Evaluating Different Methods of Estimating Retrieval Quality for Resource Selection

Henrik Nottelmann
nottelmann@uni-duisburg.de

Norbert Fuhr
fuhr@uni-duisburg.de

Institute of Informatics and Interactive Systems
University of Duisburg-Essen
47048 Duisburg, Germany

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. Most existing resource selection algorithms compute a library ranking in a heuristic way. In contrast, the decision-theoretic framework (DTF) follows a different approach on a better theoretic foundation: It computes a selection which minimises the overall costs (e.g. retrieval quality, time, money) of the distributed retrieval. For estimating retrieval quality the recall-precision function is proposed. In this paper, we introduce two new methods: The first one computes the empirical distribution of the probabilities of relevance from a small library sample, and assumes it to be representative for the whole library. The second method assumes that the indexing weights follow a normal distribution, leading to a normal distribution for the document scores. Furthermore, we present the first evaluation of DTF by comparing this theoretical approach with the heuristical state-of-the-art system CORI; here we find that DTF outperforms CORI in most cases.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models, selection process*

General Terms

Theory, Experimentation

Keywords

Resource selection, decision-theoretic framework, formal models, normal distribution, evaluation

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.

The simplest solution for the resource selection problem is that the user selects the libraries to search (e.g., from a list with all connected libraries). But in practice, a user has only limited knowledge about the content of the huge number of libraries available. Thus, the resulting selection quality is poor.

Recently several automatic selection methods have been proposed (see section 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.

These current resource selection approaches, including the state-of-the-art system CORI [3], have a good retrieval quality but a poor theoretical foundation. In contrast, the decision-theoretic framework (DTF) for resource selection described in [6] follows a theoretically founded approach: Every possible selection has assigned costs (including different selection criteria like retrieval quality, time or money), and the task is to find the selection with minimum costs. 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 specify the importance of the different cost sources by parameters. E.g., one user might prefer high quality results and is willing to spend time and money for this, whereas another user might prefer results without paying for them. Thus, a user can define her own retrieval policy.

Similar to the Probability Ranking Principle, DTF only characterises optimum results, but no algorithm for estimating costs. Here we introduce methods for estimating costs based on the most crucial cost source, retrieval quality. We start with a probabilistic retrieval model: we use probabilistic indexing weights, the document score is the probability that the document implies the query, and we estimate the probability that the document is relevant to a user.

In the following, we investigate three different, theoretically motivated methods for predicting retrieval quality (i.e., the number of relevant libraries in the result set):

1. The first method (the original one in [6]) estimates the total number of relevant documents in a library, and uses a recall-precision function for estimating the number of relevant documents in the result set.

2. The second, new method estimates the distribution of the probabilities of relevance by simulating retrieval on a small sample.

SIGIR'03, July 28–August 1, 2003, Toronto, Canada.

3. The third (also new) method aims at estimating the distribution of the probabilities of relevance from the distribution of the indexing weights. In particular, we assume a normal distribution for modelling the indexing weights.

The decision-theoretic framework has a sound theoretical foundation, but no-one ever evaluated its effectiveness. Thus, in this paper we also present the first evaluation of DTF. We compare the three different DTF methods with the best performing heuristical approach CORI (using the huge TREC-123 collection, split in 100 libraries). As CORI always selects a constant number of libraries and DTF does not, we also modified the latter for finding the optimum selection under the condition that always the same number of libraries are selected as for CORI (which, of course, leads to increased costs, and, potentially, to a reduced retrieval quality).

The rest of this paper is organised as follows. First, we describe some other resource selection algorithms, in particular CORI (which will be used as a baseline in our experiments). Then, we describe the decision-theoretic framework. In section 4, we propose two new methods for predicting retrieval quality. In section 5, we evaluate all three methods on a large test-bed and compare them with CORI.

2. RELATED WORK

In contrast to the decision-theoretic framework (DTF) considered in this paper, most of the other selection algorithms 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.

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 (in the experiments reported, 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 DTF's 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. Experiments showed that CORI outperforms GlOSS [5].

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 (i.e., number of queries), an accurate resource description can be constructed from samples of e.g. 300 documents. Although originally developed for CORI, it can be used for several other resource selection approaches (GlOSS, language models, DTF, but not for CVV [4]).

