utility of those terms belonging to the judged XML components and to add that information to the query. For example, whenever a term t_i indexes an element judged relevant by the user, perhaps we should increase the belief supporting the relevance of t_i; similarly, if a term t_i only appears in documents which are not relevant, we should decrease its relevance belief.

Once the terms have been extracted from the judged units, the system computes weights that represent the corresponding probabilities of relevance for each selected term given the query ($p(t|Q)$) and, as consequence of this process, we obtain a new query with the original terms and, perhaps, some new terms, where every term has associated a different degree of relevance (probability values). This is the main difference with the original content query which only contain relevant terms for the user and where we can not modify the degree of relevance, i.e. the original query $Q =$"XML format" is equivalent to $Q =$"1.0*XML 1.0*format", 1.0 being the value of maximum probability (the terms are completely relevant to the information need).

By means of the weights associated to those terms in the new query we can perform query term re-weighting and query expansion. In order to determine these weights it will be useful to classify the terms indexing in the judged elements according to the following categories (see Figure 1, where the original query is $Q =$"t_6 t_4 t_3"):

- RQ: terms in the original query that only appear in relevant elements, as t_6.
- NRQ: terms in the original query that only appear in non-relevant elements. In this case, t_3.
- NQ: terms in the original query that appear in both relevant and non-relevant elements, say t_4.
- RT: terms which do not appear in the original query and only appear in relevant elements. RT candidates in Figure 1 are t_1 and t_5.
- NRT: terms which do not appear in the original query and only appear in non-relevant elements, as t_2 in the example.

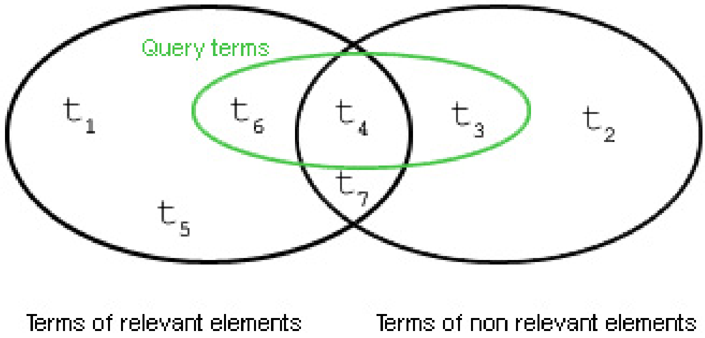

Fig. 1. Candidate Terms

3.1 Computation of Probabilities of Relevance for Original and Expansion Terms

In this section, we are going to describe how the probabilities of term relevance can be computed, given the query, for those terms belonging to the query and for those candidate for query expansion. These are the different alternatives:

- RT terms: The associated probability given the query Q for this kind of terms can be computed in the following two ways:
  - The importance of a term depends on the number of relevant elements in which it appears:
  $$p(t_{RT}|Q) = \frac{n_{tr}}{n_r} \tag{1}$$
  Here, for a candidate term t, n_{tr} denotes the number of relevant elements that contain t, and n_r denotes the number of relevant elements judged.
  - This is an extension of the previous one so the functionality is the same, however we want to penalize the terms that are very common in the document collection:
  $$p(t_{RT}|Q) = \frac{n_{tr}}{n_r} * \frac{idf_t}{maxidf_{RT}} \tag{2}$$
  Where idf_t is the idf of the term t and $maxidf_{RT}$ is the maximum idf of all the RT candidates.
- NRT terms: The probability for this kind of candidate terms is 0 because we want penalize the terms that only appear in non-relevant elements:
  $$p(t_{NRT}|Q) = 0$$
- RQ terms: The probability for this kind of candidates is 1.0 because these terms are doing well their job, so they are very significant:

$$p(t_{RQ}|Q) = 1.0$$

- NRQ terms: This kind of candidates must appear in the new query because they are part of the original content query, but they are not performing well. Consequently, they should be penalized by decreasing the belief supporting their relevance. In this paper we propose that this penalization was a function of the number of non-relevant elements in which they are contained:

