Combining CORI and the decision-theoretic approach for advanced resource selection

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 this paper we combine two existing resource selection approaches, CORI and the decision-theoretic framework (DTF). The state-of-the-art system CORI belongs to the large group of heuristic resource ranking methods which select a fixed number of libraries with respect to their similarity to the query. In contrast, DTF computes an optimum resource selection with respect to overall costs (from different sources, e.g. retrieval quality, time, money). In this paper, we improve CORI by integrating it with DTF: The number of relevant documents is approximated by applying a linear or a logistic function on the CORI library scores. Based on this value, one of the existing DTF variants (employing a recall-precision function) estimates the number of relevant documents in the result set. Our evaluation shows that precision in the top ranks of this technique is higher than for the existing resource selection methods for long queries and lower for short queries; on average the combined approach outperforms CORI and the other DTF variants.

1 Introduction

Today, there are thousands of digital libraries (DLs) in the world, most of them accessible through the WWW. For an information need, a decision must be made which libraries should be searched. This problem is called "library selection", "collection selection", "database selection" or "resource selection". We use the latter term throughout this paper.

Recently several automatic selection methods have been proposed (see Sect. 2). In general they compute a ranking of libraries (based on similarities between the library and the query), and retrieve a constant number of documents from the top-ranked libraries. CORI [3] is one of the best performing resource ranking approaches.

In contrast to these heuristic methods, the decision-theoretic framework (DTF) [6, 9] 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). Thus, the system computes a clear cutoff for the number of libraries queried, and the number of documents which should be retrieved from each of these libraries. A user can choose different selection policies by specifying the importance of the different cost sources.

For DTF, different methods for estimating retrieval quality (i.e., the number of relevant documents in the result set) have been proposed. Here, we concentrate on DTF-rp. This method estimates the number of relevant documents in the complete DL, and uses

a recall-precision function for computing the number of relevant documents in the result set. The quality of this resource selection variant is about the same as for CORI [9].

In this paper, we combine the advantages of both models: We use CORI for computing library scores, and map them with a linear or a logistic function onto the number of relevant documents in the complete DL. Then, this estimation is used in DTF-rp for estimating the number of relevant documents in the result set, and an optimum selection is computed by DTF. We investigate different approximations for the recall-precision function for both resulting "DTF-cori" variants as well as for DTF-rp.

One major advantage of our approach is that we extend the range of applications of CORI, as now other cost sources like time or money can be incorporated very easily.

The rest of this paper is organised as follows. First, we give an overview of related work, i.e. some other resource selection algorithms. Then, we describe CORI (Sec. 3) and the decision-theoretic framework (DTF, Sec. 4). In Sec. 5, we introduce our new, combined approach for resource selection with CORI and DTF. We compare this new approach with CORI and the two best performing DTF variants in Sec. 6. The last section contains some concluding remarks and an outlook to future work.

2 Related work

Most of the resource selection algorithms follow the resource ranking paradigm: First, they compute a score for every library. Then, the top-ranked documents of the top-ranked libraries are retrieved and merged in a data fusion step. CORI (see Sec. 3) belongs to the resource ranking algorithms, whereas the decision-theoretic framework (DTF, see Sec. 4) does not rank DLs but explicitly computes for each library the number of documents which have to be returned.

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 scores (e.g. SMART scores) of all documents in this library w. r. t. the current query. Libraries are ranked according to the goodness values.

Other recent resource selection approaches are language models [14] (slightly better than CORI) and the cue validity variance model (CVV) [4] (slightly worse than CORI).

Query-based sampling is a technique for deriving statistical resource descriptions (e.g. average indexing weights, document frequencies) automatically in non-co-operating environments [1]. "Random" subsequent queries are submitted to the library, and the retrieved documents are collected. With reasonably low costs (number of queries), an accurate resource description can be constructed from samples of, e.g., 300 documents. Very recently, the problem of estimating the number of documents in a library (which in particular is important for DTF) has been investigated. Starting from query-based sampling, a sample-resample algorithm has been proposed in [13].

3 CORI

CORI is the state-of-the-art resource selection system [3, 5]. CORI uses the inference network based system INQUERY for computing DL scores, and ranks the collections w. r. t. these scores.

INQUERY is an IR system, and as such, it ranks documents. In CORI, all documents in one collection are concatenated to form "meta-documents". As a consequence, the document nodes in the inference network are replaced by meta-documents, and the net has a moderate size. Therefore, resource selection with CORI is fast; for N collections, resource selection is equivalent to an IR run on N meta-documents. The frequency values are higher, but that does not affect computational complexity. A second advantage is that the same infrastructure can be used for both resource selection and document retrieval; there is no need for designing new index structures or algorithms.