Very recently, the problem of estimating the number of documents in a library has been investigated. A sample-resample algorithm is proposed in [13]. Single-term queries, created from the library sample, are sent to the library, obtaining the number of matches, i.e. the document frequency of that term. Assuming that the document frequencies in the sample and the library are the same, the library size has to be be estimated. The estimation can be improved by repeating this process.

3. DECISION-THEORETIC FRAMEWORK

This section describes the decision-theoretic framework (DTF) for resource selection [6]. This model will be extended in section 4.

3.1 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 other cost factors (beside money) like time and quality as discussed below.

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 $\vec{s} = (s_1, s_2, \ldots, s_m)^T$ with $|\vec{s}| = \sum_{i=1}^{m} s_i = n$ which minimises the overall costs:

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

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

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

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).

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 very 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).

The task then is to minimise the expected overall costs, thus we have to replace formula 1 by

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

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.

3.2 Estimating retrieval quality

Resource selection accuracy in this model heavily depends on good estimations of the number of relevant documents $E[r_i(s_i,q)]$ in the result set of size s_i. In this subsection we describe the estimation described in [6] (called "DTF-rp" in the remainder of this paper). Two other methods will be presented in section 4. Evaluation results of all three methods are depicted in section 5.

All three methods follow Risjbergen'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 [16] for computing the probability of inference:

$$Pr(q \leftarrow d) = \sum_{t \in q} Pr(q \leftarrow t) \cdot Pr(t \leftarrow d) \quad . \tag{4}$$

We can map these probabilities of inference onto probabilities of relevance with a linear mapping function [15]

$$f : [0,1] \rightarrow [0,1], f(x) := Pr(\text{rel}|q \leftarrow d) \cdot x \tag{5}$$

with the document- and query-independent constant $Pr(\text{rel}|q \leftarrow d)$.

Let μ_t denote the average indexing weight of term t in the collection. Then, the expected number $E(\text{rel}|q,DL_i)$ of relevant documents in DL_i can be computed as:

$$E(\text{rel}|q,DL_i) = \sum_{d \in DL_i} Pr(\text{rel}|q,d) \tag{6}$$

$$= \quad |DL_i| \cdot Pr(\text{rel}|q \leftarrow d) \cdot \sum_{t \in q} Pr(q \leftarrow t_j) \cdot \mu_t. \tag{7}$$

The library size $|DL_i|$ can be set manually, or can be estimated with sampling-resampling [13].

Assuming a linearly decreasing recall-precision function

$$P_i : [0,1] \rightarrow [0,1], \; P(R) := P_i^0 \cdot (1-R), \tag{8}$$

with expected precision $E[r_i(s_i,q)]/s_i$ and expected recall $E[r_i(s_i,q)]/E(\text{rel}|q,DL_i)$, we can estimate the number of relevant documents when retrieving s_i documents[6]:

$$E[r_i(s_i,q)] := \frac{P_i^0 \cdot E(\text{rel}|q,DL_i) \cdot s}{E(\text{rel}|q,DL_i) + P_i^0 \cdot s}. \tag{9}$$

4. NEW METHODS FOR ESTIMATING RETRIEVAL QUALITY

In this section we propose two new 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.

4.1 Simulated retrieval on sample

Query-based sampling is used by CORI [3] to approximate statistical resource descriptions like average term frequencies $avgtf_t$ and inverse document frequencies df_t, and by DTF-rp it is used to approximate average indexing weights μ_t.

As the sample has to be acquired anyway for resource selection, it can also be indexed completely (i.e. all document-term pairs together with the indexing weights are stored in the resource description). In the resource selection phase of method "DTF-sample", retrieval is simulated with 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.