$$p(t_{NRQ}|Q) = \frac{1}{n_{t\bar{r}}+1}$$

  Where $n_{t\bar{r}}$ denotes the number of non-relevant elements that contain the candidate term t.
- NQ terms: In spite of these candidates appear in non-relevant elements, their probability is still fixed to 1.0 because they are original query terms and are also contained in relevant elements:

$$p(t_{NQ}|Q) = 1.0$$

3.2 Generating the New Query

Using the candidates described before, the IR system builds a new content-only query from the original content query that is adapted better to the needs of the user. The new query presents the special feature that it contains the probabilities of the terms given the query, which determine the importance of each term of the query. To a better understanding of this new expanded query, its general structure is:

$$t_{RT}{*}p(t_{RT}|Q)\ t_{NRT}{*}p(t_{NRT}|Q)\ t_{RQ}{*}p(t_{RQ}|Q)\ t_{NRQ}{*}p(t_{NRQ}|Q)\ t_{NQ}{*}p(t_{NQ}|Q)$$

As an example, if the original query was “XML format” and we select the RT candidate “metadata” with probability 0.8, the NRT candidate “Java”, the RQ candidate “XML” and the NRQ candidate “format” with probability 0.5, the expanded query would be:

0.8*metadata 0.0*Java 1.0*XML 0.5*format

In the last step, Garnata runs this expanded query, so we have modified the implementation of Garnata for this new kind of queries, because the inclusion of (different) probabilities associated to terms in the query is not accepted by the original version of Garnata.

4 Architecture and Implementation

Figure 2 shows the high-level architecture of our extensible feedback framework. Each candidate term is obtained from processing the query and all the relevant and non-relevant elements judged by the user, who gives positive or negative feedback to some of the results of the original content query, so this feedback is sent together to the original query and the ranking of that query to the search engine.

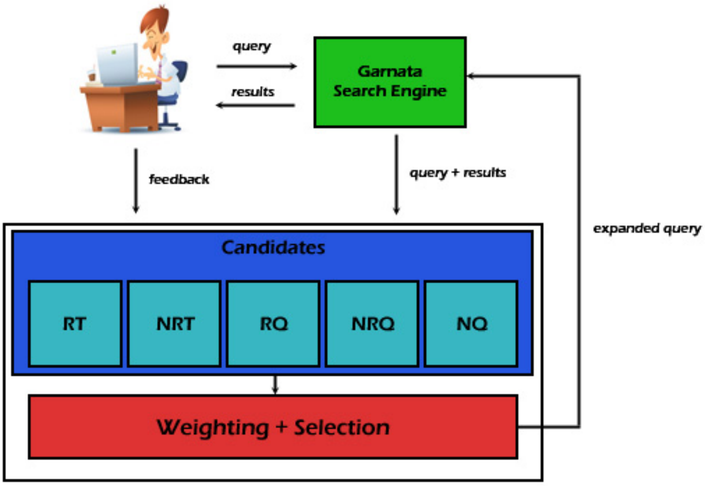

Fig. 2. Architecture of the feedback engine

```
2526465 /article[1] 1
91225 /article[1] 0.587434958179343
210254 /article[1] 0.395297002528691
521930 /article[1] 0.366706675744018
166299 /article[1] 0.346225329702393
7174 /article[1] 0.292725100175064
346830 /article[1] 0.278320145886014
238084 /article[1] 0.258506289632367
266210 /article[1] 0.25503767457693
58079 /article[1] 0.249889161641704
197082 /article[1] 0.247767185372496
1721305 /article[1] 0.23687165337483
743895 /article[1] 0.236149753938922
726304 /article[1] 0.235863131686443
1495421 /article[1] 0.234532038513908
2183748 /article[1] 0.224141738961292
680703 /article[1] 0.217530966737989
```

Fig. 3. Results of a query

When we have obtained, for each kind of candidate, all the terms and calculated their probabilities, the RF modules selects for RT and NRT the first n candidates, which are sorted by their probabilities for RT terms and by the number of non-relevant elements in which the term appears for NRT terms. The system builds the new query with these n candidates of RT and NRT, the query terms and their probabilities. This query is sent back to Garnata, which runs it and presents the results to the user. The user may now again submit feedback for some of the new results, triggering another feedback cycle.