Instead of the common $tf \cdot idf$ weighting scheme, $df \cdot icf$ is used, replacing the term frequency of a term by its document frequency df, the document frequency by the collection frequency cf (the number of libraries containing the term), and the document length by the collection length cl (the number of terms in the DL). Thus, the belief in a DL due to observing query term t (the "indexing weight" of term t in the "meta-document" DL) is determined by:

$$T := \frac{df}{df + 50 + 150 \cdot \frac{cl}{avgcl}} \tag{1}$$

$$I := \frac{\log(\frac{N+0.5}{cf})}{\log(N+1)} \tag{2}$$

$$Pr(t|DL) := 0.4 + 0.6 \cdot T \cdot I. \tag{3}$$

with N being the number of libraries which have to be ranked.

This indexing weighting scheme is quite similar to DTF's one (see next section), but applied to libraries instead of documents. As a consequence, in CORI the resource selection task is reduced to a document retrieval task on "meta-documents". The score of a DL depends on the query structure, but typically (and in this paper) it is the average of the beliefs $Pr(t|DL)$ for the query terms (i.e., a linear retrieval function is used with weight $1/ql$ for each query term).

CORI then selects the top-ranked libraries (the number of selected libraries is fixed before; typically, 10 DLs are chosen) and retrieves an equal number of documents from each selected DL.

CORI also covers the data fusion problem, where the library score is used to normalise the document score.

First the DL scores $C := Pr(q|DL)$ are normalised to $[0,1]$:

$$C' := \frac{C - C_{min}}{C_{max} - C_{min}}, \tag{4}$$

where C_{min} and C_{max} are the minimum and maximum DL scores for that query.

Then, the document score $D := Pr(q|d)$ is normalised to D'' by

$$D' := \frac{D - D_{min}}{D_{max} - D_{min}}, \tag{5}$$

$$D'' := \frac{1.0 \cdot D' + 0.4 \cdot C' \cdot D'}{1.4}. \tag{6}$$

Finally, the retrieved documents are re-ranked according to the normalised scores D''.

4 Decision-theoretic framework

This section briefly describes the decision-theoretic framework (DTF) for resource selection [6, 9].

4.1 Cost-based resource selection

The basic assumption is that we can assign specific retrieval costs $C_i(s_i,q)$ to each digital library DL_i when s_i documents are retrieved for query q. The term "costs" is used in a broad way and also includes—besides money—cost factors like time and quality.

The user specifies (together with her query) the total number n of documents which should be retrieved. The overall number of all collections is denoted by m. The task then is to compute an optimum solution, i.e. a vector $s = (s_1, s_2, \ldots, s_m)^T$ which minimises the overall costs:

$$M(n,q) := \min_{|s|=n} \sum_{i=1}^{m} C_i(s_i,q). \tag{7}$$

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

$$C_i^{rel}(s_i,q) := r_i(s_i,q) \cdot C^+ + [s_i - r_i(s_i,q)] \cdot C^-. \tag{8}$$

Time: This includes computation time at the library site 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 (per-document-charges).

All these costs are summed up to the overall cost function $C_i(s_i,q)$. With cost parameters C^+, C^-, C^t (time) and C^m (money), a user can specify her own selection policy (e.g. cheap and fast results). As the actual costs are unknown 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):

$$EM(n,q) := \min_{|s|=n} \sum_{i=1}^{m} EC_i(s_i,q). \tag{9}$$

In formula 9, 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. Finally, all DLs with $s_i > 0$ are queried.

4.2 Retrieval model

In this subsection we describe two methods for estimating retrieval quality, i.e. the expected number $E[r_i(s_i,q)]$ of relevant documents in the first s_i documents of a result set for all queries q. Both follow Rijsbergen's [15] 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, where both d and q are logical formulae (set of terms with query term weights $Pr(q \leftarrow t)$ and indexing term weights $Pr(t \leftarrow d)$, respectively).

If we assume disjointness of query terms, we can apply the widely used linear retrieval function [17] for computing the probability of inference:

$$Pr(q \leftarrow d) := \sum_{t \in q} \underbrace{Pr(q \leftarrow t)}_{\text{query condition weight}} \cdot \underbrace{Pr(t \leftarrow d)}_{\text{indexing weight}} . \tag{10}$$

So far, this model does not cope with the concept of relevance. However, the decision-theoretic framework is based on estimates of the number $r(s,q)$ of relevant documents in the result set containing the first s documents. This number can be be computed using the probability $Pr(\text{rel}|q,d)$ that document d is relevant w. r. t. query q.

In [10], mapping functions have been proposed for transforming probabilities of inference into probabilities of relevance:

$$f : [0,1] \mapsto [0,1], \quad f_p(Pr(q \leftarrow d)) \approx Pr(\text{rel}|q,d). \tag{11}$$

Different functions can be considered as mapping functions; in previous work, linear and logistic functions have been investigated.

4.3 Estimating retrieval quality with recall-precision function

Several methods for estimating retrieval quality have been developed within the decision-theoretic framework, where retrieval quality is measured as the expected number $E[r(s,q)]$ of relevant documents in the first s documents.