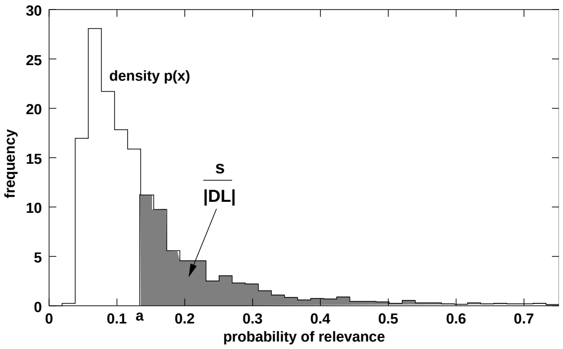

Figure 1: Density of probabilities of relevance and computation of $E[r(s,q)]$

Figure 1 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

$$s = |DL| \int_a^1 p(x) \; dx. \tag{10}$$

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:

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

This approach can be improved by using a logistic mapping function instead of a linear one [10]:

$$f : [0,1] \rightarrow [0,1], \; f(x) := \frac{\exp(b_0 + b_1 x)}{1 + \exp(b_0 + b_1 x)}. \tag{11}$$

Furthermore, early experiments showed that in most cases normalising the scores (with the maximum score in that DL, i.e. the score of the top-ranked document) yields a better performance:

$$Pr(\text{rel}|q,d) = f\left(\frac{Pr(q \leftarrow d)}{max_{d' \in DL} Pr(q \leftarrow d')}\right).$$

4.2 Modelling indexing weights by a normal distribution

Like the previous methods, we try to estimate the distribution of the probabilities of relevance $Pr(\text{rel}|q,d)$. Instead of a purely empirical approach (as before), here we develop a new theoretic model for the relationship between the desired distribution and the distribution of the indexing weights.

Our algorithm "DTF-normal" has four steps:

1. Modelling the distribution of the indexing weights $Pr(t \leftarrow d)$ (for a term t).

2. Computing—based on the indexing weight distribution—the distribution for the probabilities of inference $Pr(q \leftarrow d)$ (also called "scores").

3. Deriving—based on the score distribution and the mapping function f—the distribution of the probabilities of relevance $Pr(\text{rel}|q,d)$.

4. Estimating the number of relevant documents in the result set (as in DTF-sample).

In the first step we want to approximate the empirical, discrete distribution of the indexing weights in the collection by a simple, continuous distribution. It should be possible to express the density by a closed formula to ease computation of the score distribution.

We observed from early experiments that the indexing weight distribution can be approximated best by a normal distribution

$$p(x,\mu,\sigma) := \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \qquad (12)$$

which is defined by two parameters, the expectation μ and the variance σ.

One example is displayed in figure 2. Here, we left out a huge peak at zero for improved readability, corresponding to the large amount of documents which do not contain the term. Thus, the expectation of the indexing weights is close to zero, and the normal distribution density is positive also for negative values, although of course there are no negative indexing weights. We disregard this head of the distribution, as we are mainly interested in the high indexing weights (the tail of the distribution).

To derive the document score distribution in step 2, we can view the indexing weights of term t in all documents in a library as a random variable X_t. Then, the distribution of the scores of all documents in a library is modelled by the random variable

$$X := \sum_{i=1}^{l} a_i \cdot X_{t_i} \qquad (13)$$

with the l query terms t_i and weights $a_i := Pr(q \leftarrow t)$. As X_t follow a normal distribution, the linear combination X is also distributed normally, and the parameters can be computed efficiently:

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

Our assumption of normally distributed indexing weights and document scores contrasts the work presented in [9], where document scores are fitted using a normal distribution for the scores of the relevant documents and an exponential distribution for the non-relevant documents. About 60 relevant documents are required to compute the parameters, but in practice, libraries hardly ever contain that many relevant documents at all. Thus, we restrict our work to the case of a normal distribution. Our experiments support this assumption (see figure 3): A normal distribution is a good approximation for the tail of score distribution; however we believe that this approximations still can be improved.