We have implemented the framework in C++ because Garnata was developed in this language and it was easier to integrate both modules. The expanded query is sent to the search engine in the format we commented in the previous section and Garnata is in charge of processing it to separate the terms and the probabilities in order to execute the query with the terms and propagate it, incorporating the probabilities.

The result of a query consists of a file that is composed of tuples with information about the elements retrieved by the search engine. These tuples contain the following information:

– Identifier of the document.
– XPath of the relevant element.
– Expected Utility of the element.

All the tuples are sorted by the expected utility value. We may see an example of this file in Figure 3.

5 Experimental Results

In order to validate our proposal, we have performed several experiment with our search engine Garnata, extended with the RF module, using the collection and CO queries used at INEX 2007[1]. Previously to discuss the results, we shal consider the framework of the experimentation and the evaluation methodology used in it.

Data Set and Framework: The XML document collection considered in the experimentation is the one used in the last editions of the INEX Conference, namely *Wikipedia* (an XML version of the English Wikipedia), at the beginning of 2006 [18] with its 659388 articles (around 4600 Megabytes in size). In terms of the queries (and the corresponding relevance assessments) used in our experiments, we have selected the set of queries developed for INEX'2007 (103 CO queries).

In order to evaluate our proposal, we have considered the Focused task of INEX Ad-hoc track [7]. The objective is to retrieve the most relevant parts of the documents, witout overlapping. In addition, the relevant elements have to be the most appropriate units of retrieval (focused). Therefore, if a chapter is retrieved, and therefore, considered the most appropriate type of element that matches the query, any of its section should be retrieved.

In Garnata this task have been achieved by filtering the ordered list of structural units retrieved in such a way that overlapping has been eliminated. As consequence, those units with the greatest relevance value are chosen as the most appropriate units of retrieval where, in case of tie, we keep the more general unit (the one containing a larger amount of text).

The measures of retrieval effectiveness are those used in the focused task of the INEX'2007 ad hoc track, namely the interpolated precision (iP) at selected recall levels (iP[0.0], iP[0.01], iP[0.05] and iP[0.10]) and the average interpolated precision (AiP), all of them averaged across the 103 queries.

[1] http://inex.is.informatik.uni-duisburg.de/2007/

Evaluation methodology: RF is an iterative process where top N results returned by the search engine are marked as relevant or not relevant by the user. Following the focused task, for each INEX 2007 query, Garnata returns a ranked list of the top 1,500 non-overlapping most focused relevant document parts. Since we wish to emulate the user feedback in real applications, we consider the relevance assessments of the first 10 elements of the initial query execution, using the existing INEX assessments. Then, we process these elements in order to build the new query which is fed back to Garnata obtaining a new list of non-overlapping elements.

In order to evaluate the feedback performance we shall compare the retrieval effectiveness of those results obtained with and without feedback. With this goal in mind, we shall use a **freeze-top** approach[2], adapted for the focused task. Thus, having the two result files for a topic, we process them in the following way:

– Results of the original query (baseline): All the judged elements are removed. As consequence we have a ranked list of those non-overlapping elements at the positions 11 to 1500 in the original ranking. These will be those documents presented to the user when no RF is done.
– Result of the expanded query: We run the expanded query an obtain a ranked list of elements. Then, all the judged elements are added at the top of the ranking with the maximum retrieval value and the focused filter is used in order to remove those overlapping units. Note that, after run this filter, the judged units are necessarily at the top of the ranking. Finally, all these judged elements added before are deleted. As consequence we also have a ranked list of 1490 elements.

Then, these two retrieval lists could be compared in order to measure the impact of RF.

Experimental Results: We have performed different experiments in order to determine the impact of query term re-weighting and query expansion isolately and also combining both approaches. These experiments are:

– Exp. 1: This experiment is a query term re-weighting, i.e. the expanded query is only composed of RQ, NRQ and NQ terms.
– Exp. 2: Original query (query terms with probabilities equal to 1) and the top 10 RT candidates (those with higher probability of relevance) using eq. 1.
– Exp. 3: Original query (query terms with probabilities equal to 1) and the top 10 RT candidates using eq. 2.
– Exp. 4: Original query (query terms with probabilities equal to 1) and those top 10 NRT candidates which appear most frequently in non-relevant units.
– Exp. 5: Query term re-weighting (RQ, NRQ, NQ) and expanded query using the top 10 RT terms using eq. 1.