We only employ "DTF-rp" [6] in this work. This method first estimates the number of relevant documents in the complete DL. Then, a recall-precision function is used for computing the expected number of relevant documents in a result set.

DTF-rp is based on a linear mapping function [16]

$$f : [0,1] \mapsto [0,1], f(x) := c \cdot x \tag{12}$$

with constant $c := Pr(\text{rel}|q \leftarrow d)$.

We can compute the expected number $E(\text{rel}|q,DL)$ of relevant documents in DL as

$$E(\text{rel}|q,DL) = \sum_{d \in DL} Pr(\text{rel}|q,d) \tag{13}$$

$$= |DL| \cdot c \cdot \sum_{t \in q} Pr(q \leftarrow t) \cdot \mu_t \tag{14}$$

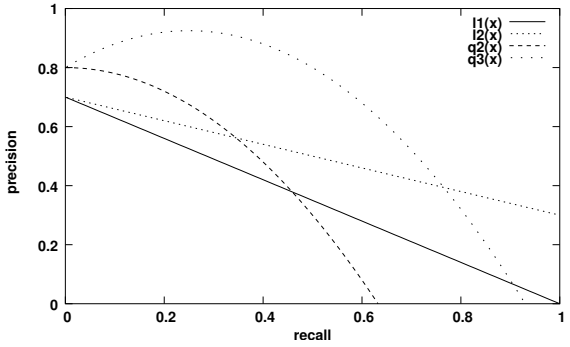

Fig. 1. Different recall/precision functions

with the average indexing weight of term t in DL

$$\mu_t := \frac{1}{|DL|} \sum_{d \in DL} Pr(t \leftarrow d). \tag{15}$$

We can assume different shapes of recall-precision functions, e.g. a linearly decreasing function with only one variable (called "l1" in the remainder) or two degrees of freedom ("l2"), or a quadratic function with two ("q2") or three degrees of freedom ("q3").

The shape of these recall-precision functions is depicted in Fig. 1. They are defined by:

$$P_{l1} : [0,1] \mapsto [0,1] \quad P_{l1}(R) := l_0 \cdot (1-R) = l_0 - l_0 \cdot R, \tag{16}$$

$$P_{l2} : [0,1] \mapsto [0,1] \quad P_{l2}(R) := l_0 - l_1 \cdot R, \tag{17}$$

$$P_{q2} : [0,1] \mapsto [0,1] \quad P_{q2}(R) := q_0 - q_2 \cdot R^2, \tag{18}$$

$$P_{q3} : [0,1] \mapsto [0,1] \quad P_{q3}(R) := q_0 + q1 \cdot R - q_2 \cdot R^2. \tag{19}$$

Thus, P_{l1} is a special case of P_{l2} with $l_1 = l_0$, and P_{q2} is a special case of P_{q3} with $q1 = 0$.

Expected precision is defined as $EP := E[r(s,q)]/s$, expected recall as $ER := E[r(s,q)]/E(\text{rel}|q,DL)$.

So, when we assume a linear recall-precision function, we can estimate the number of relevant documents in a result set of s documents by

$$\frac{E[r(s,q)]}{s} = EP = P(ER) = l_0 - l_1 \cdot \frac{E[r(s,q)]}{E(\text{rel}|q,DL)}, \tag{20}$$

$$E[r(s,q)] := \frac{l_0 \cdot E(\text{rel}|q,DL) \cdot s}{E(\text{rel}|q,DL) + l_1 \cdot s}. \tag{21}$$

When we assume a quadratic recall-precision function, we have to solve the quadratic equation:

$$\frac{q_2}{E(\text{rel}|q,DL)^2} \cdot E[r(s,q)]^2 + \left(\frac{1}{s} - \frac{q_1}{E(\text{rel}|q,DL)^2} \right) E[r(s,q)] - q_0 = 0. \tag{22}$$

Thus, we can compute potential values for $E[r(s,q)]$ as:

$$p := \frac{E(\text{rel}|q,DL)^2}{q_2 \cdot s} - \frac{q_1}{q_2} \cdot E(\text{rel}|q,DL)^2, \tag{23}$$

$$q := \frac{q_0}{q_2} \cdot E(\text{rel}|q,DL)^2, \tag{24}$$

$$E[r(s,q)] \;=\; -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}. \tag{25}$$

5 Combining CORI and DTF

In this section we introduce a new method for estimating retrieval quality, called DTF-cori in the remainder of this paper. DTF-cori is similar to DTF-rp insofar as it also employs a recall-precision function, but here we estimate the number of relevant documents in the DL based on the CORI scores instead of formula 14.

CORI computes a ranking of libraries, based on DL scores $Pr(q|DL)$. Our basic assumption is that this score is related to the quality of the library. This is reasonable as the scores are used for ranking the libraries, and the system should favour high-quality DLs (w. r. t. the given query).

