Optimum Polynomial Retrieval Functions Based on the Probability Ranking Principle

NORBERT FUHR
Technische Hochschule Darmstadt, Darmstadt, West Germany

We show that any approach to developing optimum retrieval functions is based on two kinds of assumptions: first, a certain form of representation for documents and requests, and second, additional simplifying assumptions that predefine the type of the retrieval function. Then we describe an approach for the development of optimum polynomial retrieval functions: request-document pairs (f_l, d_m) are mapped onto description vectors $\vec{x}(f_l, d_m)$, and a polynomial function $e(\vec{x})$ is developed such that it yields estimates of the probability of relevance $P(R \mid \vec{x}(f_l, d_m))$ with minimum square errors. We give experimental results for the application of this approach to documents with weighted indexing as well as to documents with complex representations. In contrast to other probabilistic models, our approach yields estimates of the actual probabilities, it can handle very complex representations of documents and requests, and it can be easily applied to multivalued relevance scales. On the other hand, this approach is not suited to log-linear probabilistic models and it needs large samples of relevance feedback data for its application.

Categories and Subject Descriptors: G.1.2 [**Numerical Analysis**]: Approximation—*least squares approximation*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*indexing methods*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models*

General Terms: Experimentation, Theory

Additional Keywords and Phrases: Complex document representation, linear retrieval functions, multivalued relevance scales, probabilistic indexing, probabilistic retrieval, probability ranking principle

1. INTRODUCTION

A major goal of IR research is the development of effective retrieval methods. Any effort in this direction is based (explicitly or implicitly) on a certain concept of optimum retrieval. In contrast to perfect retrieval (e.g., retrieve all relevant documents ahead of the first nonrelevant one), optimum retrieval is defined with respect to certain realistic restrictions concerning the retrieval process. Furthermore, some evaluation criteria must be given. Then optimum retrieval is the best retrieval (in terms of the evaluation criteria) that can be achieved while following predefined restrictions.

Authors' current address: Technische Hochschule Darmstadt, Fachbereich Informatik, Karolinenplatz 5, D-6100 Darmstadt, West Germany.

With the exception of a recent approach ([26], Sec. 4) only for probabilistic IR models, optimum retrieval has been defined precisely and the optimality has been proved theoretically: the "Probability Ranking Principle" (PRP) described in [21] says that optimum retrieval is achieved when documents are ranked according to decreasing values of their probability of relevance (with respect to the current request). For a binary relevance scale, the decision-theoretic justification of the PRP is as follows: Let R denote the event that a request-document pair (f_l, d_m) is judged relevant by the user. Furthermore, the costs for the retrieval of a nonrelevant document are denoted by C_1, and the costs for a retrieved relevant document by C_2. Now the decision-theoretic rule says that document d_m should be retrieved next from the collection if

$$C_2 \cdot \mathrm{P}(R \mid f_l, d_m) + C_1(1 - \mathrm{P}(R \mid f_l, d_m)) \leq C_2 \cdot \mathrm{P}(R \mid f_l, d_j) + C_1(1 - \mathrm{P}(R \mid f_l, d_j))$$

for any other document d_j in the collection (that has not been retrieved yet). In other terms: Retrieve that document for which the expected costs of retrieval are a minimum. Because $C_2 < C_1$, the above condition is equivalent to

$$\mathrm{P}(R \mid f_l, d_m) \geq \mathrm{P}(R \mid f_l, d_j).$$

So we have the rule that documents should be ranked according to decreasing values of their probability of relevance.

The explanation of the PRP given in [21] and later discussions of this topic lack a precise definition of the events the probabilities relate to. In our view, these probabilities are connected to the system's representations of documents and requests, not to the documents and requests themselves. For example, many retrieval models take sets of terms as representations of documents and queries. We denote a specific request with $\underline{f_l}$ and a specific document with $\underline{d_m}$, while f_l and d_m stand for the corresponding representations. This distinction also makes the difference between perfect and optimum retrieval clearer: Because of the representation, the system's knowledge about documents and requests is limited, and it cannot distinguish between different objects that have the same representation. Therefore, perfect retrieval is not possible for a real system. This statement is not in conflict with the observation that there are hardly ever two objects in existing collections that share the same representation: We regard collections as samples from possibly infinite sets of documents and requests, where there may be several (up to infinity) objects with the same representation. Because of the poor representation of retrieval objects that is actually in use (in comparison to the human understanding of documents and requests), only a probabilistic approach is adequate for these representations. Approaches that attempt to perform perfect retrieval for the current collection fail as soon as the next document or request is added to the collection.

Considering probabilistic IR models in more detail, it becomes obvious that these models do not really estimate probabilities for each specific representation: They use additional simplifying assumptions in order to approximate these probabilities. As an example, consider models that use sets of features (e.g., terms) as representations. In practice, it is impossible to estimate the correct

probability for a specific set of features because in most cases there is not any relevance information available for this set. Therefore, probabilistic models need additional independence assumptions; this way, probability estimates can be performed for single features (or pairs or triplets), and the probability for a set is computed by combining the probabilities of the elements.

These two kinds of approximations can be found with any probabilistic IR model: First is the choice of a certain form of representation for documents and requests. With respect to this representation, optimum retrieval (according to the PRP) can be defined. Second is the selection of an approximation method (e.g., certain independence assumptions).

For the least squares polynomial (LSP) approach described in this paper, only a framework is defined for both kinds of approximations, so an optimization with respect to a specific application can take place. This approach is based on the representation of request-document relationships in vector form $\vec{x}(f_l, d_m)$. Then a polynomial function $e(\vec{x})$ that yields estimates of $P(R \mid \vec{x})$ is developed. Both the mapping of a request-document relationship onto the so-called description vector and the class of the polynomial function (e.g., linear, quadratic) can be chosen.

The LSP approach is described in detail in the following section. Here we want to pinpoint the major features of this approach:

—In contrast to other probabilistic models, the LSP approach can cope with rather complex representations of request-document relationships (cf. Section 7).

—LSP retrieval functions do not only provide a probabilistic ranking of documents, they also give good estimates of the actual probability of relevance (cf. Section 2).

—The LSP approach can make use of multivalued relevance scales (cf. Section 3).

2. LEAST SQUARES POLYNOMIAL RETRIEVAL FUNCTIONS

The LSP approach originally was developed in the context of pattern recognition methods [25] as a refinement of general minimum square error procedures [6, pp. 151–159]. In [17] experiments with LSP, indexing functions are described and the application for the development of retrieval functions is proposed.

The task of the LSP approach is to classify objects into one of n classes. In the retrieval case, the objects are request-document pairs $(\underline{f}_l, \underline{d}_m)$, and the classification is the assignment of a value r_k from a relevance scale $R_n = (r_1, \ldots, r_n)$. (The case of multivalued relevance scales is discussed in more detail in the following section.) In order to develop an LSP retrieval function, a mapping of request-document relationships onto description vectors $\vec{x} = \vec{x}(f_l, d_m)$ has to be defined. Furthermore, a representative sample of request-document pairs together with their relevance judgements must be given. From this sample, a function is derived that yields for a pair $(\underline{f}_l, \underline{d}_m)$ estimates of the probabilities $P(r_k \mid \vec{x}(f_l, d_m))$, $k = 1, \ldots, n$.

In the following, the relevance judgement r_k for a pair (f_l, d_m) is represented by a vector $\vec{y} = (y_1, \ldots, y_n)$ with

$$y_i = \begin{cases} 1, & \text{if } i = k \\ 0 & \text{otherwise.} \end{cases}$$

Now we seek for a vector regression function $\vec{e}_{\text{opt}}(\vec{x})$, which yields optimum approximations $\hat{\vec{y}}$ of the class vectors $\vec{y}$. As the optimizing criterion we use the condition ($\mathrm{E}(\cdot)$ denotes the expectation):

$$\mathrm{E}(|\vec{y} - \vec{e}_{\text{opt}}(\vec{x})|^2) \stackrel{!}{=} \min. \tag{1}$$

With this condition, $\vec{e}_{\text{opt}}(\vec{x})$ yields probabilistic estimates $\mathrm{P}(r_k \mid \vec{x})$ in the components of $\hat{\vec{y}}$ (see [25, pp. 163–164]).