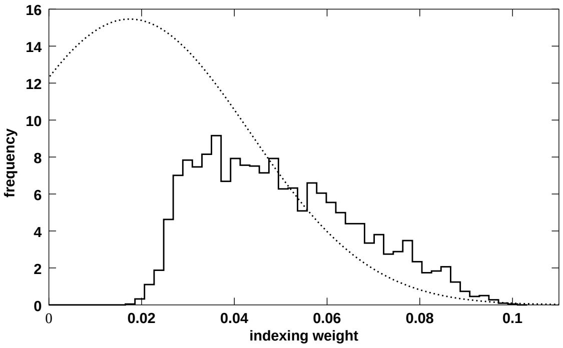

Figure 2: Indexing weight distribution, normal distribution fit

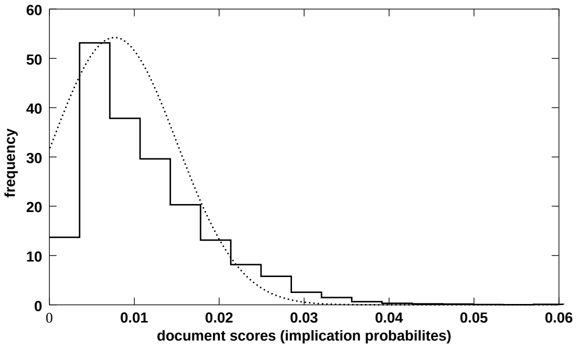

Figure 3: Score distribution, normal distribution fit

As for DTF-sample, using a logistic mapping function and normalising scores improves retrieval quality. Note that even with this modification, the normal distribution for the probabilities of inference $Pr(q \leftarrow d)$ is still valid.

5. EVALUATION

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

5.1 Experimental Setup

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. The documents inside a library are from the same source and the same time-frame. We took a constant number of 300 documents for each library. Table 1 depicts the summarised statistics for this 100 library test-bed.

(a) Complete libraries

	Minimum	Average	Maximum
Documents	752	10,782	33,723
Bytes	28,070,646	33,365,514	41,796,822

(b) Library samples

	Minimum	Average	Maximum
Documents	300	300	300
Bytes	229,915	2,701,449	15,917,750

Table 1: Summarised statistics for the 100 collection test-bed

We used the same document indexing terms and query terms for both CORI and the three DTF variants:

1. The document index only contains the `<text>` sections of the documents.
2. Queries are based on TREC topics 51–100 and 101–150 [8], respectively. We use three different sets of queries:

Short queries: only `<title>` field, between 1 and 7 terms (average 3.3), similar to those submitted to a web search engine.

Mid-length queries: only `<description>` field, between 4 and 19 terms (average 9.9), may be used by advanced searchers.

Long queries: all fields, between 39 and 185 terms (average 87.5), common in TREC-based evaluations.

For both documents and queries, terms are stemmed (using the Porter stemmer), and TREC stop words are removed.

The relevance judgements are the standard TREC relevance judgements [8], documents with no judgement are treated as irrelevant.

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:

$$P(t \leftarrow d) := \frac{tf(t,d)}{tf(t,d) + 0.5 + 1.5 \cdot \frac{dl(d)}{avgdl}} \cdot \frac{\log \frac{numdl}{df(t)}}{\log |DL|}. \tag{15}$$

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 linear factor $Pr(\text{rel}|q \leftarrow d)$ of the linear mapping function 5 and the linear factor b_1 of the logistic mapping function 11.

Normalised tf values are used as query term weights:

$$P(q \leftarrow t) := \frac{tf(t,q)}{ql(q)} \quad . \tag{16}$$

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

For the DTF experiments we used the same indexing and retrieval methods for the 100 libraries as we use for the resource selection index. We always requested 300 documents.

For CORI, we used two different variants:

CORI-all: Resource selection, retrieval on the selected libraries and merging the results are all performed by CORI. For each of the 100 libraries we built another index using the INQUERY retrieval engine [2].

CORI-rs: Resource selection is performed by CORI, but retrieval and result fusion (sorting w. r. t. the probabilities of relevance) is done with library implementations used by DTF.

In both cases the 10 top-ranked libraries are selected, and 30 documents are retrieved from each (resulting in 300 documents all together as for DTF).

With CORI-rs we can separate the effect of resource selection from the effects of different retrieval and result merging methods.

These experiments are run with the Lemur toolkit implementation of CORI.[1]

DTF-rp, DTF-sample and DTF-normal all require a learning phase:

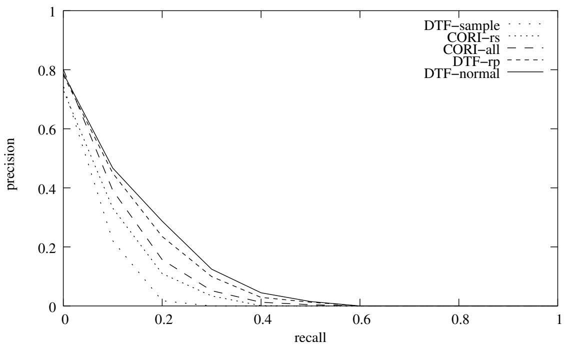

Figure 4: Recall-precision graph for optimum experiments, long queries

1. DTF-rp uses the starting point P^0 of the recall-precision function and the linear parameter $Pr(\text{rel}|q \leftarrow d)$ of the mapping function.