We are mainly interested in libraries containing many relevant documents in the top ranks. In Sect. 4.3 we presented DTF-rp, a technique for estimating the number of relevant documents in the top ranks based on the number $E(rel|q,DL)$ of relevant documents in the whole library. We can use this method by estimating $E(rel|q,DL)$ based on the DL score $Pr(q|DL)$ computed by CORI.

This is very similar to the case of single documents discussed in [10]. A retrieval engine computes a document "score" (called retrieval status value, RSV for short), and transforms it into the probability that this document is relevant. The relationship between score and probability of relevance is approximated by a mapping function.

In our setting, the retrieval engine is CORI, operating on "meta-documents" (the concatenation of all documents in a library). So, CORI computes a RSV $Pr(q|DL)$ for a meta-document DL, which has to be transformed into the probability $Pr(\text{rel}|q,DL)$ that the DL is relevant (i.e., the average probability of relevance in that DL). Then, the expected number of relevant documents can easily be computed as:

$$E(rel|q,DL) = |DL| \cdot Pr(\text{rel}|q,DL). \tag{26}$$

Similar to a mapping function, we introduce a "quality estimation function" which maps the CORI score $Pr(q|DL)$ of the DL onto the average probability of relevance in that DL, $Pr(\text{rel}|q,DL)$:

$$f' : [0,1] \mapsto \mathbb{R}, \quad f'(Pr(q|DL)) \approx Pr(\text{rel}|q,DL). \tag{27}$$

In this paper, we start with a linear estimator (DTF-cori-lin). If we assume that the number of relevant documents in a library is proportional to the DL score, we arrive at a linear function. We can add one degree of freedom by using a constant part:

$$f'_{lin}(x) := c'_0 + c'_1 \cdot x. \tag{28}$$

We also investigate the use of logistic functions (DTF-cori-log). These functions perform well on the document level [10]:

$$f'_{log}(x) := \frac{\exp(b'_0 + b'_1 x)}{1 + \exp(b'_0 + b'_1 x)}. \tag{29}$$

As for the functions mapping document RSVs onto probabilities of relevance, these parameters are query-specific in general. However, as the number of relevant documents is unknown in advance, we only can learn DL-specific parameters.

6 Evaluation

This section describes our detailed evaluation of the decision-theoretic framework and its comparison with CORI.

6.1 Experimental Setup

As in previous resource selection evaluations, we used the TREC-123 test bed with the CMU 100 library split [1]. The libraries are of roughly the same size (about 33 megabytes), but vary in the number of documents they contain (from 752 to 33 723, the average is 10 782). The documents inside a library are from the same source and the same time-frame. All samples contain 300 documents.

We used the same document indexing terms and query terms (after stemming and stop word removal) for both CORI and the three DTF variants. The document index only contains the `<text>` sections of the documents. Queries are based on TREC topics 51–100 and 101–150 [8], respectively. We used three different sets of queries: short queries (`<title>` field, on average 3.3 terms, web search), mid-length queries (`<description>` field, on average 9.9 terms, advanced searchers) and long queries (all fields, on average 87.5 terms, common in TREC-based evaluations).

The standard weighting schemes for documents and queries are used for the CORI experiments. For the DTF experiments, a modified BM25 weighting scheme [12] is employed for documents:

$$P(t \leftarrow d) := \frac{tf(t,d)}{tf(t,d) + 0.5 + 1.5 \cdot \frac{dl(d)}{avgdl}} \cdot \frac{\log \frac{numdl}{df(t)}}{\log |DL|}. \tag{30}$$

Here, $tf(t,d)$ is the term frequency, $dl(d)$ denotes the document length (in terms), *avgdl* the average document length, *numdl* the sample or library size (number of documents), $|DL|$ the library size, and $df(t)$ the document frequency. 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 be regarded as a probability.

Normalised tf values are used as query term weights:

$$P(q \leftarrow t) := \frac{tf(t,q)}{ql(q)} \quad . \tag{31}$$

Here, $tf(t,q)$ denotes the term frequency, and $ql(q) := \sum_{t \in q} tf(t,q)$ is the query length.

For DTF, we applied the same indexing and retrieval methods for the 100 libraries as we used for the resource selection index. We always requested 300 documents. For CORI, we employed the Lemur toolkit implementation[1] and selected 10 libraries (with 30 documents per selected DL) as in previous evaluations of Lemur[2]

The variants of the decision-theoretic framework (DTF-cori and DTF-rp) require a learning phase for bridging heterogeneous collections. The parameters are learned using a cross-evaluation strategy: Parameters are learned on TREC topics 51–100 and evaluated on topics 101–150, and vice versa. We used the Gnuplot[3] implementation of the nonlinear least-squares (NLLS) Marquardt-Levenberg algorithm [11] and the relevance judgements as probabilities of relevance for learning the parameters. As we don't have relevance judgements for all documents in practice, we only considered the 100 top-ranked documents.