Equation (1) is the formulation of a general variation problem: Among all possible functions $\vec{e}(\vec{x})$, we seek for the optimum function fulfilling the above criterion. Because of the probabilistic nature of the approach, $\vec{e}_{\text{opt}}(\vec{x})$ is the optimum retrieval function (with respect to the chosen vector representation of request-document relationships) that follows the PRP.

However, this variation problem cannot be solved in its general form, so we have to restrict the search to a predefined class of functions. With this restriction, we will only find a suboptimum solution, but our general variation problem now becomes a parameter optimization task. The resulting functions yield least squares approximations of $\vec{e}_{\text{opt}}$: In [25, pp. 178–180], it is shown that an approximation with respect to the expression $\mathrm{E}(|\vec{y} - \hat{\vec{y}}|^2) \stackrel{!}{=} \min$ yields the same result as an optimization fulfilling the condition $\mathrm{E}(|\mathrm{E}(\vec{y} \mid \vec{x}) - \hat{\vec{y}}|^2) \stackrel{!}{=} \min$. So our restricted optimization process yields least squares approximations of the probabilities $\mathrm{P}(r_k \mid \vec{x}(f_l, d_m))$.

In the LSP approach, polynomials with a predefined structure are taken as function classes. For that, a polynomial structure $\vec{v}(\vec{x}) = (v_1, \ldots, v_L)$ has to be defined as

$$\vec{v}(\vec{x}) = (1, x_1, x_2, \ldots, x_N, x_1^2, x_1 x_2, \ldots)$$

where N is the number of dimensions of $\vec{x}$. The class of polynomials is given by the components $x_i^l \cdot x_j^m \cdot x_k^n \cdot \ldots$ $(i, j, k, \ldots \in [1, N]; l, m, n, \ldots \geq 0)$, which are to be included in the polynomial. In practice, mostly linear and quadratic polynomials are regarded.

The regression function now yields $\vec{e}(\vec{x}) = A^{\mathrm{T}} \cdot \vec{v}(\vec{x})$, where $A = (a_{ik})$ with $i = 1, \ldots, L$; $k = 1, \ldots, n$ is the coefficient matrix that has to be estimated (A^{T} denotes the transpose of A). So $\mathrm{P}(r_k \mid \vec{x})$ is approximated by the polynomial

$$\begin{aligned} e_k(\vec{x}) = {} & a_{1k} + a_{2k} \cdot x_1 + a_{3k} \cdot x_2 + \cdots + a_{N+1,k} \cdot x_N \\ & + a_{N+2,k} \cdot x_1^2 + a_{N+3,k} \cdot x_1 \cdot x_2 + \cdots \end{aligned}$$

The coefficient matrix is computed by solving the linear equation system [25, pp. 175–176]

$$\mathrm{E}(\vec{v} \cdot \vec{v}^{\mathrm{T}}) \cdot A = \mathrm{E}(\vec{v} \cdot \vec{y}). \tag{2}$$

The development of a polynomial retrieval function is performed in two steps:

(1) Statistical evaluation of a learning sample: A representative sample of request-document relationships together with relevance judgements must be given. From these relationships, the pairs $(\vec{x}, \vec{y})$ are derived, and then the empirical momental matrix M is computed:

$$M = (\overline{\vec{v} \cdot \vec{v}^T}\ \overline{\vec{v} \cdot \vec{y}^T}).$$

The matrix M contains both sides of the equation system (2).

(2) Computation of the coefficient matrix by means of the Gauss–Jordan algorithm [25, pp. 199–227]. For the iterative solution of the equation system, our algorithm always chooses the coefficient that maximizes the reduction of the overall error s^2. Let $M^{(i)} = (m_{kj}^{(i)})$ with $k = 1, \ldots, n; j = 1, \ldots, L + n$ denote the matrix M before the ith elimination step. Then the reduction $d_j^{(i)}$ that is achieved by choosing component j can be computed from the matrix [25, pp. 209–217] as follows:

$$d_j^{(i)} = \frac{1}{m_{jj}^{(i)2}} \cdot \sum_{k=L+1}^{L+n} m_{jk}^{(i)2}. \tag{3}$$

After each iteration, this procedure yields a preliminary solution: With the ith step, we get a polynomial function $\vec{e}^{(i)}(\vec{x})$ with i coefficients that can be regarded as the result of a correspondingly limited optimization process. This property is important in the case of small learning samples where a limited optimization might yield better retrieval results for other (test) samples than an unlimited one (see Section 6).

We illustrate our approach by giving an example: Assume that we have a binary relevance scale, and that request-document pairs are described by two-dimensional vectors $\vec{x} = (x_1, x_2)$. We define a linear polynomial structure of the form $\vec{v} = (1, x_1, x_2)$. Our learning sample consisting of 8 vectors is shown in Table I. The momental matrix for this sample is

$$M = \frac{1}{8} \cdot \begin{pmatrix} 8 & 5 & 6 & 4 & 4 \\ 5 & 5 & 3 & 3 & 2 \\ 6 & 3 & 6 & 3 & 3 \end{pmatrix}.$$

In order to select the first component to be eliminated, we compute the error reductions by applying eq. (3): $d_1^{(1)} = 0.50$, $d_2^{(1)} = 0.52$, and $d_3^{(1)} = 0.50$. So we have to choose the second component of $\vec{v}$, namely x_1, thus yielding the first polynomials $e_1^{(1)}(\vec{x}) = \frac{3}{5} x_1$ and $e_2^{(1)} = \frac{2}{5} x_1$.

The corresponding values of $e_1^{(1)}$ for the different vectors $\vec{x}$ are depicted in Table I. These values are already minimum square error approximations of $P(r_k \mid \vec{x})$.

Now the matrix $M^{(2)}$ has the value

$$M^{(2)} = \frac{1}{8} \cdot \begin{pmatrix} 3 & 0 & 3 & 1 & 2 \\ 5 & 5 & 3 & 3 & 2 \\ 3 & 0 & 4.2 & 1.2 & 1.8 \end{pmatrix}.$$

Table I. Example for the LSP Approach, Binary Relevance Scale

$\vec{x}$	r_k	$\vec{y}$	$P(r_1 \mid \vec{x})$	$e_1^{(1)}(\vec{x})$	$e_1^{(2)}(\vec{x})$	$e_1^{(3)}(\vec{x})$	$e_1^{(3)'}(\vec{x})$
(1, 1)	r_1	(1, 0)	0.67	0.6	0.60	0.67	0.69
(1, 1)	r_1	(1, 0)	0.67	0.6	0.60	0.67	0.69
(1, 1)	r_2	(0, 1)	0.67	0.6	0.60	0.67	0.69
(1, 0)	r_1	(1, 0)	0.50	0.6	0.60	0.50	0.46
(1, 0)	r_2	(0, 1)	0.50	0.6	0.60	0.50	0.46
(0, 1)	r_1	(1, 0)	0.33	0.0	0.33	0.33	0.31
(0, 1)	r_2	(0, 1)	0.33	0.0	0.33	0.33	0.31
(0, 1)	r_2	(0, 1)	0.33	0.0	0.33	0.33	0.31

For the selection of the next component, we get $d_1^{(2)} = 0.56$ and $d_3^{(2)} = 0.27$. Integrating v_1 into our polynomials we get the functions $e_1^{(2)} = 0.33 + 0.27x_1$ and $e_2^{(2)} = 0.67 - 0.27x_1$. Finally we have

$$M^{(3)} = \frac{1}{8} \cdot \begin{pmatrix} 3 & 0 & 3 & 1 & 2 \\ 5 & 5 & 3 & 3 & 2 \\ 0 & 0 & 1.2 & 0.2 & -0.2 \end{pmatrix},$$

$e_1^{(3)} = 0.17 + 0.33x_1 + 0.17x_2$, and $e_2^{(3)} = 0.83 - 0.33x_1 - 0.17x_2$.

