Combining Model-Oriented and Description-Oriented Approaches for Probabilistic Indexing

Norbert Fuhr, Ulrich Pfeifer *
Technische Hochschule Darmstadt, Germany

Abstract

We distinguish model-oriented and description-oriented approaches in probabilistic information retrieval. The former refer to certain representations of documents and queries and use additional independence assumptions, whereas the latter map documents and queries onto feature vectors which form the input to certain classification procedures or regression methods. Description-oriented approaches are more flexible with respect to the underlying representations, but the definition of the feature vector is a heuristic step. In this paper, we combine a probabilistic model for the Darmstadt Indexing Approach with logistic regression. Here the probabilistic model forms a guideline for the definition of the feature vector. Experiments with the purely theoretical approach and with several heuristic variations show that heuristic assumptions may yield significant improvements.

1 Introduction

Probabilistic retrieval functions aim at the estimation of the probability $P(R|q_k, d_m)$ that a document d_m will be judged relevant by a user with request q_k. A large number of probabilistic models have been developed for this problem, differing in the assumptions about the representation of queries and documents and about the statistical dependence or independence of elements of these representations. In [Fuhr 89c], an alternative approach (called description-oriented in the following) is presented, which adapts concepts from pattern recognition methods for probabilistic retrieval: First, request-document pairs (q_k, d_m) are mapped onto description vectors $\boldsymbol{x}(q_k, d_m)$, where each component gives the value of a specific feature of the pair (q_k, d_m) (e.g. the number of terms common to query and document, the sum of the weights of these terms w.r.t. the document, the total number of terms in the document). Second, a regression method (i.e. least square polynomials) is applied in order to develop a retrieval function $e(\boldsymbol{x})$ such that it yields estimates of the probability $P(R|\boldsymbol{x}(q_k, d_m))$. For this purpose, a learning sample of request-document pairs with relevance judgements must be given. A major advantage of the description-oriented approach is its flexibility with respect to the underlying representations. In the model-oriented approach, most changes of the representation (e.g. considering the within-document frequency of a term instead of regarding only the presence or absence of a term) would require the formulation of a new probabilistic model. With the description-oriented approach, only the set of features that are included in the description vector has to be revised.

A second advantage of the description-oriented approach is its adaptability to the amount of learning data available. For example, several dependence models have been developed [Rijsbergen 77] [Yu et al. 83] in order to overcome the limitations of the binary independence retrieval model [Yu & Salton 76] [Robertson & Sparck Jones 76]. However, due to the small size of the learning samples available, parameter estimation problems lead to worse retrieval results for the dependence models in comparison to those of the independence models. In the description-oriented approach, only the number of features considered in the description vector has to be modified according to the size of the learning sample. As described in [Fuhr & Buckley 90] [Fuhr & Buckley 91] (for the case of indexing functions), the description vector represents an abstraction from specific elements (queries, documents, or terms), and the level of abstraction can be

*Authors' new address: Universität Dortmund, Lehrstuhl Informatik VI, W-4600 Dortmund 50, Germany.
E-mail: [fuhr|pfeifer]@grete.informatik.uni-dortmund.de

chosen according to the amount of learning data available.

On the other hand, a major disadvantage of the description-oriented approach is the heuristics employed in the definition of the description vector. In order to optimize the retrieval function, a large series of experiments with varying definitions has to be performed (although fairly good functions can be achieved with little effort).

In this paper, we try to combine both approaches. As application, we regard indexing functions (instead of retrieval functions). In probabilistic document indexing, we seek for the estimation of the probability $P(C|s_i, d_m)$ that the asssigment of the indexing term s_i to the document d_m is correct; the decision about the correctness can be specified either explicitly (by comparison with manual indexing) or implicitly (derived from relevance judgements, see [Fuhr & Buckley 91]). This task is analoguous to the retrieval problem: Instead of a query, we regard a single index term (from a prescribed indexing vocabulary), and relevance is replaced by correctness. As a major advantage, we have large learning samples available for our experiments.