6.2 Result quality

The precision in the top ranks 5, 10, 15, 20 and 30 (averaged over all 100 topics) is depicted in Tab. 1–4, as well as the average precision.

The percentage values denote the difference to CORI; differences which are significant (assuming a t-Test with p=0.05) are marked with an asterisk.

When we assume a P_{l1} recall-precision function (linear function with one variable), then DTF-cori-lin outperforms DTF-cori-log, and both yield a better quality than CORI and DTF-rp in most cases. As already reported [9], DTF-rp also outperforms CORI in most cases.

Average precision for both DTF-cori variants is always higher than for CORI and DTF-rp. The difference is significant for DTF-cori-lin for all query types and for DTF-cori-log for all except short queries.

When we add a second degree of freedom, then DTF-cori-lin and DTF-cori-log still outperform DTF-rp (using the same recall-precision function), but quality significantly decreases compared to CORI. DTF-cori-lin is slightly better than DTF-cori-log. These results are surprising: In principle, adding one degree of freedom should increase the quality, as P_{l1} is a special case of P_{l2}. However, it seems that the parameters for P_{l2} fitted too much to the learning data ("overfitting").

When we employ a P_{q2} quadratic recall-precision function with 2 variables (i.e., it is monotonically decreasing), the DTF-cori quality is only slightly better compared to P_{l2}, but is still dramatically (and, in most cases, also significantly) worse than CORI.

Finally, we evaluated all 3 DTF methods with a quadratic recall-precision function with 3 variables. For all DLs and query types, the system learned a quadratic function $q_0 + q_1 \cdot x - q_2 \cdot x^2$ with $q_2 < 0$. The results w. r. t. precision in the top ranks are heterogeneous: For short queries, both DTF-cori variants perform worse than CORI in the lower ranks and better in the higher ranks. For mid-length and long queries, DTF-cori outperforms CORI. Average precision of DTF-cori is better (except for short queries,

[1] `http://www-2.cs.cmu.edu/~lemur/`

[2] The optimal constant number of selected libraries never has been evaluated for Lemur.

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

(a) Learned/evaluated on short queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.4260 / +0.0%	0.3940 / -7.5%	0.3940 / -7.5%	0.4020 / -5.6%
10	0.3930 / +0.0%	0.3880 / -1.3%	0.3840 / -2.3%	0.3820 / -2.8%
15	0.3840 / +0.0%	0.3853 / +0.3%	0.3820 / -0.5%	0.3767 / -1.9%
20	0.3640 / +0.0%	0.3765 / +3.4%	0.3745 / +2.9%	0.3665 / +0.7%
30	0.3487 / +0.0%	0.3593 / +3.0%	0.3583 / +2.8%	0.3393 / -2.7%
Avg.	0.0517 / +0.0%	0.0730 / +41.2% *	0.0723 / +39.8%	0.0616 / +19.1%

(b) Learned/evaluated on mid queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.3840 / +0.0%	0.4380 / +14.1%	0.4400 / +14.6%	0.4140 / +7.8%
10	0.3630 / +0.0%	0.4140 / +14.0%	0.4140 / +14.0%	0.3980 / +9.6%
15	0.3500 / +0.0%	0.4067 / +16.2%	0.4087 / +16.8%	0.3820 / +9.1%
20	0.3350 / +0.0%	0.3945 / +17.8%	0.3950 / +17.9%	0.3710 / +10.7%
30	0.3107 / +0.0%	0.3710 / +19.4%	0.3710 / +19.4%	0.3460 / +11.4%
Avg.	0.0437 / +0.0%	0.0716 / +63.8% *	0.0716 / +63.8% *	0.0509 / +16.5%

(c) Learned/evaluated on long queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.5780 / +0.0%	0.5820 / +0.7%	0.5680 / -1.7%	0.5680 / -1.7%
10	0.5590 / +0.0%	0.5660 / +1.3%	0.5510 / -1.4%	0.5570 / -0.4%
15	0.5340 / +0.0%	0.5587 / +4.6%	0.5420 / +1.5%	0.5387 / +0.9%
20	0.5175 / +0.0%	0.5500 / +6.3%	0.5440 / +5.1%	0.5335 / +3.1%
30	0.5013 / +0.0%	0.5403 / +7.8%	0.5313 / +6.0%	0.5160 / +2.9%
Avg.	0.0883 / +0.0%	0.1371 / +55.3% *	0.1315 / +48.9% *	0.1029 / +16.5%

Table 1. Precision in top ranks and average precision, l1

significantly better) than for CORI. In all cases, DTF-cori performs slightly worse than in the case of a linear recall-precision function with 1 variable.

These results are also reflected in the corresponding recall-precision plots (which we leave out due to space restrictions).