2. DTF-sample and DTF-normal need the two parameters b_0, b_1 of the logistic mapping function.

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[2] 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.

5.2 Optimum retrieval quality experiments

This first set of experiments is conducted to check the retrieval quality of DTF in contrast to CORI, the state-of-the-art representative of the current resource ranking systems. For optimum retrieval quality the DTF cost function only contains costs for retrieval quality ($C^+ = 0$ and $C^- = 1$), and the algorithms computes an optimum solution with a variable number of selected DLs (in contrast to CORI which always selects a fixed number of libraries).

For DTF, we learned parameters separately for short, mid-length and long queries, and applied them on the same query type (see section 5.4 for an evaluation of the sensitivity of the learned parameters w. r. t. query length).

The results are depicted in table 2. For all query types (short, mid-length and long queries), DTF-rp and DTF-normal perform about the same (DTF-sample performs bad all the time). Compared to CORI, the quality is better for mid-length and long queries (for short queries, only in some ranks). The improvement is maximal for mid-length queries, and competitive for short queries typically for the Web. For long queries (well-known from former CORI evaluations [5]), the improvement is better than for short queries but worse than for mid-length queries.

The recall-precision graph for long queries is depicted in figure 4; the graphs for the mid-length and the long queries are about the same and were left out due to space limitations. For long queries one can see that DTF-rp performs best, followed (in this order) by DTF-normal, CORI-all, CORI-rs and DTF-sample.

As CORI-all outperforms CORI-rs, the improvement of retrieval quality is only due to the resource selection method of DTF.

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

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

(a) Learned/evaluated on short queries

	CORI-all	CORI-rs	DTF-rp	DTF-sample	DTF-normal
5	0.4260 / +0.0%	0.4000 / -6.1%	0.4060 / -4.7%	0.3680 / -13.6%	0.3980 / -6.6%
10	0.3930 / +0.0%	0.3780 / -3.8%	0.3950 / +0.5%	0.3520 / -10.4%	0.3840 / -2.3%
15	0.3840 / +0.0%	0.3473 / -9.6%	0.3840 / +0.0%	0.3300 / -14.1%	0.3813 / -0.7%
20	0.3640 / +0.0%	0.3265 / -10.3%	0.3730 / +2.5%	0.3150 / -13.5%	0.3750 / +3.0%
30	0.3487 / +0.0%	0.3013 / -13.6%	0.3480 / -0.2%	0.2853 / -18.2%	0.3493 / +0.2%
Avg.	0.0517 / +0.0%	0.0428 / -17.2%	0.0662 / +28.0%	0.0349 / -32.5%	0.0645 / +24.8%

(b) Learned/evaluated on mid queries

	CORI-all	CORI-rs	DTF-rp	DTF-sample	DTF-normal
5	0.3840 / +0.0%	0.3520 / -8.3%	0.4440 / +15.6%	0.3840 / +0.0%	0.4480 / +16.7%
10	0.3630 / +0.0%	0.3450 / -5.0%	0.4190 / +15.4%	0.3580 / -1.4%	0.4220 / +16.3%
15	0.3500 / +0.0%	0.3300 / -5.7%	0.3987 / +13.9%	0.3327 / -4.9%	0.4087 / +16.8%
20	0.3350 / +0.0%	0.3220 / -3.9%	0.3830 / +14.3%	0.3160 / -5.7%	0.3980 / +18.8%
30	0.3107 / +0.0%	0.3017 / -2.9%	0.3627 / +16.7%	0.2900 / -6.7%	0.3747 / +20.6%
Avg.	0.0437 / +0.0%	0.0384 / -12.1%	0.0605 / +38.4%	0.0297 / -32.0%	0.0703 / +60.9%

