Decision-theoretic resource selection for different data types in MIND

Henrik Nottelmann and Norbert Fuhr

Institute of Informatics and Interactive Systems, University of Duisburg-Essen,
47048 Duisburg, Germany, {nottelmann,fuhr}@uni-duisburg.de

Abstract. In a federated digital library system, it is too expensive to query every accessible library. Resource selection is the task to decide to which libraries a query should be routed. In this paper, we describe a novel technique that is used in the MIND project. Our approach, decision-theoretic framework (DTF), differs from existing algorithms like CORI in two ways: It computes a selection which minimises the overall costs (e.g. retrieval quality, time, money) of the distributed retrieval. And it allows for other data types beside text (e.g., names, years, images), whereas other resource selection techniques are restricted to text.

1 Introduction

Resource selection is the task to determine automatically useful ("relevant") collection in a federation of digital libraries (DLs). Traditional algorithms (e.g. GlOSS, CORI) compute similarities between the library and the query, and retrieve a constant number of documents from the top-ranked libraries.

The GlOSS system [7] is based on the vector space model and – thus – does not refer to the concept of relevance. For each library, a goodness measure is computed which is the sum of all (SMART) scores of all documents in this library w. r. t. the current query. Libraries are ranked according to the goodness values.

The state-of-the-art system CORI [3] uses the INQUERY retrieval system which is based on inference networks. The resource selection task is reduced to a document retrieval task, where a "document" is the concatenation of all documents of one library. The indexing weighting scheme is quite similar to the DTF one, but applied to libraries instead of documents. Thus, term frequencies are replaced by document frequencies, and document frequencies by collection frequencies. CORI also covers the data fusion problem, where the library score is used to normalise the document score. CORI is one of the best performing resource selections models.

In contrast, the decision-theoretic framework (DTF) [6, 11] has a better theoretic foundation: The task is to find the selection with minimum costs (which depend on different criteria like retrieval quality, time or money). A user can choose different selection policies by specifying the importance of the different cost sources. Three different methods for estimating retrieval quality for text libraries have been developed for DTF so far.

In this paper, we extend DTF towards additional data types (like names, years or images) and search predicates (e.g. sounds-like for names or = for years), which have to be handled slightly different.

We implemented this new method within the MIND project. MIND is a federated digital library system for non-co-operating and multi-media DLs.

The rest of this paper is organised as follows. In the next section, we introduce the decision-theoretic framework for resource selection. In section 3, we describe the retrieval model used in DTF, and how different data types (text, names, years, images) are integrated. Different methods for estimating retrieval quality are presented in section 4. The last section contains some concluding remarks and an outlook to future work.

2 Decision-theoretic framework

The basic assumption of the decision-theoretic framework (DTF) [6, 11] is that we can assign specific retrieval costs $C_i(s_i,q)$ to each digital library DL_i, where s_i is the number of documents retrieved for query q. The term "costs" is used in a broad way and also includes—besides money—cost factors like time and quality.

If the user specifies (together with her query) the total number n of documents which should be retrieved, the task then is to compute an optimum solution, i.e. a vector $s = (s_1, s_2, \ldots, s_m)^T$ with $|s| = \sum_{i=1}^m s_i = n$ which minimises the overall costs:

$$M(n,q) := \min_{|s|=n} \sum_{i=1}^{m} C_i(s_i,q). \tag{1}$$

For $C_i(s_i,q)$, costs from different sources should be considered:

Effectiveness: Probably most important, a user is interested in getting many relevant documents. Thus we assign user-specific costs C^+ for viewing a relevant document and costs $C^- > C^+$ for viewing an irrelevant document. If $r_i(s_i,q)$ denotes the number of relevant documents in the result set when s_i documents are retrieved from library DL_i for query q, we obtain the cost function

$$C_i^{rel}(s_i,q) := r_i(s_i,q) \cdot C^+ + [s_i - r_i(s_i,q)] \cdot C^-. \tag{2}$$

Time: This includes computation time at the library and communication time for delivering the result documents over the network. These costs can easily be approximated by measuring the response time for several queries. In most cases, a simple affine linear cost function is sufficient.

Money: Some DLs charge for their usage, and monetary costs often are very important for a user. These costs have to be specified manually. In most cases, the cost function is purely linear (reflecting per-document-charges).