6.3 Overall retrieval costs

Actual costs for retrieval (of 300 documents) and the number of selected DLs are shown in Tab. 5. Costs only refer to retrieval quality:

$$C_i(s_i, q) = s_i - r(s_i, q). \tag{32}$$

With a few exceptions, especially for the linear recall-precision function with two parameters, the costs for DTF-cori are lower than for CORI; in all cases, they are lower than DTF-rp.

On the other hand, DTF-cori selects a lot more DLs than CORI (always 10 DLs) and DTF-rp do; the number of selected DLs is maximal for the linear recall-precision function with one variable.

(a) Learned/evaluated on short queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.4260 / +0.0%	0.2820 / -33.8% *	0.2800 / -34.3% *	0.2400 / -43.7% *
10	0.3930 / +0.0%	0.2380 / -39.4% *	0.2350 / -40.2% *	0.1950 / -50.4% *
15	0.3840 / +0.0%	0.2027 / -47.2% *	0.1933 / -49.7% *	0.1627 / -57.6% *
20	0.3640 / +0.0%	0.1810 / -50.3% *	0.1730 / -52.5% *	0.1415 / -61.1% *
30	0.3487 / +0.0%	0.1477 / -57.6% *	0.1393 / -60.1% *	0.1187 / -66.0% *
Avg.	0.0517 / +0.0%	0.0085 / -83.6% *	0.0079 / -84.7% *	0.0097 / -81.2% *

(b) Learned/evaluated on mid queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.3840 / +0.0%	0.2340 / -39.1% *	0.2060 / -46.4% *	0.1580 / -58.9% *
10	0.3630 / +0.0%	0.1980 / -45.5% *	0.1600 / -55.9% *	0.1190 / -67.2% *
15	0.3500 / +0.0%	0.1753 / -49.9% *	0.1380 / -60.6% *	0.0960 / -72.6% *
20	0.3350 / +0.0%	0.1530 / -54.3% *	0.1175 / -64.9% *	0.0775 / -76.9% *
30	0.3107 / +0.0%	0.1283 / -58.7% *	0.0920 / -70.4% *	0.0603 / -80.6% *
Avg.	0.0437 / +0.0%	0.0088 / -79.9% *	0.0042 / -90.4% *	0.0025 / -94.3% *

(c) Learned/evaluated on long queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.5780 / +0.0%	0.4960 / -14.2%	0.5280 / -8.7%	0.2020 / -65.1% *
10	0.5590 / +0.0%	0.5000 / -10.6%	0.5140 / -8.1%	0.1550 / -72.3% *
15	0.5340 / +0.0%	0.4847 / -9.2%	0.4993 / -6.5%	0.1260 / -76.4% *
20	0.5175 / +0.0%	0.4715 / -8.9%	0.4735 / -8.5%	0.1070 / -79.3% *
30	0.5013 / +0.0%	0.4497 / -10.3%	0.4467 / -10.9%	0.0860 / -82.8% *
Avg.	0.0883 / +0.0%	0.0811 / -8.2%	0.0637 / -27.9% *	0.0043 / -95.1% *

Table 2. Precision in top ranks and average precision, l2

6.4 Approximation quality

The mean square approximation error (linear recall-precision function with one variable) is depicted in Tab. 6. One can see that the linear estimator generates a significantly better approximation than DTF-cori-log and DTF-rp, where the latter one always heavily overestimates the number of relevant documents in the collection.

6.5 Evaluation summary

From a theoretical point of view, integrating CORI into DTF has the advantage that other cost sources besides retrieval quality (e.g. time or money) can easily be incorporated. The evaluation results we reported in this section show that it also allows for better resource selections (on a theoretically founded basis) compared to the heuristic selection strategy of CORI (“select the 10 DLs with the highest scores and retrieve an equal amount of documents from each of these 10 DLs”). Precision both in the top ranks and on average is maximised by using DTF-cori-lin with a linear approximation (1 parameter) of the recall-precision function.

(a) Learned/evaluated on short queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.4260 / +0.0%	0.2808 / -34.1% *	0.2404 / -43.6% *	0.3060 / -28.2% *
10	0.3930 / +0.0%	0.2838 / -27.8% *	0.2384 / -39.3% *	0.2930 / -25.4% *
15	0.3840 / +0.0%	0.2761 / -28.1% *	0.2290 / -40.4% *	0.2753 / -28.3% *
20	0.3640 / +0.0%	0.2667 / -26.7% *	0.2177 / -40.2% *	0.2560 / -29.7% *
30	0.3487 / +0.0%	0.2478 / -28.9% *	0.2007 / -42.4% *	0.2330 / -33.2% *
Avg.	0.0517 / +0.0%	0.0442 / -14.5%	0.0323 / -37.5% *	0.0357 / -30.9%