(c) Learned/evaluated on long queries

	CORI-all	CORI-rs	DTF-rp	DTF-sample	DTF-normal
5	0.5780 / +0.0%	0.5360 / -7.3%	0.5880 / +1.7%	0.5200 / -10.0%	0.5780 / +0.0%
10	0.5590 / +0.0%	0.5250 / -6.1%	0.5680 / +1.6%	0.5130 / -8.2%	0.5690 / +1.8%
15	0.5340 / +0.0%	0.5187 / -2.9%	0.5440 / +1.9%	0.4927 / -7.7%	0.5587 / +4.6%
20	0.5175 / +0.0%	0.4970 / -4.0%	0.5400 / +4.3%	0.4605 / -11.0%	0.5450 / +5.3%
30	0.5013 / +0.0%	0.4723 / -5.8%	0.5233 / +4.4%	0.4260 / -15.0%	0.5323 / +6.2%
Avg.	0.0883 / +0.0%	0.0764 / -13.5%	0.1111 / +25.8%	0.0506 / -42.7%	0.1230 / +39.3%

Table 2: Precision in top ranks and average precision, optimum retrieval quality experiments

(a) Learned/evaluated on short queries

	CORI-all	CORI-rs	DTF-rp	DTF-sample	DTF-normal
5	0.4260 / +0.0%	0.4000 / -6.1%	0.3840 / -9.9%	0.3660 / -14.1%	0.3760 / -11.7%
10	0.3930 / +0.0%	0.3780 / -3.8%	0.3670 / -6.6%	0.3520 / -10.4%	0.3600 / -8.4%
15	0.3840 / +0.0%	0.3473 / -9.6%	0.3480 / -9.4%	0.3353 / -12.7%	0.3493 / -9.0%
20	0.3640 / +0.0%	0.3265 / -10.3%	0.3280 / -9.9%	0.3200 / -12.1%	0.3325 / -8.7%
30	0.3487 / +0.0%	0.3013 / -13.6%	0.3073 / -11.9%	0.2880 / -17.4%	0.3087 / -11.5%
Avg.	0.0517 / +0.0%	0.0428 / -17.2%	0.0410 / -20.7%	0.0353 / -31.7%	0.0401 / -22.4%

(b) Learned/evaluated on mid queries

	CORI-all	CORI-rs	DTF-rp	DTF-sample	DTF-normal
5	0.3840 / +0.0%	0.3520 / -8.3%	0.3900 / +1.6%	0.3860 / +0.5%	0.3940 / +2.6%
10	0.3630 / +0.0%	0.3450 / -5.0%	0.3710 / +2.2%	0.3700 / +1.9%	0.3730 / +2.8%
15	0.3500 / +0.0%	0.3300 / -5.7%	0.3500 / +0.0%	0.3453 / -1.3%	0.3533 / +0.9%
20	0.3350 / +0.0%	0.3220 / -3.9%	0.3360 / +0.3%	0.3210 / -4.2%	0.3365 / +0.4%
30	0.3107 / +0.0%	0.3017 / -2.9%	0.3103 / -0.1%	0.2933 / -5.6%	0.3107 / +0.0%
Avg.	0.0437 / +0.0%	0.0384 / -12.1%	0.0372 / -14.9%	0.0307 / -29.7%	0.0354 / -19.0%

(c) Learned/evaluated on long queries

	CORI-all	CORI-rs	DTF-rp	DTF-sample	DTF-normal
5	0.5780 / +0.0%	0.5360 / -7.3%	0.5420 / -6.2%	0.5320 / -8.0%	0.5480 / -5.2%
10	0.5590 / +0.0%	0.5250 / -6.1%	0.5140 / -8.1%	0.5220 / -6.6%	0.5300 / -5.2%
15	0.5340 / +0.0%	0.5187 / -2.9%	0.5007 / -6.2%	0.4967 / -7.0%	0.5033 / -5.7%
20	0.5175 / +0.0%	0.4970 / -4.0%	0.4815 / -7.0%	0.4705 / -9.1%	0.4845 / -6.4%
30	0.5013 / +0.0%	0.4723 / -5.8%	0.4477 / -10.7%	0.4323 / -13.8%	0.4490 / -10.4%
Avg.	0.0883 / +0.0%	0.0764 / -13.5%	0.0671 / -24.0%	0.0518 / -41.3%	0.0617 / -30.1%