As can be seen from the comparison of the values of $P(r_1 \mid \vec{x})$ and $e_1^{(3)}(\vec{x})$ in Table I, this function yields the correct probabilities. So we have $\vec{e}^{(3)}(\vec{x}) = \vec{e}_{opt}(\vec{x})$ in our example.

But now we add another vector $\vec{x} = (0, 0)$ with the relevance judgement r_2 to our learning sample, thus yielding the momental matrix

$$M' = \frac{1}{9} \cdot \begin{pmatrix} 9 & 5 & 6 & 4 & 5 \\ 5 & 5 & 3 & 3 & 2 \\ 6 & 3 & 6 & 3 & 3 \end{pmatrix}$$

and $e_1^{(3)'} = 0.08 + 0.38x_1 + 0.23x_2$. For our new vector, we get $e_1^{(3)'}((0, 0)) = 0.08$; the other values are listed in Table I. Obviously, now $\vec{e}^{(3)'} \neq \vec{e}_{opt'}$, that is, our linear function is only an approximation of $\vec{e}_{opt'}$ and thus also yields only approximate values of $P(r_k \mid \vec{x})$.

There are two important properties of the LSP retrieval functions:

(1) The basic assumptions of the LSP approach is that we can approximate the expectations by average values. In order to do this, learning samples of sufficient size are required. From previous experience with LSP applications in the IR context [9, 16], we can derive a rule of thumb for this size: There should be about 50–100 elements in the learning sample per component of the polynomial structure $\vec{v}$. With smaller learning samples, there may be parameter estimation problems. As a result, we would get a very effective retrieval function for the learning sample, which would perform significantly worse on other (test) samples. So the definition of $\vec{v}(\vec{x})$ should obey this relationship between vector length and learning sample size. For this reason, it seems to be inappropriate to develop request-specific retrieval functions. Instead, we define request-independent retrieval functions by mapping request-document pairs $(\underline{f}_l, \underline{d}_m)$ onto description vectors $\vec{x}(f_l, d_m)$.

(2) LSP retrieval functions yield estimates of the probability of relevance $P(r_k \mid \vec{x}(f_l, d_m))$. This feature makes the LSP approach very different from other probabilistic models where it is nearly impossible to get these estimates because there are too many parameters that can hardly be estimated. Therefore other probabilistic models only yield a request-specific ranking following the PRP (with the additional assumptions mentioned in Section 1), but no estimates of $P(R \mid f_l, d_m)$ can be computed. We think that estimates of the probability of relevance could help a user of an IR system get some impression of the overall quality of an answer.

3. LSP RETRIEVAL FUNCTIONS AND PROBABILISTIC MODELS

In this section, we first describe the development of LSP retrieval functions for a nonbinary relevance scale, then we discuss the relationship between LSP and log-linear models.

Bookstein [4] has shown how the PRP can be extended to nonbinary ordinal relevance scales: Assume that for the n relevance values with $r_1 < r_2 < \cdots < r_n$, the corresponding costs for the retrieval of a document with that relevance judgement are $C_1, C_2, \ldots, C_n$. Then documents should be ranked according to their expected costs

$$EC(f_l, d_m) = \sum_{k=1}^{n} C_k \cdot P(r_k \mid f_l, d_m).$$

Now we apply the LSP approach for the estimation of the probabilities $P(r_k \mid \vec{x}(f_l, d_m))$:

$$P(r_k \mid \vec{x}) \approx \sum_{j=1}^{L} a_{jk} \cdot v_j(\vec{x})$$

(where L is the number of dimensions of $\vec{v}(\vec{x})$). Then the expected costs can be approximated by

$$EC(\vec{x}) \approx \sum_{k=1}^{n} C_k \left(\sum_{j=1}^{L} a_{jk} \cdot v_j \right) = \sum_{j=1}^{L} \left(\sum_{k=1}^{n} C_k \cdot a_{jk} \right) v_j.$$

From the above expression, we can see that we only need a single polynomial instead of n polynomials for the different relevance classes. This single polynomial yields estimates of the expected costs, which is our ranking criterion. For the coefficients of this polynomial, the following relationship holds:

$$a_j' = \sum_{k=1}^{n} C_k \cdot a_{jk}.$$

In order to develop this kind of polynomial, we have to change the definition of the class vector $\vec{y}$: Now this vector has only one component to which we assign the appropriate cost value C_k for a request-document relationship with relevance judgement r_k. This way, the polynomial yields with $e(\vec{v}(\vec{x}))$ a least squares approximation of $E(y \mid \vec{x}) = EC(\vec{x})$.

Table II. Example With a Nonbinary Relevance Scale

$\vec{x}$	r_k	y	$EC(\vec{x})$	$e(\vec{x})$
(1, 1)	r_2	0.5	0.63	0.60
(1, 1)	r_1	0.7	0.63	0.60
(1, 1)	r_1	0.7	0.63	0.60
(1, 0)	r_3	0.3	0.40	0.45
(1, 0)	r_2	0.5	0.40	0.45
(0, 1)	r_4	0.0	0.33	0.36
(0, 1)	r_2	0.5	0.33	0.36
(0, 1)	r_2	0.5	0.33	0.36
(0, 0)	r_3	0.3	0.30	0.21

We can illustrate this procedure by assuming a nonbinary relevance scale for our example from the previous section (cf. Table II). As momental matrix M, we get

$$M = \frac{1}{9} \cdot \begin{pmatrix} 9 & 5 & 6 & 4.0 & 5.0 \\ 5 & 5 & 3 & 2.7 & 2.3 \\ 6 & 3 & 6 & 2.9 & 3.1 \end{pmatrix},$$

and our retrieval function is $e(\vec{x}) = 0.21 + 0.24x_1 + 0.16x_2$.

As shown in Table II, this function yields minimum square error approximations of $EC(\vec{x})$. Some experiments with this approach are described in Section 6.

In [4], there is also a discussion of the relationship between probabilistic ranking based on a binary relevance scale and a ranking according to descending values of a multivalued relevance scale. It is shown that under additional (reasonable) assumptions, both ranking methods yield the same results when the binary scale is derived from the multivalued scale by application of a cut-off value. We give some experimental results supporting this theoretic finding in Section 6.

Most probablistic ranking functions can be expressed in a log-linear form (e.g., [20], [22]) similar to

$$\log P(R \mid \vec{x}) = \sum_{i=1}^{N} a_i \log p_i$$

where the a_i's are some coefficients (mostly one of $\{-1, +1\}$), and the p_i's are single probabilities, products, or quotients of probabilities. This log-linear form seems to be very well suited for the application of the LSP approach in order to estimate coefficients (a_i) or probabilistic parameters ($\log p_i$).