(b) Learned/evaluated on mid queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.3840 / +0.0%	0.2820 / -26.6% *	0.2820 / -26.6% *	0.2300 / -40.1% *
10	0.3630 / +0.0%	0.2590 / -28.7% *	0.2620 / -27.8% *	0.2020 / -44.4% *
15	0.3500 / +0.0%	0.2447 / -30.1% *	0.2453 / -29.9% *	0.1840 / -47.4% *
20	0.3350 / +0.0%	0.2320 / -30.7% *	0.2310 / -31.0% *	0.1700 / -49.3% *
30	0.3107 / +0.0%	0.2170 / -30.2% *	0.2160 / -30.5% *	0.1657 / -46.7% *
Avg.	0.0437 / +0.0%	0.0341 / -22.0%	0.0336 / -23.1%	0.0161 / -63.2% *

(c) Learned/evaluated on long queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.5780 / +0.0%	0.5020 / -13.1%	0.3580 / -38.1% *	0.3120 / -46.0% *
10	0.5590 / +0.0%	0.4750 / -15.0%	0.3280 / -41.3% *	0.3030 / -45.8% *
15	0.5340 / +0.0%	0.4660 / -12.7%	0.3220 / -39.7% *	0.2907 / -45.6% *
20	0.5175 / +0.0%	0.4525 / -12.6%	0.3140 / -39.3% *	0.2815 / -45.6% *
30	0.5013 / +0.0%	0.4420 / -11.8%	0.2980 / -40.6% *	0.2627 / -47.6% *
Avg.	0.0883 / +0.0%	0.1143 / +29.4%	0.0683 / -22.7%	0.0337 / -61.8% *

Table 3. Precision in top ranks and average precision, q2

7 Conclusion and outlook

In this paper, we combined the decision-theoretic framework [6, 9] with CORI [3]. DTF has a better theoretic foundation (selection with minimum costs) than traditional resource ranking algorithms like CORI, considers additional cost sources like time and money, and computes the number of digital libraries to be queried as well as the number of documents which should be retrieved from each of these libraries. In contrast, heuristic methods like CORI compute a ranking of digital libraries, and additional heuristics are needed for determining the number of libraries and the number of documents to be retrieved. The retrieval quality of DTF is competitive with CORI.

Our new approach DTF-cori combines DTF and CORI. It first computes library scores with CORI which specify the similarity between the library and the query. This score is then mapped onto the expected number of relevant documents in the complete DL. We investigated the use of a linear and a logistic "estimation function" for this mapping. Then, the estimates of the number of relevant documents in the DL are used together with a recall-precision function (as for the DTF-rp variant) for approximating the number of relevant documents in the result set of given size. In this paper, we considered four different recall-precision functions: a linear one with one and two variables, and a quadratic one with two and three variables.

(a) Learned/evaluated on short queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.4260 / +0.0%	0.3860 / -9.4%	0.3820 / -10.3%	0.3660 / -14.1%
10	0.3930 / +0.0%	0.3760 / -4.3%	0.3740 / -4.8%	0.3350 / -14.8%
15	0.3840 / +0.0%	0.3753 / -2.3%	0.3727 / -2.9%	0.3160 / -17.7%
20	0.3640 / +0.0%	0.3695 / +1.5%	0.3655 / +0.4%	0.3050 / -16.2%
30	0.3487 / +0.0%	0.3507 / +0.6%	0.3497 / +0.3%	0.2807 / -19.5%
Avg.	0.0517 / +0.0%	0.0675 / +30.6%	0.0663 / +28.2%	0.0379 / -26.7%

(b) Learned/evaluated on mid queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.3840 / +0.0%	0.4200 / +9.4%	0.4200 / +9.4%	0.3380 / -12.0%
10	0.3630 / +0.0%	0.3960 / +9.1%	0.3960 / +9.1%	0.3010 / -17.1%
15	0.3500 / +0.0%	0.3927 / +12.2%	0.3940 / +12.6%	0.2760 / -21.1%
20	0.3350 / +0.0%	0.3775 / +12.7%	0.3795 / +13.3%	0.2585 / -22.8%
30	0.3107 / +0.0%	0.3647 / +17.4%	0.3650 / +17.5%	0.2290 / -26.3% *
Avg.	0.0437 / +0.0%	0.0650 / +48.7% *	0.0648 / +48.3% *	0.0208 / -52.4% *

(c) Learned/evaluated on long queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.5780 / +0.0%	0.5800 / +0.3%	0.5700 / -1.4%	0.4140 / -28.4% *
10	0.5590 / +0.0%	0.5660 / +1.3%	0.5610 / +0.4%	0.3940 / -29.5% *
15	0.5340 / +0.0%	0.5587 / +4.6%	0.5533 / +3.6%	0.3727 / -30.2% *
20	0.5175 / +0.0%	0.5485 / +6.0%	0.5480 / +5.9%	0.3435 / -33.6% *
30	0.5013 / +0.0%	0.5337 / +6.5%	0.5287 / +5.5%	0.2987 / -40.4% *
Avg.	0.0883 / +0.0%	0.1334 / +51.1% *	0.1315 / +48.9% *	0.0227 / -74.3% *

Table 4. Precision in top ranks and average precision, q3

This new technique has two advantages: First, it extends the range of applications of CORI. Together with DTF, now other cost sources like time and money can also be incorporated in a natural way.