Table 3: Precision in top ranks and average precision, fixed number of selected DLs (10 libraries)

(a) Learned/evaluated on short queries

	CORI-all	CORI-rs	DTF-rp	DTF-sample	DTF-normal
0.0	245.7 / 10.0	250.7 / 10.0	236.3 / 36.3	259.2 / 8.0	237.8 / 45.9
2.0	265.7 / 10.0	270.7 / 10.0	275.2 / 14.5	276.4 / 6.0	272.3 / 5.5
5.0	295.7 / 10.0	300.7 / 10.0	295.2 / 8.4	290.8 / 4.2	282.6 / 2.5

(b) Learned/evaluated on mid queries

	CORI-all	CORI-rs	DTF-rp	DTF-sample	DTF-normal
0.0	256.8 / 10.0	260.3 / 10.0	249.2 / 27.8	269.5 / 7.8	243.7 / 43.2
2.0	276.8 / 10.0	280.3 / 10.0	284.5 / 11.4	285.2 / 6.1	284.9 / 4.5
5.0	306.8 / 10.0	310.3 / 10.0	304.9 / 7.3	300.0 / 4.0	296.6 / 2.1

(c) Learned/evaluated on long queries

	CORI-all	CORI-rs	DTF-rp	DTF-sample	DTF-normal
0.0	229.0 / 10.0	235.6 / 10.0	222.1 / 28.2	255.1 / 8.2	215.7 / 37.4
2.0	249.0 / 10.0	255.6 / 10.0	265.4 / 11.1	273.8 / 6.5	266.6 / 8.4
5.0	279.0 / 10.0	285.6 / 10.0	287.4 / 7.1	290.6 / 4.7	286.9 / 3.5

Table 4: Actual costs and number of libraries selected (averaged over all 100 topics)

5.3 Fixed number of selected DLs

One of the differences between CORI and DTF is that CORI always selects a fixed number of libraries (10 in our experiments) whereas that number varies in DTF.

As we are interested in the effect of this difference, we modified the selection algorithm to perform an optimum selection under the condition that it always selects 10 libraries. Still the algorithm does not always select 30 documents per library, but in total 300 documents are retrieved from 10 libraries (as in CORI).

The experimental results are presented in table 3. Still DTF-rp and DTF-normal are very close together; both have about the same performance as CORI-all for mid-length queries only. However, for short and long queries all three DTF methods perform worse than CORI.

5.4 Sensitivity to query length

In the experiments presented so far we applied the parameters on the same query type we used for learning these parameters. The same will be done in practice.

However it is interesting how retrieval quality changes when we use different query types for learning and evaluation (of course, this only affects the DTF variants).

Due to space limitations we only discuss some qualitative results:

Short queries: Retrieval quality remains quite stable when we use short, mid-length or long queries for learning.

Mid-length queries: Quality significantly drops when we learn with short or long queries instead of mid-length queries.

Long queries: Using short queries for learning leads (not surprisingly) to a dramatic loss of precision in the top ranks (for all DTF variants with DTF-normal being the best DTF variant). Mid-length queries for learning lead to a better quality than learning with long queries for all three DTF variants for the top 15 ranks and a slightly worse quality for lower ranks and in average.

Concluding, the results depend on the query types for learning and evaluation, and parameters should be learned for queries which have about the same length as real queries have.

5.5 Additional costs

So far we only considered retrieval quality as cost factor. This is suitable for obtaining optimum retrieval quality but does not use the full strength of the decision-theoretic framework.

Now we use a modified cost function including other costs (linear in the number of documents retrieved) besides retrieval quality:

$$C_i(s_i, q) = c_0 + s_i - r(s_i, q). \qquad (17)$$

Thus, each selected library produces costs c_0, so total costs increase with the number of selected libraries, and the number of selected libraries can be decreased by increasing c_0. On the other hand, selecting more libraries can increase retrieval performance, thus decreasing the total costs.

Actual total costs (averaged over all 100 topics) and the average number of libraries selected can be found in table 4.