[2] The other alternative is to follow a residual collection approach, but this approach can not be applied properly for the focused task.

– Exp. 6: Query term re-weighting (RQ, NRQ, NQ) and expanded query using the top 10 RT terms using eq. 2.
– Exp. 7: Query term re-weighting (RQ, NRQ, NQ) and expanded query using the top 10 RT terms using eq. 1 and the top 10 NRT elements.
– Exp. 8: Query term re-weighting (RQ, NRQ, NQ) and expanded query using the top 10 RT terms using eq. 2 and the top 10 NRT elements.

The results of these experiments are summarized in Table 1. It displays the corresponding effectiveness measures for the baseline (Base-CO) and for each one of the above experimental settings. We also indicate the percentage of improvement achieved by these experiments.

Table 1. Comparison between the different RF experiments and the baseline (Base-CO) systems. (% imp. = % improvement).

	Base-CO	Exp. 1	% imp.	Exp. 2	% imp.	Exp. 3	% imp.	Exp. 4	% imp.
iP[0.00]	0.338646	0.352185	3.99	0.325602	-3.85	0.339003	0.11	0.342393	1.10
iP[0.01]	0.296273	0.301896	1.90	0.312909	5.61	0.317543	7.18	0.296757	0.16
iP[0.05]	0.210609	0.211884	0.61	0.268876	27.67	0.244061	15.88	0.210940	0.16
iP[0.10]	0.175126	0.176834	0.98	0.243192	38.87	0.199332	13.82	0.174750	-0.21
MAiP	0.058443	0.058556	0.19	0.085947	47.06	0.080139	37.12	0.058409	-0.06

	Base-CO	Exp. 5	% imp.	Exp. 6	% imp.	Exp. 7	% imp.	Exp. 8	% imp.
iP[0.00]	0.338646	0.328314	-3.05	0.342393	1.10	0.326174	-3.68	0.348205	2.82
iP[0.01]	0.296273	0.311541	5.15	0.316852	6.95	0.309401	4.43	0.322664	8.90
iP[0.05]	0.210609	0.267845	27.18	0.245233	16.44	0.261159	24.00	0.251045	19.20
iP[0.10]	0.175126	0.242914	38.71	0.200200	14.32	0.241597	37.96	0.197924	13.02
MAiP	0.058443	0.085852	46.90	0.080040	36.95	0.085368	46.07	0.080163	37.16

The results of our experiments are quite conclusive: Our RF module obtains, in most cases, better results than the baseline system, nevertheless the best results have been obtained expanding the query using RT terms. With respect to term re-weighting, Exp. 1 shows the best results when it is applied isolately. In this case, we can see that using different probabilities for the term queries does not affect too much in the results. Moreover, when query term re-weighting is used in combination with term expansion (Experiment 5 to 8) the performance decreases. We believe that this performance is due to the short length of the queries (a typical query contains 3 or 4 terms) and the fact that these terms could be selected carefully by the user.

With respect to query expansion, using RT terms (Exp. 2 and 3) is the best solution because the improvements achieved are highly significant, ranging from a minimum of 37% to a maximum of 47%. The best results have been obtained using Eq. 1, i.e. without considering the importance of the term in the collection. Finally, the expansion of the query using NRT terms (Exp. 4 isolately and Exp. 5 to 8) seems that they do not affect the the effectiveness of the system. This performance is due to the low prior probabilities associated to the terms, $P(t)$.

Thus, we have that for a given term, t, in NRT the $P(t|Q)$ is quite similar to $P(t)$, so there is not going to be a big difference if the system propagates the prior probability (in the baseline) or zero (in RF approach) for NRT terms.

6 Concluding Remarks and Future Research

This paper is based on the relevance feedback of content only (CO) queries. It presents an integrated framework for modifying new queries, adding the probabilities of relevance of the different terms of the queries aas well as adding new terms and their probabilities. This framework has been evaluated with INEX 2007 collection.