By summing up the costs from different sources, we arrive at an overall cost function $C_i(s_i,q)$ with user-defined cost parameters C^+, C^-, C^t (for time) and C^m (money). Thus, a user can specify her own selection policy (e.g. cheap and fast results with a potentially smaller number of relevant documents). But as we do not know the actual costs (particularly the number of relevant documents) in advance, we switch to expected costs $EC_i(s_i,q)$ (for relevancy costs, using the expected number $E[r_i(s_i,q)]$ of relevant documents):

$$EM(n,q) := \min_{|s|=n} \sum_{i=1}^{m} EC_i(s_i,q), \tag{3}$$

$$EC_i(s_i,q) := EC_i^{rel}(s_i,q) + C^t \cdot EC_i^{time}(s_i,q) + C^m \cdot EC_i^{money}(s_i,q). \tag{4}$$

In function 3, the expected costs $EC_i(s_i, q)$ are increasing with the number s_i of documents retrieved. Thus, the algorithm presented in [6] can be used for computing an optimum solution.

3 The MIND retrieval model

In this section we describe the underlying retrieval model and the integration of different data types.

3.1 General model

The retrieval model follows Risjbergen's [16] paradigm of IR as uncertain inference, a generalisation of the logical view on databases. In uncertain inference, IR means estimating the probability $Pr(q \leftarrow d)$ that the document d logically implies the query q ("probability of inference").

In MIND, documents adhere to a schema which defines their structure. In this paper, we simply assume that there is only one single schema; a method for dealing with heterogeneous schemas is presented in [10]. Schemas are built on top of data types. A data type D defines the set of possible values in the documents $dom(D)$ (the "domain") and a set of search predicates $p{:}dom(D) \times dom(D,p) \rightarrow [0,1]$, where $dom(D,p)$ is the set of all possible comparison values w. r. t. this search predicate.

A schema S is a set of schema attributes, and each schema attribute $A \in S$ belongs to a specific data type $dt(A)$. Analogously, a document d contains a set of document attributes, where a document attribute is a pair $(A, v(A))$ of a schema attribute $A \in S$ and a document value $v(A) \in dom(dt(A))$:

$$d := \{(au, C.J.\ Rijsbergen), (ti, Probabilistic\ Retrieval\ Revisited)\}.$$

A query q (adhering to the schema S) consists of a set of query conditions; a query condition c is a tuple $(w(c), A(c), p(c), v(c))$ of a probabilistic weight $w(c) \in [0,1]$ specifying the importance of this condition, a schema attribute $A(c) \in S$, a predicate $p(c)$ (supported by the data type $dt(A)$)) and a comparison value $v(c) \in dom(dt(A), p)$ Queries are normalised, i.e. the sum of the query condition weights $s(q) := \sum_{c \in q} w(c)$ equals 1:

$$q := \{(0.3, ti, =, xml), (0.2, ti, =, retrieval), (0.5, au, =, fuhr)\}.$$

As we consider different data types and predicates, we split the query q into subqueries q_p w. r. t. the different predicates p, and normalise the subquery (with the normalisation factor $s(q_p)$). The query q_1 from above would be split into 2 subqueries:

$$q_{ti,=} = \{0.6, ti, =, xml), (0.4, ti, =, retrieval)\}, q_{au,=} = \{(1, au, =, fuhr)\}.$$

We assume disjoint query terms, and view the query condition weight $w(c)$ as the probability $Pr(q \leftarrow c)$. Then, we can apply a linear retrieval function [17]:

$$Pr(q_p \leftarrow d) := \sum_{c_j \in q_p} \underbrace{Pr(q_p \leftarrow c_j)}_{\text{query condition weight}} \cdot \underbrace{Pr(c_j \leftarrow d).}_{\text{indexing weight}}$$

The probability $Pr(c \leftarrow d)$ is the indexing weight of document d w. r. t. condition c. Of course, the notion "indexing weight" does not imply that this probability is actually stored in an index, it can also be computed on the fly.