For a more detailed discussion of this approach, let us make the simplifying assumption that the components of the description vector are assigned the values $x_i = \log p_i$. Furthermore, we only regard a single polynomial for the estimation of $P(R \mid \vec{x})$. Instead of the class value $y \in \{0, 1\}$, we would now have to take the value $\log y$, where we must define a suitable value for $\log 0$. With the LSP retrieval function, we would try to approximate

$$\log P(R \mid \vec{x}) = \log E(y \mid \vec{x}) \approx \vec{a}^T \cdot \vec{x}.$$

However, this approach cannot work, because the optimizing criterion for the LSP now is

$$E(|\log y - \vec{a}^T \cdot \vec{x}|^2) \stackrel{!}{=} \min,$$

from which we get a least squares approximation of $E(\log(y \mid \vec{x}))$, but

$$E(\log(y \mid \vec{x})) \neq \log E(y \mid \vec{x}).$$

So this kind of LSP retrieval function does not yield the desired estimates. This is a paradox: The favorable property of the LSP retrieval function that yields estimates of the probability of relevance is here in conflict with the probabilistic model. The probabilistic model suggests a log-linear form, whereas the LSP approach only can handle polynomials. However, there is an approximate solution, starting with the relationship

$$E(y \mid \vec{x}) \approx \exp(\vec{a}^T \cdot \vec{x}).$$

We can approximate the exponential function $\exp(u)$ by a polynomial for the form $\sum_{j=0}^{r} b_j \cdot u^j$. So we get the following polynomial retrieval function:

$$e(\vec{x}) = b_0 + b_1 \sum_{i_1=1}^{N} a_{i_1} x_{i_1} + b_2 \sum_{i_1=1}^{N} \sum_{i_2=i_1}^{N} a_{i_1} a_{i_2} x_{i_1} x_{i_2} + \cdots$$

$$+ b_r \sum_{i_1=1}^{N} \cdots \sum_{i_r=i_r-1}^{N} a_{i_1} \cdot \cdots \cdot a_{i_r} \cdot x_{i_1} \cdot \cdots \cdot x_{i_r}.$$

Since the LSP approach cannot take into account the relationship between the different coefficients following from the above equation, we finally end up with a general polynomial of the rth grade. An alternative method of developing log-linear functions by using maximum-likelihood estimates has been published recently [3]. We compare both approaches in a forthcoming paper.

4. OPTIMUM LINEAR RETRIEVAL

In [26], Wong et al., give an optimum linear retrieval function for the vector model [24]. Earlier work in this area has been done by Rocchio [23] who defined an optimum retrieval function based on the cosine similarity function. However, Rocchio could not give a theoretical proof for his approach.

Wong et al., formulate an "acceptable ranking strategy" for optimum retrieval. They assume that a user gives "preference relations" for document pairs (d, d') instead of relevance judgements for single documents. (The concept of preference relations is more powerful than that of relevance scales, but for reasons of simplicity we restrict the following discussion to a binary relevance scale.) The fact that a user prefers document d' over d is denoted as $d <\bullet d'$.

Now linear retrieval functions of the form $\vec{q}^T \cdot \vec{d}$, where $\vec{q}$ represents the current request and $\vec{d}$ a document are examined. Then the acceptable ranking strategy can be formulated: For any two pairs d, d' the following implication

should hold:

$$d <\bullet\ d' \Rightarrow \vec{q}^{\mathrm{T}} \cdot \vec{d} < \vec{q}^{\mathrm{T}} \cdot \vec{d}'.$$

This means that for any document pair for which the user has given a preference relation, the retrieval function should yield the correct ranking.

For the task of finding a vector $\vec{q}$ that fulfills the above criterion for a given sample of documents with preference relations, Wong et al., propose a gradient descent procedure. This procedure was originally developed for pattern recognition problems [6, pp. 138–147].

In our view, the major weakness of the approach described above is that it is well suited for retrospective retrieval where complete relevance information is available, but of limited value for the predictive case. Duda and Hart already stated "Of course, even if a separating vector is found for the design samples, it does not follow that the resulting classifier will perform well on independent test data" [6, p. 150].

A second problem with the "acceptable ranking strategy" occurs in the case where no optimum solution exists (e.g., when two documents with the same representation have different relevant judgements as in our examples in Section 2) because the gradient descent procedure does not converge in this case. For this problem, Duda and Hart proposed minimum squared error procedures, and these procedures are in fact preliminary versions of the LSP approach.

The latter problem deserves a more detailed discussion. When we assume that different documents might have the same representations, then the PRP gives us the optimum ranking strategy. The LSP approach, however, only gives us an approximate solution. Now the question is whether LSP retrieval functions are always optimum polynomial retrieval functions. In fact, if we return to the decision-theoretic foundation of the PRP, we can show that LSP retrieval functions might not be the optimum. To show this, we introduce additional cost factors C_1' and C_2' for the case when a nonrelevant/relevant document is not retrieved. With these cost factors, we can define a stopping point in retrieval [4]: Do not present any more documents to the user if the expected costs for the presentation of the next document d_m are higher than the expected costs when the document is not presented. From this rule, a cut-off value β for the probability of relevance can be derived:

$$\mathrm{P}(R \mid f_l, d_m) \geq \beta = \frac{C_1 - C_1'}{C_2' - C_2 + C_1 - C_1'}.$$

If we assume a situation where a user only wants a set of documents (e.g., as a result of an SDI run), then the estimation of the probability values near to this cut-off value becomes important. Because of the approximation that is performed by the LSP approach, there may be two sets of documents for which a wrong decision is made:

$$D_{\mathrm{sel}} = \{\underline{d}_m \mid \mathrm{P}(R \mid f_l, d_m) < \beta \quad \text{and} \quad e(\vec{x}(f_l, d_m)) \geq \beta\}.$$

$$D_{\mathrm{rej}} = \{\underline{d}_m \mid \mathrm{P}(R \mid f_l, d_m) \geq \beta \quad \text{and} \quad e(\vec{x}(f_l, d_m)) < \beta\}.$$

With the cost factors, we can compute the expected overall costs $EC(e(\vec{x}))$ of the errors produced by a specific polynomial retrieval function:

$$\begin{aligned} EC(e(\vec{x})) = & \sum_{d_m \in D_{sel}} P(R \mid f_l, d_m) \cdot (C_2 - C_2') \\ & + (1 - P(R \mid f_l, d_m)) \cdot (C_1 - C_1') \\ & - \sum_{d_m \in D_{rej}} P(R \mid f_l, d_m) \cdot (C_2 - C_2') \\ & + (1 - P(R \mid f_l, d_m)) \cdot (C_1 - C_1'). \end{aligned}$$

For this kind of application, the optimum polynomial retrieval function is the one that minimizes the value of the above expression. In contrast to this criterion, the LSP approach stands to get a good approximation for the whole range of probabilities. It is possible to modify the development process for an LSP retrieval function appropriately: In [16, pp. 183–186], some experiments of this kind (for the development of indexing functions) are described: A second, improved polynomial function $e'(\vec{x})$ is developed for those vectors $\vec{x}$ that yield values $e(\vec{x})$ near the cut-off value in the first iteration.

However, in most retrieval applications the cost factors are not known. So the overall approximation of the LSP approach seems to be adequate.

5. EXPERIMENTAL SETTING

For the experiments described in the following section, we took the collection from the AIR retrieval test [13] with the (Boolean) search formulations relating to controlled language terms (called descriptors here) only. First, a very broad Boolean search with binary indexing was performed. For that, we applied a small cut-off value of 0.01 to the probabilistic indexing called A1 in [13] which is based on an LSP indexing function. In the following, only the sets of output documents selected in this way are regarded. We compare the effect of different retrieval functions on the ranking of the documents within the output sets.

The 244 queries from the AIR retrieval test that had nonempty answer sets for the Boolean search were divided randomly into three samples named A, B, and C. For the development of the LSP retrieval functions, we used sample B as a learning sample and samples A and C as test samples. The number of queries and request-document pairs in each sample is shown in Table III. The multivalued scale that has been used for the relevance judgments and the distribution of these judgments in the output sets is depicted in Table IV.

Most of the experiments described in the following sections (unless stated otherwise) were performed using a binary relevance scale where all documents not judged "irrelevant" or "digressive" were treated as being relevant.