Second, the evaluation showed that we can increase precision both in the top ranks and on average when we integrate CORI into DTF. This indicates that DTF-cori can compute a better selection than CORI alone. The best results were obtained when a primitive linear function with only one variable and a linear estimation function is used. However, the differences in precision in the top ranks are not significant (in contrast to most differences in average precision).

When more degrees of freedom are allowed, we observe the effect of overfitting of the parameters to the learning data for a linear and the quadratic recall-precision function with two parameters each. The quadratic recall-precision function with three degrees of freedoms performs only slightly worse.

DTF-cori-lin approximates the number of relevant documents in a DL better than DTF-cori-log and DTF-rp whose estimates are much to high. This is only partially reflected by the retrieval quality, as the differences are not as high as suggested by the approximation errors.

(a) Learned/evaluated on short queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
l1	245.7 / 10.0	237.3 / 69.2	238.3 / 70.1	239.0 / 40.5
l2	245.7 / 10.0	266.2 / 6.7	267.5 / 6.8	281.5 / 5.7
q2	245.7 / 10.0	248.7 / 30.3	241.9 / 25.8	256.0 / 15.6
q3	245.7 / 10.0	238.9 / 47.6	239.9 / 48.1	258.7 / 16.5

(b) Learned/evaluated on mid-length queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
l1	256.8 / 10.0	241.7 / 67.6	241.8 / 68.6	254.6 / 28.1
l2	256.8 / 10.0	291.5 / 9.0	294.0 / 6.1	297.1 / 4.3
q2	256.8 / 10.0	270.3 / 22.6	270.6 / 22.8	282.8 / 11.9
q3	256.8 / 10.0	244.6 / 44.5	244.8 / 44.3	278.9 / 9.9

(c) Learned/evaluated on long queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
l1	229.0 / 10.0	205.5 / 66.0	211.4 / 69.0	226.1 / 29.7
l2	229.0 / 10.0	250.0 / 30.8	261.6 / 32.2	296.6 / 3.8
q2	229.0 / 10.0	226.7 / 39.7	252.6 / 30.3	272.9 / 9.8
q3	229.0 / 10.0	209.4 / 51.6	211.2 / 52.6	280.5 / 6.6

Table 5. Actual costs and number of libraries selected

	DTF-cori-lin	DTF-cori-log	DTF-rp
short	110.49	123.81 / +12.1% *	140426.08 / >10^5% *
mid	96.33	122.62 / +27.3% *	527568.52 / >10^5% *
long	95.83	122.57 / +27.9% *	1585465.20 / >10^6% *

Table 6. Approximation error for number of relevant documents in DL

In the future, we will have a look at better estimation functions. We are particularly interested in improving the retrieval quality for shorter queries, because this query type is commonly issued by users (e.g. on the web).

In addition, we will investigate the learning step of the estimation function parameters in more detail. In this paper, we learned parameters with 50 queries. The interesting question is how many documents per query and how many queries are really required for obtaining good parameters, and how the quality of the parameters is related to the size of the learning data. A major goal is to avoid overfitting.

8 Acknowledgements

This work is supported in part by the EU commission (grant IST-2000-26061, MIND), and in part by the DFG (grant BIB47 DOuv 02-01, PEPPER).

References

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

[2] J. Callan, G. Cormack, C. Clarke, D. Hawking, and A. Smeaton, editors. *Proceedings of the 26st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, 2003. ACM.

[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] J. Callan, A. L. Powell, J. C. French, and M. Connell. The effects of query-based sampling on automatic database selection algorithms. *ACM Transactions on Information Systems (submitted for publication)*.

[5] J. French, A. Powell, J. Callan, C. Viles, T. Emmitt, K. Prey, and Y. Mou. Comparing the performance of database selection algorithms. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, pages 238–245, New York, 1999. ACM.

[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] D. Harman, editor. *The Second Text REtrieval Conference (TREC-2)*, Gaithersburg, Md. 20899, 1994. National Institute of Standards and Technology.

[9] H. Nottelmann and N. Fuhr. Evaluating different methods of estimating retrieval quality for resource selection. In Callan et al. [2].

[10] 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, Heidelberg et al., 2003. Springer.

[11] 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.

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

[13] L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In Callan et al. [2].

[14] L. Si, R. Jin, J. Callan, and P. Ogilvie. Language model framework for resource selection and results merging. In D. Grossman, editor, *Proceedings of the 11th International Conference on Information and Knowledge Management*, New York, 2002. ACM. `http://www-2.cs.cmu.edu/~callan/Papers/cikm02-lsi.pdf`.

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

[16] C. J. van Rijsbergen. Probabilistic retrieval revisited. *The Computer Journal*, 35(3):291–298, 1992.

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