We are interested in the probability of relevance $Pr(\mathrm{rel}|q_p, d)$, so we use predicate-specific mapping functions [11] for approximating the relationship between probabilities $Pr(\mathrm{rel}|q_p, d)$ and $Pr(q_p \leftarrow d)$:

$$f_p : [0,1] \mapsto [0,1], \quad f_p(Pr(q_p \leftarrow d)) \approx Pr(\mathrm{rel}|q_p, d). \tag{5}$$

We can convert the probabilities of relevance of the subqueries into a probability of relevance for the complete document:

$$Pr(\mathrm{rel}|q,d) := \sum_{p \in q} s(q'_p) \cdot Pr(\mathrm{rel}|q_p, d).$$

In the rest of this section, we present definitions for indexing weights $Pr(c \leftarrow d)$ and mapping functions f_p for different data types and predicates.

3.2 Data type "Text"

For `Text`, the comparison value of a condition c is a term t. In MIND, we use two predicates `contains` (with stemming and stop word removal) and `containsNoStem` (terms are not stemmed, but stop words are removed). We use a modified BM25 scheme [14] for the indexing weights:

$$P(t \leftarrow d) := \frac{tf(t,d)}{tf(t,d) + 0.5 + 1.5 \cdot \frac{dl(d)}{avgdl}} \cdot \frac{\log \frac{numdl}{df(t)}}{\log |DL|}. \tag{6}$$

Here, $tf(t,d)$ is the term frequency (number of times term t occurs in document d), $dl(d)$ denotes the document length (number of terms in document d), *avgdl* the average document length, *numdl* the sample or library size (number of documents), $|DL|$ the library size), and $df(t)$ the document frequency (number of documents containing term t). We modified the standard BM25 formula by the normalisation component $1/\log|DL|$ to ensure that indexing weights are always in the closed interval $[0,1]$, and can thus be regarded as a probability. The resulting indexing weights are rather small; but this can be compensated by the mapping functions.

Experiments [12] showed that a logistic function [4, 5], defined by two parameters b_0 and b_1, yields a good approximation quality (see also Fig. 1):

$$f_p : [0,1] \rightarrow [0,1], \; f(x) := \frac{\exp(b_0 + b_1 x)}{1 + \exp(b_0 + b_1 x)}. \tag{7}$$

The logistic function can also be justified by a theoretic point of view: In an ideal situation, exactly the documents in the ranks $1, \ldots, l$ are relevant, and the documents in the remaining ranks $l+1, \ldots$ are irrelevant. Thus, the relationship function should be a step function. Obviously, no information retrieval system can ensure this requirement, so a continuous function f which approximates the discrete step function is more appropriate. The logistic function in formula 7 is such an approximation.

Within MIND, we also investigated transcripts of speech recognisers. We observed that we can handle them in the same way as "ordinary" text.

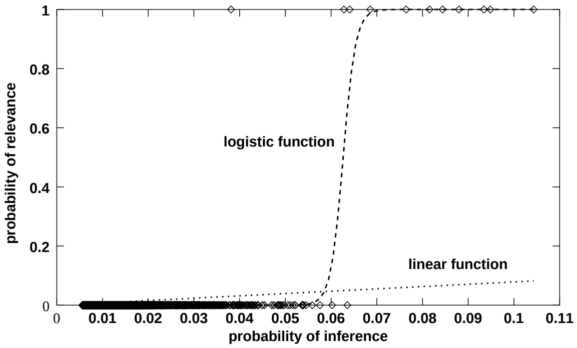


Fig. 1. Example query results and fit with linear and logistic function

3.3 Data type “Name”

This data type supports two Boolean predicates `=` and `sounds-like`:

$$Pr(c \leftarrow d) \in \{0,1\},$$

$$Pr((author, sounds-like, Jones) \leftarrow (author, Johnson)) = 1.$$

For Boolean predicates, the identity function is a natural choice for the mapping function:

$$f_p \equiv id\ ,\ f_p(x) := x.$$

Vague predicates like `edit-distance` or `n-grams` could be considered as well but are not investigated yet.

3.4 Data Type “Year”

The data type `Year` has the Boolean predicates `=`, `<`, `>`, `<=` and `>=`, i.e. the indexing weight is in $\{0,1\}$. As for the data type `Name`, we use the identify mapping function.