For evaluation, we use the normalized recall measure as defined in [2] for multivalued-relevance scales. This measure only considers documents in different ranks and with different relevance judgments. A pair of these documents is in

Table III. Number of Requests and Request-Document Pairs in the Different Samples

	Total	A	B	C
Number of requests	244	79	82	83
Number of request-document pairs	8,476	2,835	2,822	2,819

Table IV. Distribution of Relevance Judgments in the Output Sets

Judgment		Pairs with Judgment		Sample		
		Number	%	A	B	C
relevant	r_6	2,258	(27%)	732	776	750
conditionally relevant/more relevant	r_5	729	(9%)	200	217	312
conditionally relevant	r_4	440	(5%)	135	160	145
conditionally relevant/more irrelevant	r_3	716	(8%)	249	241	226
irrelevant	r_2	3,489	(41%)	1,204	1,057	1,228
digressive	r_1	844	(10%)	315	371	158
Total		8,476	(100%)	2,835	2,822	2,819

the right order if the document with the higher relevance judgment comes first; otherwise it is in the wrong order. Let S^+ be the number of document pairs in the right order, S^- the number of those in the wrong order, and $S^+_{\max}$ the number of documents in the right order for an optimum ranking. The normalized recall is then defined as follows:

$$R_{norm} = \frac{1}{2}\left(1 + \frac{S^+ - S^-}{S^+_{\max}}\right).$$

A random ordering of documents will have a R_{norm} value of 0.5 in the average. For the cases with $S^+_{\max} = 0$ we defined $R_{norm} = 1$. Because of the large scattering of the answer sizes, we use two average methods besides the macro average (arithmetic mean) R^M_{norm}:

(1) the micro-macro average R^m_{norm} is a weighted average with respect to the answer sizes. Let n_i be the answer size of retrieval result Δ_i; then the micro-macro average of R_{norm} for a set of t queries is defined as

$$R^m_{norm}(\Delta_1, \ldots, \Delta_t) = \frac{\sum_{i=1}^{t} n_i R_{norm(\Delta i)}}{\sum_{i=1}^{t} n_i}.$$

(2) the micro average R^μ_{norm} is computed by first combining the rank orders of different queries and then computing the R_{norm} measure for this combined rank order. There are several methods of combining rank orders for the computation of micro measures [15]. Here we use the retrieval status value ζ computed by the retrieval function as criterion.

Table V. Example Retrieval Result

ζ	Δ_1	Δ_2	Δ_2'
0.8	r_5	r_3	
0.6	r_1	r_1	r_3
0.4	r_3		r_1

For the illustration of the different averaging methods, we show an example retrieval result in Table V. The values of the R_{norm} measures are given below.

$$R_{norm}(\Delta_1) = \frac{2}{3}, \qquad R_{norm}(\Delta_2) = 1, \qquad R_{norm}(\Delta_2') = 1,$$

$$R^M_{norm}(\Delta_1, \Delta_2) = \frac{1}{2}\left(\frac{2}{3} + 1\right) = \frac{5}{6} = R^M_{norm}(\Delta_1, \Delta_2'),$$

$$R^m_{norm}(\Delta_1, \Delta_2) = \frac{1}{5}\left(3 \cdot \frac{2}{3} + 2 \cdot 1\right) = \frac{4}{5} = R^m_{norm}(\Delta_1, \Delta_2'),$$

$$R^\mu_{norm}(\Delta_1, \Delta_2) = \frac{11}{16}, \qquad R^\mu_{norm}(\Delta_1, \Delta_2') = \frac{12}{16}.$$

In our example, the retrieval results Δ_2 and Δ_2' yield the same ranking but with different relevance values. Here R^M_{norm} and R^m_{norm} have identical values for the retrieval results (Δ_1, Δ_2) and (Δ_1, Δ_2'), but the R^μ_{norm} values differ. This is due to the property of the R^μ_{norm} measure that takes into account the query-wise ranking of the documents as well as the query-independent interpretation of the retrieval status values assigned by the retrieval function. This property is important for the evaluation of retrieval functions that are assumed to compute estimates of the probability of relevance because these probabilities also have a query-independent interpretation.

6. EXPERIMENTAL RESULTS

We have performed numerous experiments with LSP retrieval functions [9, 18]. Because of the limited space, we only describe the major results here.

In order to develop LSP retrieval functions, a description vector has to be defined first. This description vector is derived from the representations of requests and documents. In our case, requests are represented by a set of descriptors (the Boolean connectives were dropped here), and documents by a set of pairs (descriptor, indexing weight). The elements of the corresponding description vector $\vec{x}(f_l, d_m)$ are listed in Table VI.

Originally, we worked with about twice the number of elements, from which these elements showed the best performance. Then the development of the LSP retrieval functions was performed in the following way: For the description vector $\vec{x}$, different polynomial structures $\vec{v}(\vec{x})$ were defined. For the computation of the coefficient vector $\vec{a}$, sample B was used, and then the retrieval functions were compared by regarding their results for sample C. On the basis of these results, conclusions for better definitions of $\vec{v}(\vec{x})$ were drawn, and the process of adaption and evaluation was repeated. Because of this procedure (and the large number of

Table VI. Elements of the Description Vector $\vec{x}(f_l, d_m)$

Element	Description
x_1	number of descriptors common to query and document
x_2	log(number of descriptors common to query and document)
x_3	highest indexing weight of a common descriptor
x_4	lowest indexing weight of a common descriptor
x_5	number of common descriptors with weight ≥ 0.15
x_6	number of noncommon descriptors with weight ≥ 0.15
x_7	number of descriptors in the document with weight ≥ 0.15
x_8	log $\sum$ (indexing weights of common descriptors)
x_9	log(number of descriptors in the query)
x_{10}	log(min(size of output set, 100))
x_{11}	= 1, if size of output set > 100
x_{12}	= 1, if request is about nuclear physics
x_{13}	proportion of relevant documents in the output set

iterations), sample C can also be regarded as a kind of learning sample. Therefore we finally chose about a dozen retrieval functions that performed best on sample C and tested the performance of these functions on sample A.

For reasons of simplicity, we only regard one component of $\vec{e}(\vec{v}(\vec{x}))$, denoted as $e(\vec{v}(\vec{x}))$ in the following, and the subscript of e_i relates to functions of different polynomial structures $\vec{v}(\vec{x})$. The following discussion refers to two polynomials derived from the description vector listed in Table VI:

$$e_1(\vec{x}) = a_0 + a_1x_1 + a_2x_3 + a_3x_4 + a_4x_5 + a_5x_6 + a_6x_7 + a_7x_9 + a_8x_{10} + a_9x_{11} + a_{10}x_{12} + a_{11}x_5^2 + a_{12}x_5x_6 + a_{13}x_5x_{10} + a_{14}x_6^2 + a_{15}x_6x_{10} + a_{16}x_{10}^2,$$

$$e_2(\vec{x}) = a_0 + a_1x_1 + a_2x_2 + a_3x_8 + a_4x_9 + a_5x_{10} + a_6x_{11} + a_7x_{12} + a_8x_1^2 + a_9x_1x_8 + a_{10}x_1x_{10} + a_{11}x_8^2 + a_{12}x_8x_{10} + a_{13}x_{10}^2.$$

As can be seen from Table VI, the elements $x_9 \cdots x_{13}$ are constant for one request, so their value is not needed for a pure request-specific ranking of the documents. However, the LSP approach needs this information for the estimation of the probability of relevance $P(R \mid \vec{x}(f_l, d_m))$. Experiments without these elements showed inferior results with respect to all three measures R^{μ}_{norm}, R^{M}_{norm}, and R^{m}_{norm}. Because of these results, we regarded a variation e_1', e_2' of the polynomials listed above where each occurrence of one of the elements x_{10}, x_{11}, and x_{12} is replaced by the element x_{13}. This element gives the probability of relevance of an arbitrary document from the output set. The idea behind this was that we wanted to ease the adaption process of the coefficients: As the LSP approach attempts to produce good probability estimates, the retrieval results are optimized with respect to R^{μ}_{norm}. With the additional information about the average probability of relevance for all documents of a query, this task becomes easier. Now the hope was that (as a side effect) the request-oriented ranking (measured by R^{M}_{norm} and R^{m}_{norm}) is also improved. It is obvious that the value of x_{13} would not be available in a real application, but this would only affect the estimation of the probability and thus the results of R^{μ}_{norm} (marked by an asterisk here).