Our future work will concentrate on adding new candidate terms and probabilities to the terms given the query using other methods to calculate them and extending this work to queries with content and structural constraints.

In the feedback model could be interesting in the fact that the user could assign exhaustiveness values and specificity values for the results of the content queries.

Note that even though this paper considers only binary relevance, it is possible to extend the mechanism presented here to approaches where relevance is measured with a probability-like number between 0 and 1.

Acknowledgements

This work has been jointly supported by the Spanish Ministerio de Educación y Ciencia, and the research program Consolider Ingenio 2010, under projects TIN2008-06566.C04.01 and CSD2007-00018, respectively.

References

1. Balmin, A., et al.: A system for keyword proximity search on XML databases. In: VLDB 2003, pp. 1069–1072 (2003)
2. Guo, L., et al.: XRANK: ranked keyword search over XML documents. In: SIGMOD 2003, pp. 16–27 (2003)
3. Xu, Y., Papakonstantinou, Y.: Efficient keyword search for smallest LCAs in XML databases. In: SIGMOD 2005, pp. 537–538 (2005)
4. de Campos, L., Fernández-Luna, J.M., Huete, J., Romero, A.: Garnata: An information retrieval system for structured documents based on probabilistic graphical models. In: Proceedings of the IMPU 2006 conference, pp. 1024–1031 (2006)
5. Kazai, G., et al.: The INEX evaluation initiative. In: Blanken, et al. (eds.), pp. 279–293.
6. Rocchio Jr., J.: Relevance feedback in information retrieval. In: Salton, G. (ed.) The SMART Retrieval System: Experiments in Automatic Document Processing, ch. 14, pp. 313–323. Prentice Hall, Englewood Cliffs (1971)
7. Kamps, J., Pehcevski, J., Kazai, G., Lalmas, M., Robertson, S.: INEX 2007 evaluation measures. In: Fuhr, N., Kamps, J., Lalmas, M., Trotman, A. (eds.) INEX 2007. LNCS, vol. 4862, pp. 24–33. Springer, Heidelberg (2008)

8. Ruthven, I., Lalmas, M.: A survey on the use of relevance feedback for information access systems. Knowledge Engineering Review 18(1) (2003)
9. Crouch, C.J., Mahajan, A., Bellamkonda, A.: Flexible XML retrieval based on the extended vector model. In: Fuhr, N., Lalmas, M., Malik, S., Szlávik, Z. (eds.) INEX 2004. LNCS, vol. 3493, pp. 292–302. Springer, Heidelberg (2005)
10. Mass, Y., Mandelbrod, M.: Relevance feedback for XML retrieval. In: INEX 2004 Workshop, pp. 154–157 (2004)
11. Sigurbjörnsson, B., Kamps, J., de Rijke, M.: The University of Amsterdam at INEX 2004. In: INEX 2004 Workshop, pp. 104–109 (2004)
12. Pan, H., Theobald, A., Schenkel, R.: Query refinement by relevance feedback in an XML retrieval system. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) ER 2004. LNCS, vol. 3288, pp. 854–855. Springer, Heidelberg (2004)
13. de Campos, L.M., Fernández-Luna, J.M., Huete, J.F.: Using context information in structured document retrieval: An approach using Influence Diagrams. IP&M 40(5), 829–847 (2004)
14. Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer, Heidelberg (2001)
15. Shachter, R.D.: Probabilistic inference and influence diagrams. Oper. Res. 36(4), 589–604 (1988)
16. de Campos, L.M., Fernández-Luna, J.M., Huete, J.F.: Improving the context-based influence diagram for structured retrieval. In: Losada, D.E., Fernández-Luna, J.M. (eds.) ECIR 2005. LNCS, vol. 3408, pp. 215–229. Springer, Heidelberg (2005)
17. Robertson, S., Sparck-Jones, K.: Relevance weighting of search terms. Journal of the American Society of Information Science 27, 129–146 (1976)
18. Denoyer, L., Gallinari, P.: The Wikipedia XML corpus. SIGIR Forum 40, 64–69 (2006)