Furthermore, three different vague predicates $\sim$=, $\sim$< and $\sim$> are defined:

$$Pr(c_{\sim>} \leftarrow d) := \begin{cases} 1 - \frac{v(c)-v(d)}{v(d)} & , \quad v(c) > v(d) \\ 1 & , \quad else \end{cases} , \tag{8}$$

$$Pr(c_{\sim=} \leftarrow d) := 1 - \left(\frac{v(c) - v(d)}{v(d)}\right)^2 . \tag{9}$$

For example, we get

$$Pr((1, pub, \sim=, 1999) \leftarrow (pub, 2000)) = 1 - \left(\frac{1999 - 2000}{2000}\right)^2 = 0.99999975.$$

For these vague predicates, we apply a logistic function

$$f_p : [0,1] \rightarrow [0,1],\ f(x) := \frac{\exp(b_0 + b_1 \cdot x)}{1 + \exp(b_0 + b_1 \cdot x)}.$$

For $\sim=$, useful parameters are $b_0 = 4.7054 - 12590100$, $b_1 = 12590100$. Then, we get

$$b_0 + b_1 \cdot 0.99999975 \approx 1.5578, \exp(b_0 + b_1 \cdot 0.99999975) \approx 4.7487,$$
$$f(0.99999975) \approx 0.8260.$$

These definitions of the indexing weights and mapping function are equivalent to the retrieval function used in fact retrieval for single condition queries, and a close approximation for queries with multiple conditions.

3.5 Data Type “Image”

For images, different similarity functions (usually distance measures) for comparing the condition value c and the image stored in document d can be considered. For colour histograms, the colour space is divided into bins. A histogram $hist(c)$ (comparison value in query) or $hist(d)$ (document image) is a vector H, where the component H_i counts the frequency of the colours in bin i in the image

Possible distance measures for colour histograms include Minkowski-form distance, Kullback-Leibler divergence, χ^2 statistics or a quadratic-form distance [15], [8], [9], or a normalised histogram intersection measure for the indexing weights:

$$Pr(c \leftarrow d) := \frac{\sum_i \min(hist(c)_i, hist(d)_i)}{\sum_i hist(d)_i}.$$

$$Pr(((image, colour, (0.5, 0.7)) \leftarrow (image, (0.4, 0.2))) = \frac{0.4 + 0.2}{0.7 + 0.3} = 0.6.$$

For images, linear or logistic mapping functions are possible.

3.6 Parameter learning

Most mapping functions depend on some query-specifiy parameters. However, in practice we do not have query-specific relevance data, so we have to derive query-independent parameters from a learning sample (for each digital library).

In MIND, the parameters are learned using the Gnuplot[1] implementation of the nonlinear least-squares (NLLS) Marquardt-Levenberg algorithm [13]. As we don’t have relevance judgements for all documents in practice, we only considered the 100 top-ranked documents.

4 Estimating retrieval quality for resource selection

Resource selection accuracy in this model heavily depends on good approximations of the number of relevant documents in the result set. Within MIND, we developed two new methods [11]: DTF-sample (simulated retrieval on sample) can be used for all data types, whereas DTF-normal (modelling indexing weights by a normal distribution)

[1] `http://www.ucc.ie/gnuplot/gnuplot.html`

only works for text (but there, it outperforms DTF-sample). The third and oldest method DTF-rp [6] can only be used for linear mapping functions.

In this section we describe briefly these three methods for estimating the expected number $E[r(s,q)]$ of relevant documents in the first s documents. All of them require some metadata (e.g. average indexing weights, document frequencies). In non-co-operating environments, these "resource descriptions" can be created automatically by query-based sampling [2]. "Random" subsequent queries are submitted to the library, and the retrieved documents are collected (forming the "sample", from which the metadata is extracted). With reasonably low costs, an accurate resource description can be constructed from samples of e.g. 300 documents.

For textual documents, a TREC-based evaluation of all three methods can be found in [11]. We were not able to evaluate the technique for other media types yet due to lack of an appropriate test-bed with relevance judgements.

4.1 Recall-precision-function

With the linear retrieval function for $Pr(q \leftarrow d)$, a linear mapping functions $f_p(x) := c_0 + c_1 \cdot x$ and a query q where all conditions refer to the same predicate (which allows us to use the same mapping function), we can compute the expected number $E(\text{rel}|q, DL_i)$ of relevant documents in DL_i as:

$$\begin{aligned} E(\text{rel}|q,DL) &= |DL| \cdot E(\text{rel}|q,d), \\ E(\text{rel}|q,d) &= c_0 + c_1 \cdot \sum_{c_j \in q} Pr(q \leftarrow c_j) \cdot \underbrace{\left[\frac{1}{|DL|} \sum_{d \in DL} Pr(c_j \leftarrow d)\right]}_{\text{average indexing weight}}. \end{aligned}$$

Obviously, this can be extended towards queries referring to multiple data types.