Table VII. Results of the Polynomial Retrieval Functions and the Cosine Measure

Retrieval Function	Sample	R^{μ}_{norm}	R^{M}_{norm}	R^{m}_{norm}
e_1	B	0.752	0.754	0.717
e_1'		0.774*	0.751	0.716
e_2		0.752	0.763	0.717
e_2'		0.771*	0.745	0.714
e_1	C	0.721	0.753	0.717
e_1'		0.771*	0.740	0.708
e_2		0.721	0.714	0.700
e_2'		0.769*	0.710	0.699
cosine		0.668	0.728	0.704
e_1	A	0.741	0.775	0.723
e_1'		0.764*	0.756	0.715
e_2		0.741	0.771	0.731
e_2'		0.778*	0.760	0.729
cosine		0.727	0.769	0.731

The results for the four polynomials are shown in Table VII. The R^{μ}_{norm} values follow the theoretical considerations described above: With the element x_{13} instead of x_{10}, x_{11}, and x_{12}, we get better results. However, the R^{m}_{norm} and R^{M}_{norm} values show that the request-specific ranking gets worse—just the contrary of what we expected. Other experiments where the polynomials e_1 and e_2 were extended by the element x_{13} showed similar results. So we end up with the conclusion that LSP retrieval functions are optimized with respect to $P(R \mid \vec{x}(f_l, d_m))$ and thus yield good R^{μ}_{norm} values, but this does not necessarily imply a good request-specific ranking of the documents.

For comparison, Table VII also shows the results for the cosine similarity function. This retrieval function yields results similar to those of the LSP approach. We think that there are three reasons why the LSP approach does not perform better in this application:

(1) The representation of requests and documents are simple and well suited to the application of the cosine measure. The probabilistic indexing weights of the documents especially cover all the information necessary for achieving a good ranking (this also has been shown for probabilistic models; see [7] and [10]). The main strength of the LSP approach is its ability to handle complex representations, for example, the development of indexing functions [1, 13].

(2) Our learning sample is too small with respect to the number of requests. Since there are 82 queries in sample B, we have only 82 (statistically) independent elements from which we derive the request-specific properties $x_{10}, \ldots, x_{13}$. The experiments described above have shown the importance of request-specific information for the adaptation process of the LSP retrieval functions. So we assume that we would perform better with a larger number of requests in the learning sample.

Table VIII. $R_{norm(2)}$ Values (for Binary Relevance Scale $R_{(2)}$) of Five Retrieval Functions (Sample C)

Retrieval Function	$R^{\mu}_{norm(2)}$	$R^{M}_{norm(2)}$	$R^{m}_{norm(2)}$
e_3	0.761	0.801	0.765
e_4	0.759	0.811	0.768
e_5	0.760	0.784	0.753
e_6	0.767	0.803	0.764
e_7	0.767	0.813	0.770

Table IX. Ranking of the $R^{\mu}_{norm(i)}$ Values for Different Relevance Scales $R_{(i)}$ (Sample C)

Retrieval Function	Relevance Scale					
	$R_{(0)}$	$R_{(1)}$	$R_{(2)}$	$R_{(3)}$	$R_{(4)}$	$R_{(5)}$
e_3	3	1	3	5	5	5
e_4	5	2	5	4	4	4
e_5	4	3	4	3	3	3
e_6	1	4	1	2	2	2
e_7	1	5	1	1	1	1

(3) Because of the probabilistic indexing weights, a log-linear model would be appropriate. We tried several variants of log-linear models for the binary indexing model [7, 10] and the unified model [22], but the best results that we achieved were only similar to those of the more heuristically developed functions described above. So the theoretical problems with log-linear LSP functions also seem to occur in practice.

A second series of experiments we want to describe here relates to multivalued relevance scales. For evaluation, we regard the full relevance scale $R_{(0)}$ with its six values (cf. Table IV), as well as the five possible binary scales $R_{(i)}$, where i denotes the index of the maximum relevance value r_i for which a document is regarded as being nonrelevant. On the basis of these six relevance scales, we compute R_{norm} measures and then rank five retrieval functions (similar to e_1, e_2, see [9, pp. 165–188]) with the R_{norm} values shown in Table VIII according to decreasing values of these measures. The results are shown in Tables IX and X. Although the R_{norm} values for the different functions are very similar, there is nearly no difference in the ranking of these functions with respect to the various relevance scales. The only exception from this observation is the scale $R_{(1)}$. This may be due to the fact that only 6 percent of all documents are treated as being nonrelevant in this case, and that furthermore the distinction between "irrelevant" and "digressive" was perhaps not very clear for the referees. These results support Bookstein's theoretical finding that a probabilistic ranking for a binary relevance scale also gives good results for nonbinary relevance scales.

In all experiments described above, a binary relevance scale was used for the adaptation of the coefficient vectors $\vec{a}$. In order to investigate the influence of

Table X. Ranking of the ($R^{M}_{norm(i)}$, $R^{m}_{norm(i)}$) Values for Different Relevance Scales $R_{(i)}$ (Sample C)

Retrieval Function	Relevance Scale					
	$R_{(0)}$	$R_{(1)}$	$R_{(2)}$	$R_{(3)}$	$R_{(4)}$	$R_{(5)}$
e_3	(4, 4)	(3, 2)	(4, 3)	(4, 3)	(4, 3)	(4, 4)
e_4	(2, 2)	(2, 3)	(2, 2)	(1, 2)	(1, 2)	(2, 2)
e_5	(5, 5)	(1, 1)	(5, 5)	(5, 5)	(5, 5)	(5, 5)
e_6	(3, 3)	(5, 4)	(3, 4)	(3, 4)	(3, 4)	(3, 3)
e_7	(1, 1)	(3, 5)	(1, 1)	(2, 1)	(1, 1)	(1, 1)

Table XI. Retrieval Function Based on Binary ($R_{(2)}$) and Multivalued Relevance ($R_{(0)}$) Scales Adapted on Large (B) and Small (B/10 = Every 10th Request-Document Pair of B) Learning Samples