Here costs for all three DTF variants are close together, DTF-sample performs slightly worse than the other two methods. With $c_0 = 0$, DTF-sample always selects the smallest number of libraries (and has highest costs), as it estimates a worse retrieval quality compared to the other methods. DTF-normal selects more libraries than the other variants (also having smaller costs). With fixed costs $c_0 > 0$, slightly less libraries are selected by DTF-sample (with increased costs); for the other variants, the number of selected libraries decreases dramatically, but both methods still produce lower costs than DTF-sample.

Compared with CORI, costs for DTF-rp and DTF-normal are smaller than costs for CORI in most cases, even if they select less libraries than CORI (for higher fixed costs c_0).

Concluding, our best-performing variants DTF-rp and DTF-normal also produce lower costs than DTF-sample and the two CORI variants.

6. CONCLUSION AND OUTLOOK

In this paper, we extended the decision-theoretic framework described in [6]. It has a better theoretic foundation (selection with minimum costs) than traditional resource ranking algorithms, 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, other heuristic methods 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.

We introduced two new methods for predicting retrieval quality as the expected number of relevant documents in the library, using the distribution of the probabilities of relevance in a small sample as representative for the whole library (DTF-sample), and modelling the indexing weights with a normal distribution, leading to a normal distribution also for the document scores (the theoretically motivated method DTF-normal).

Furthermore we evaluated these two methods for estimating retrieval quality together with the original one (DTF-rp) on a large test-bed. We also compared the quality with CORI, the best performing heuristic resource selection approach.

These experiments show that DTF-rp and DTF-normal perform about the same (with DTF-normal being slightly better than DTF-rp), and that the quality is competitive with CORI. However, quality for all DTF variants drop below CORI when we fix the number of selected libraries (as CORI does). Further experiments point out that parameters should be learned on the queries whose length is comparable to the length of queries later issued to the system. The results of our experiments can be attributed to parameter learning phase as it allows DTF to adopt to the libraries.

We computed actual retrieval costs with different additional fixed costs. The results are inconsistent: In some cases, costs seem higher for CORI than for DTF-rp and DTF-normal, in other cases it is the other way round.

Concluding, in this paper we propose a class of resource selection algorithms whose performance is competitive with CORI. The major advantage is that they have a better theoretic foundation and integrate other cost sources besides retrieval quality.

In the future, we will improve the estimation of retrieval quality. In particular, we will continue studying the relationship between score $Pr(q \leftarrow d)$ and probability of relevance $Pr(\mathrm{rel}|q,d)$, and will investigate how the approximation of the indexing weights with a normal distribution can be improved. The major problem is that the normal distribution is a good approximation for the mid-range document scores, but for the highest scores (in which we are mainly interested) the approximation error has to be reduced.

Another interesting research direction will be to investigate the distribution of indexing values for other media types (e.g., images, facts like dates or names).

Furthermore, we plan to consider library overlaps as a fourth source (as described briefly in [17]). Once we have a good estimator for the number of duplicate documents, we can easily integrate this into our cost model.

7. ACKNOWLEDGEMENTS

This work is supported by the EU commission under grant IST-2000-26061 (MIND). Part of this work was performed while both authors where affiliated at the University of Dortmund. We wish to thank Paul Ogilvie and Luo Si from CMU for their assistance with and a bug-fix for their Lemur toolkit.

8. 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, W. Croft, and S. Harding. The INQUERY retrieval system. In *Proceedings of DEXA-92, 3rd International Conference on Database and Expert Systems Applications*, pages 78–83, Berlin et al., 1992. Springer.

[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, *Proceedings of the International Conference on Very Large Databases*, 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] R. Manmatha, T. Rath, and F. Feng. Modeling score distributions for combining the outputs of search engines. In W. Croft, D. Harper, D. Kraft, and J. Zobel, editors, *Proceedings of the 24th Annual International Conference on Research and development in Information Retrieval*, New York, 2001. ACM.

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

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

[14] L. Si, R. Jin, J. Callan, and P. Ogilvie. Proceedings of the 11th international conference on information and knowledge management. In D. Grossman, editor, *Proceedings of the 11th International Conference on Information and Knowledge Management*, New York, 2002. ACM.

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

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

[17] S. Wu and F. Crestani. Multi-objective resource selection in distributed information retrieval. In *Proceedings of The 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, 2002.