In addition, DTF-rp approximates the recall-precision function of a DL by a linearly decreasing function

$$P : [0,1] \to [0,1],\ P(R) := P^0 \cdot (1 - R). \tag{10}$$

With expected precision $E[r(s,q)]/s$ and expected recall $E[r(s,q)]/E(\text{rel}|q,DL)$, we can estimate the number of relevant documents when retrieving s documents [6]:

$$E[r(s,q)] := \frac{P^0 \cdot E(\text{rel}|q,DL) \cdot s}{E(\text{rel}|q,DL) + P^0 \cdot s}. \tag{11}$$

4.2 Simulated retrieval on sample

For DTF-sample, we index the complete library sample instead of only extracting statistical metadata. In the resource selection phase, retrieval is simulated with the same query q on this small sample (e.g. 300 documents), obtaining a probability of relevance $Pr(\text{rel}|q,d)$ for each sample document. This results in the empirical, discrete density p of the corresponding distribution.

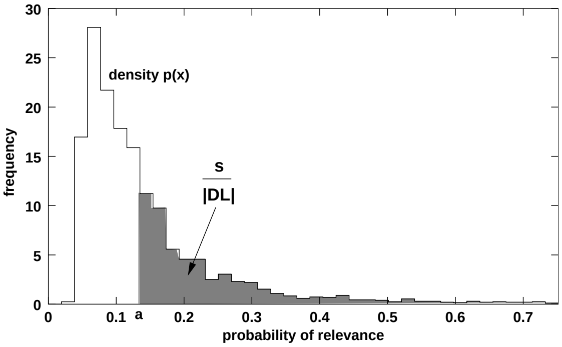


Fig. 2. Density of probabilities of relevance and computation of $E[r(s,q)]$

Figure 2 shows how we can estimate the number of relevant documents in the result set of s documents. The grey area (the area below the graph from a to 1) denotes the fraction $s/|DL|$ of the documents in the library which are retrieved. Thus, we have to find a point $a \in [0,1]$ (the smallest probability of relevance among the s retrieved documents) such that

$$s = |DL| \int_a^1 p(x)\, dx. \tag{12}$$

With this point a, the expected number of relevant documents in the result set can be computed as the expectation of the probabilities of relevance in this area:

$$E[r(s,q)] = |DL| \int_a^1 p_i(x) \cdot x\, dx.$$

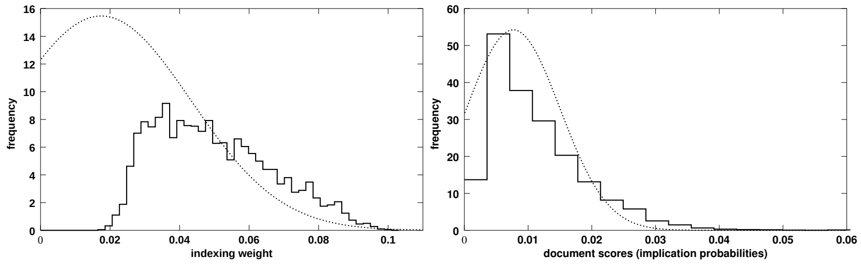
Obviously, this method can be used for all data types.

4.3 Normal distribution

As DTF-sample, DTF-normal estimates the distribution of the probabilities of relevance $Pr(\text{rel}|q,d)$, but based on a new theoretic model. For text, early experiments showed that the empirical, discrete distribution of the indexing weights $Pr(t \leftarrow d)$ (viewed as a random variable X_t for a term t) can be approximated by a normal distribution (defined by the expectation μ and the variance σ)

$$p(x,\mu,\sigma) := \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp(-\frac{(x-\mu)^2}{2\sigma^2}). \tag{13}$$

For improved readability, we left out a huge peak at zero in figure 3(a). This peak is one result of the large amount of documents which do not contain the term. The second effect is that the expectation is close to zero, and the normal distribution density is positive also for negative values. We ignore this, as we are mainly interested in the high indexing weights (the tail of the distribution).