Retrieval Function	Relevance Scale	Learning Sample	Test Sample	R^{u}_{norm}	R^{M}_{norm}	R^{m}_{norm}
e_1	$R_{(2)}$	B	B	0.752	0.754	0.718
e_1	$R_{(0)}$	B	B	0.756	0.763	0.719
e_1	$R_{(2)}$	B/10	B	0.730	0.739	0.690
e_1	$R_{(0)}$	B/10	B	0.734	0.741	0.691
e_1	$R_{(2)}$	B	C	0.721	0.753	0.717
e_1	$R_{(0)}$	B	C	0.721	0.749	0.714
e_1	$R_{(2)}$	B/10	C	0.699	0.713	0.685
e_1	$R_{(0)}$	B/10	C	0.695	0.718	0.684
e_1	$R_{(2)}$	B	A	0.741	0.775	0.723
e_1	$R_{(0)}$	B	A	0.749	0.772	0.728
e_1	$R_{(2)}$	B/10	A	0.725	0.713	0.689
e_1	$R_{(0)}$	B/10	A	0.730	0.732	0.699
e_2	$R_{(2)}$	B	B	0.752	0.763	0.717
e_2	$R_{(0)}$	B	B	0.753	0.765	0.716
e_2	$R_{(2)}$	B/10	B	0.739	0.749	0.699
e_2	$R_{(0)}$	B/10	B	0.746	0.758	0.706
e_2	$R_{(2)}$	B	C	0.721	0.714	0.700
e_2	$R_{(0)}$	B	C	0.720	0.707	0.695
e_2	$R_{(2)}$	B/10	C	0.689	0.692	0.667
e_2	$R_{(0)}$	B/10	C	0.702	0.689	0.680
e_2	$R_{(0)}$	B	A	0.747	0.764	0.728
e_2	$R_{(2)}$	B	A	0.741	0.771	0.731
e_2	$R_{(0)}$	B/10	A	0.711	0.727	0.690
e_2	$R_{(2)}$	B/10	A	0.696	0.716	0.684

the choice of the relevance scale on final retrieval quality, experiments with a multivalued scale were performed. Therefore, we defined the following cost factors for the retrieval of a document with relevance judgement C_k: $C_1 = C_2 = 1$, $C_3 = 0.7$, $C_4 = 0.5$, $C_5 = 0.3$, and $C_6 = 0$. The results for the retrieval functions based on different relevance scales are given in Table XI. Since the differences between the results of binary and multivalued scales are negligible (other experiments with the binary relevance scale $R_{(5)}$ instead of $R_{(2)}$ showed similar results), we

explored the effect of using smaller learning samples. Even in this case there are only minor differences between the two types of scales. From these results we can draw the conclusion that nonbinary relevance scales do not give more information (in a stochastic sense) than a binary scale. Therefore, the restriction of former work on probabilistic models with binary relevance scales does not have any influence on the findings that were achieved.

Finally, another result can be seen from Table XI: With the smaller learning sample that is $\frac{1}{10}$ of the original size, the retrieval results for the functions adapted on these small samples get worse. Obviously, this is a case of "over-adaptation": We might get improved results for our learning sample, which cannot be reproduced with the test samples.

7. EXPERIMENTS WITH COMPLEX REPRESENTATIONS

As already mentioned, the major advantage of the LSP approach is its ability to cope with complex representations. In fact, the weighted indexing that was used for the experiments described in the previous section is the result of the application of LSP indexing functions to complex representations of document-descriptor relationships [1, 13, 19]. We have also performed a few experiments with retrieval functions based on complex representations of request-document relationships, which will be described in this section.

For the development of the LSP retrieval function, we use the same kind of information as for the indexing functions. Our indexing procedure, which is based on the Darmstadt Indexing Approach [7, 8], yields a probabilistic indexing with descriptors from a controlled vocabulary for documents consisting of titles and abstracts. The procedure starts with the identification of terms (single words, noun phrases, or formula components) in the document text. Since this task cannot be done perfectly, we distinguish several forms of occurrence, corresponding to different levels of certainty of identification and of significance of the term with respect to the document. For noun phrases, the certainty of identification is measured by the number of stopwords between the first and the last component of the phrase in the text. We distinguish good/medium/bad (forms of occurrence of) noun phrases with 0/1, ..., 3/4, ..., 6 stopwords in between. As a measure of significance, we regard the location of the phrase in the document, namely, title versus abstract. For single words and formula components, the form of occurrence is described only by the information about the location of the term.

Our indexing system uses an indexing dictionary that contains relations $r(t_j, s_i)$ between terms and descriptors. For the experiments described below, we used three kinds of relations from the dictionary:

(1) *ID.* The identity relation, that is the information that a term itself is a descriptor.
(2) *USE.* The USE relation of the underlying thesaurus.
(3) *Z.* A statistical relation derived from manually indexed documents. For a term-descriptor pair (t_j, s_i), the weight $z(t_j, s_i)$ is an estimate of the probability that the descriptor s_i would be assigned manually to a document, given that it contains the term t_j in the text.

Table XII. Definition of the Description Vector (n.i.w. = number of indications with . . .)

Element	Description
x_1	n.i.w. USE relations
x_2	n.i.w. ID relations
x_3	n.i.w. Z relations
x_4	n.i.w. Z relations with $z(t_j, s_i) \geq 0.7$
x_5	$\log \sum z(t_j, s_i)$
x_6	= 1, if any indications with Z relation
x_7	n.i.w. single words in the title
x_8	n.i.w. single words in the abstract
x_9	n.i.w. formulas in the title
x_{10}	n.i.w. formulas in the abstract
x_{11}	n.i.w. good noun phrases in the title
x_{12}	n.i.w. medium noun phrases in the title
x_{13}	n.i.w. bad noun phrases in the title
x_{14}	n.i.w. good noun phrases in the abstract
x_{15}	n.i.w. medium noun phrases in the abstract
x_{16}	n.i.w. bad noun phrases in the abstract
x_{17}	number of descriptors common to query and document
x_{18}	number of indications from different terms
x_{19}	log(number of indications)
x_{20}	number of common descriptors with Z and ID/USE relations
x_{21}	log(document length)
x_{22}	log(min(size of output set, 100))
x_{23}	= 1, if size of output set $\leq$ 100)
x_{24}	number of descriptors in the query
x_{25}	= 1, if request is about nuclear physics
x_{26}	= 1, if request is about physics of solid materials
x_{27}	= 1, if request is about other subfields of physics

When a term t_j is identified in a document d_m and the indexing dictionary contains a relation $r(t_j, s_i)$, this is called an indication from t_j to s_i. The description of the indication formed by the system contains information about the form of occurrence of t_j in d_m and about the relation $r(t_j, s_i)$. The collection of all indications from a document d_m to the same descriptor s_i is called the relevance description of s_i with respect to d_m. The relevance description is the basis for the development of the indexing function, which assigns a probabilistic weight to the document-descriptor relationship.

In order to develop retrieval functions, we start with request-document pairs (f_l, d_m) and divide the set of all relevance descriptions from the document into two subsets, one for the descriptors occurring in the query formulation as well as in the document, and one for the remaining descriptors. In the following, we only regard the first set of relevance descriptions; some experiments described in [9, pp. 188–191] show that retrieval functions based on both sets yield about the same results as the functions based on the first set only. So our representation of the request-document relationship is a set of relevance descriptions, where each relevance description is a set of indications. For the application of the LSP approach, this tree-like structure has to be mapped onto a description vector. The structure of this vector is shown in Table XII. With this vector, three linear retrieval functions $e_8^{(i)}$ were developed by restricting the number of coefficients

Table XIII. Retrieval Results for Retrieval Functions $e_8^{(i)}$ Based on Complex Representations versus e_1 Based on Weighted Indexing

Retrieval Function	Sample	R^{μ}_{norm}	R^{M}_{norm}	R^{m}_{norm}
$e_8^{(25)}$	C	0.717	0.710	0.690
$e_8^{(20)}$		0.718	0.712	0.693
$e_8^{(15)}$		0.715	0.707	0.689
e_1		0.721	0.753	0.717
$e_8^{(20)}$	A	0.738	0.774	0.723
$e_8^{(15)}$		0.736	0.779	0.719
e_1	A	0.741	0.775	0.723

to be computed to $i = 15, 20, 25$. The retrieval results of these functions are shown in Table XIII.

As can be seen from these results, the retrieval quality is about the same as for the retrieval functions based on weighted indexing. In contrast to the weighted indexing case, however, it is almost impossible to define a retrieval function based only on a theoretical model that would yield similar retrieval results for the complex representation. Here using some kind of regression method is inevitable because any other model would be based on simplifying assumptions that are too far away from the real data.