(a) Indexing weights

(b) Document scores

Fig. 3. Distributions with normal distribution fit

The probabilities of inference $Pr(q \leftarrow d)$ (also called "scores") can also be seen as a random variable X, a linear combination of the X_t with constants $a_i := Pr(q \leftarrow t_i)$. As the X_t follow a normal distribution with parameters μ_t and σ_t, X is also normally distributed (see figure 3(b)) with parameters

$$\mu = \sum_{i=1}^{l} a_i \cdot \mu_{t_i}, \quad \sigma = \sqrt{\sum_{i=1}^{l} (a_i \cdot \sigma_{t_i})^2}. \tag{14}$$

From the document score distribution, we can estimate the scores (probabilities of inference) of the s top-ranked documents easily. By applying the mapping function, the probabilities of relevance can be computed.

5 Conclusion and outlook

In this paper we introduce a resource selection method which can—in addition to text—also deal with other data types like names, years, images and others. This method is based on the decision-theoretic framework which assigns costs to document retrieval, and aims at minimising the overall costs from all selected libraries. In contrast to traditional resource ranking algorithms like GlOSS or CORI, DTF computes a clear cutoff for the number of libraries queried, and the number of documents which should be retrieved from each of these libraries. For DTF, the selection solution is a vector specifying for each library the number of documents which have to be retrieved.

In the decision-theoretic framework, data types play their role in estimating retrieval quality (time spent on a library and monetary costs are independent of the data types used in queries). Earlier work focused on text. However, two of the three methods for estimating retrieval quality—DTF-rp (recall-precision function) and DTF-sample (simulated retrieval on sample)— can also be used for non-text datatypes. The third method—DTF-normal (normal distribution of indexing weights)—can only be used for texts.

In future, we will extend our mapping function evaluations on other data types. The biggest challenge will be to find data sets using non-text data types together with relevance judgements (the evaluation for text is based on TREC data).

6 Acknowledgements

This work is supported by the EU commission under grant IST-2000-26061 (MIND).

References

[1] *Proceedings of the 26st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, 2003. ACM.

[2] J. Callan and M. Connell. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19(2):97–130, 2001.

[3] J. Callan, Z. Lu, and W. Croft. Searching distributed collections with inference networks. In E. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, New York, 1995. ACM. ISBN 0-89791-714-6.

[4] S. Fienberg. *The Analysis of Cross-Classified Categorical Data*. MIT Press, Cambridge, Mass., 2. edition, 1980.

[5] D. Freeman. *Applied Categorical Data Analysis*. Dekker, New York, 1987.

[6] N. Fuhr. A decision-theoretic approach to database selection in networked IR. *ACM Transactions on Information Systems*, 17(3):229–249, 1999.

[7] L. Gravano and H. Garcia-Molina. Generalizing GIOSS to vector-space databases and broker hierarchies. In U. Dayal, P. Gray, and S. Nishio, editors, *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases*, pages 78–89, Los Altos, California, 1995. Morgan Kaufman.

[8] S. Kullback. Information theory and statistics. Dover, New York, NY, 1968.

[9] W. Niblack, R. Barber, E. W., M. Flickner, G. E.H., D. Petkovic, P. Yanker, C. Faloutsos, G. Taubin, and Y. Heights. Querying images by content, using color, texture, and shape. In *Proceedings SPIE Conference on Storage and Retrieval for Image and Video Databases*, 1993.

[10] H. Nottelmann and N. Fuhr. Combining DAML+OIL, XSLT and probabilistic logics for uncertain schema mappings in MIND. In *European Conference on Digital Libraries (ECDL 2003)*. Springer, 2003.

[11] H. Nottelmann and N. Fuhr. Evaluating different methods of estimating retrieval quality for resource selection. In [1].

[12] H. Nottelmann and N. Fuhr. From uncertain inference to probability of relevance for advanced IR applications. In F. Sebastiani, editor, *25th European Conference on Information Retrieval Research (ECIR 2003)*, pages 235–250. Springer, 2003.

[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, editors. *Nested Relations and Complex Objects in Databases*. Cambridge University Press, 1992.

[14] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC. In *Text REtrieval Conference*, pages 21–30, 1992.

[15] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[16] C. J. van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29(6):481–485, 1986.

[17] S. Wong and Y. Yao. On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1):38–68, 1995.