Since we performed a large series of experiments with weighted indexing and only a few with complex representations, it may be possible to improve the results in the latter case by investigating other definitions of the description vector. As long as there is no significant improvement, it seems to be appropriate and efficient to apply the LSP approach for a probabilistic indexing of the documents and to use one of the simpler retrieval functions based on probabilistic models [7]. This statement, however, is not valid for a dynamic document collection, like that in office information systems. In such an environment, probabilistic indexing may no longer be valid after some time. For the application of an LSP retrieval function, the indexing dictionary could change over time, leading to a different value of the description vector and to estimates of the probability of relevance that relate to the current state of the collection. This kind of dynamic and adaptive behavior of the document representation and the retrieval function is also a goal of "genetic algorithms" (see, e.g., [14]) which, however, are based on a less solid theoretical framework.

8. CONCLUSIONS

In this paper, we have discussed the probabilistic concept of optimum retrieval. We have shown that actual retrieval functions cannot reach this optimum because it is impossible to estimate the correct probabilities for each specific representation of documents or requests. Therefore, additional simplifying assumptions are necessary. For most probabilistic models, certain independence assumptions are

postulated, so the performance of a model depends on the extent to which the assumptions fit reality. With the LSP approach, the situation is quite different: Here a class of polynomial retrieval functions is defined on the basis of some heuristics, and then the function that fits the available data best is selected. Of course, the definition of the description vector and the class of polynomials is also based on some implicit assumptions. However, these assumptions are in general more vague, and it is not required to state them as explicitly as with the other probabilistic models.

The crucial point of this approach is the selection of features that make up the description vector. It would be desirable to have some theoretically well-founded method for this step. Another weakness of the LSP approach is the need for large learning samples, which is due to the problem of parameter estimation (see [11] and [12]). Therefore, appropriate estimation methods for small samples should be developed, for example, by modifying the learning data. Finally, the LSP approach is not suited to log-linear functions that are suggested by some theoretical models.

In contrast to most other types of retrieval functions, the LSP approach yields estimates of the probability of relevance and can also cope with multivalued relevance scales. The major advantage of the LSP approach is its ability to deal with complex representations of retrieval objects. Especially for knowledge-based retrieval methods (see, e.g., [5]), this approach is well suited: The representation is more complex than in the simple term-based case, and some kind of weighting scheme is needed in any case. For this problem, LSP functions can yield an optimum weighting based on the Probability Ranking Principle.

In the areas of pattern recognition and of machine learning, a number of sophisticated procedures for classifying complex objects have been developed. These methods should be considered with respect to their applicability in the field of information retrieval, especially those that are based on a probabilistic model: They have a well-founded theoretical background and can be shown to be optimum with respect to certain reasonable restrictions.

REFERENCES

1. BIEBRICHER, P., FUHR, N., KNORZ, G., LUSTIG, G., AND SCHWANTNER, M. The automatic indexing system AIR/PHYS—from research to application. In *11th International Conference on Research and Development in Information Retrieval*, Y. Chiaramella, Ed. Presses Universitaires de Grenoble, Grenoble, France, 1988, pp. 333–342.
2. BOLLMANN, P., JOCHUM, R., REINER, U., WEISSMANN, V., AND ZUSE, H. Planung und Durchführung der Retrievaltests. In *Leistungsbewertung von Information Retrieval Verfahren (LIVE)*, H.-J. Schneider et al., eds. TU Berlin, Fachbereich Informatik, Computergestützte Informationssysteme (CIS), Institut für Angewandte Informatik, 1986, pp. 183–212.
3. BOOKSTEIN, A. *Loglinear Analysis of Library Data.* Research Report, OCLC, Office of Research, 1988.
4. BOOKSTEIN, A. Outline of a general probabilistic retrieval model. *J. Doc. 39*, 2 (1983), 63–72.
5. CROFT, W. B. Approaches to intelligent information retrieval. *Inf. Process. Manage. 23*, 4 (1987), 249–254.
6. DUDA, R. O., AND HART, P. E. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.
7. FUHR, N. Models for retrieval with probabilistic indexing. *Inf. Process. Manage. 25*, 1 (1989), 55–72.

8. FUHR, N. A probabilistic model of dictionary based automatic indexing. In *Proceedings of the Riao 85 (Recherche d'Informations Assistee par Ordinateur)* (Grenoble, France, March 18–20). 1985, pp. 207–216.
9. FUHR, N. *Probabilistisches Indexing und Retrieval.* Fachinformationszentrum Karlsruhe, Eggenstein-Leopoldshafen, West Germany, 1988.
10. FUHR, N. Two models of retrieval with probabilistic indexing. In *Proceedings of the 9th Annual Conference on Research and Development in Information Retrieval* (Pisa, Italy, Sept. 8–10). F. Rabitti, ed. ACM, New York, 1986, pp. 249–257.
11. FUHR, N., AND HÜTHER, H. Optimum probability estimation based on expectations. In *11th International Conference on Research and Development in Information Retrieval,* Y. Chiaranella, ed. Presses Universitaires de Grenoble, Grenoble, France, 1988, pp. 257–273.
12. FUHR, N., AND HÜTHER, H. Optimum probability estimation from empirical distributions. *Inf. Process. Manage. 25,* 3 (1989).
13. FUHR, N., AND KNORZ, G. Retrieval test evaluation of a rule based automatic indexing (AIR/PHYS). In *Research and Development in Information Retrieval,* C. J. Van Rijsbergen, ed. Cambridge University Press, Cambridge, England 1984, pp. 391–408.
14. GORDON, M. Probabilistic and genetic algorithms for document retrieval. *Commun. ACM 31,* 10 (Oct. 1988), 1208–1218.
15. KEEN, E. M. Evaluation parameters. In *The SMART Retrieval System—Experiments in Automatic Document Processing,* G. Salton, ed. Prentice Hall, Englewood Cliffs, N.J., 1971, pp. 74–112.
16. KNORZ, G. *Automatisches Indexieren als Erkennen abstrakter Objekte.* Niemeyer, Tübingen, West Germany, 1983.
17. KNORZ, G. A decision theory approach to optimal automatic indexing. In *Research and Development in Information Retrieval,* G. Salton and H.-J. Schneider, eds. Springer, Berlin, West Germany, 1983, pp. 174–193.
18. KONSTANTIN, J. *Untersuchung von nach dem Quadratmittel-Polynomansatz erstellten Rankingfunktionen.* Diplomarbeit, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II, Darmstadt, West Germany 1985.
19. LUSTIG, G. *Automatische Indexierung zwischen Forschung und Anwendung.* Olms, Hildesheim, West Germany 1986.
20. RIJSBERGEN, C. J. *Information Retrieval,* 2nd ed. Butterworth, London, 1979.
21. ROBERTSON, S. E. The probability ranking principle in IR. *J. Doc. 33* (1977), 294–304.
22. ROBERTSON, S. E., MARON, M. E., AND COOPER, W. S. Probability of relevance: A unification of two competing models for document retrieval. *Inf. Tech. Res. 1* (1982), 1–21.
23. ROCCHIO, J. J. Relevance feedback in information retrieval. In *The SMART Retrieval System—Experiments in Automatic Document Processing,* G. Salton, ed. Prentice Hall, Englewood Cliffs, N.J., 1971.
24. SALTON, G., ED. *The SMART Retrieval System—Experiments in Automatic Document Processing.* Prentice Hall, Englewood Cliffs, N.J., 1971.
25. SCHÜRMANN, J. *Polynomklassifikatoren für die Zeichenerkennung. Ansatz, Adaption, Anwendung.* Oldenbourg, München, West Germany, 1977.
26. WONG, S. K. M., YAO, Y. Y., AND BOLLMANN, P. Linear structure in information retrieval. In *11th International Conference on Research and Development in Information Retrieval,* Y. Chiaramella, ed. Presses Universitaires de Grenoble, Grenoble, France, June 1988, pp. 219–232.

Received April 1989; revised May 1989; accepted May 1989