For the combination of both approaches, we present a new probabilistic indexing model, which can be transformed into a log-linear form. This kind of function allows the application of logistic regression, where the model leads us to the definition of the description vector. The regression method generalizes the model and fits the parameters of the function to the data.

In the following, we first present the probabilistic indexing model. Then we describe the logistic regression method and show how this method can be combined with log-linear models. Section 5 presents the test setting and the evaluation methods used for our experiments, followed by the experimental results given in section 6.

2 A new probabilistic model for the Darmstadt Indexing Approach

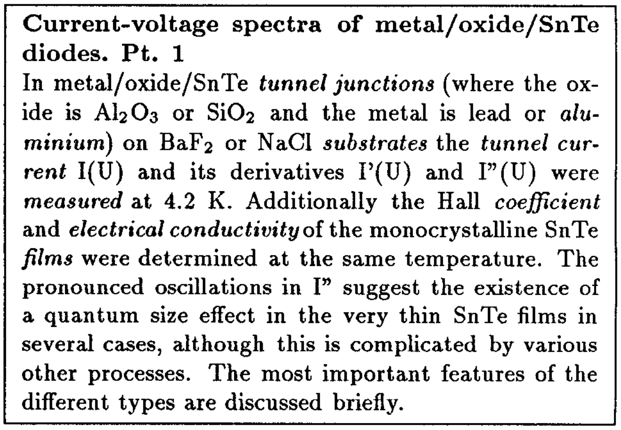
The "Darmstadt Indexing Approach" (DIA) is a dictionary-based approach for automatic indexing from document titles and abstracts, with index terms (called descriptors here) from a prescribed indexing vocabulary ([Fuhr 89b] [Fuhr et al. 91]). For the task of mapping text content onto the set of descriptors, the approach needs an indexing dictionary containing term-descriptor rules for as many terms (i.e. words or phrases) of the application field as possible. Furthermore, the indexing dictionary contains term-specific and descriptor-specific information. This data can be derived automatically from a large sample of manually indexed documents (see below).

The indexing process starts with the identification of terms in the text. As this task cannot be done perfectly, each term is identified in a certain *form of occurrence* (FOC) v, where different FOCs correspond to different levels of confidence.

FOCs are defined with respect to some formal parameters that are computed by the system. Actually, the concept of FOC comprises two aspects:

1.) the certainty with which a term is identified,
2.) the significance of a term with respect to the document.

In our experiments described here, we consider for all kinds of terms the location of the term (title vs. abstract) as a criterion for the significance of the term. For phrases, the certainty of identification is measured by the number of stopwords between the first and the last component: P+ with no stopwords and P- with 1 ...3 stopwords in between. For the document shown in figure 1, examples for P+ are "TUNNEL JUNCTION", "TUNNEL CURRENT" and "ELECTRICAL CONDUCTIVITY", and for P– "ALUMINIUM SUBSTRATES", "MEASURE ELECTRICAL" and "FILM COEFFICIENT". So we have two different FOCs for words, namely W/TI and W/AB according to their location in the title or in the abstract, and four different FOCs for phrases: P+TI, P-TI, P+AB and P-AB.



Current-voltage spectra of metal/oxide/SnTe diodes. Pt. 1

In metal/oxide/SnTe *tunnel junctions* (where the oxide is Al_2O_3 or SiO_2 and the metal is lead or *aluminium*) on BaF_2 or NaCl *substrates* the *tunnel current* I(U) and its derivatives I'(U) and I"(U) were *measured* at 4.2 K. Additionally the Hall *coefficient* and *electrical conductivity* of the monocrystalline SnTe *films* were determined at the same temperature. The pronounced oscillations in I" suggest the existence of a quantum size effect in the very thin SnTe films in several cases, although this is complicated by various other processes. The most important features of the different types are discussed briefly.

Figure 1: Example document

If a term t is identified in a document d and a term-descriptor rule $t \rightarrow s$ is stored in the directory, a *descriptor indication* from t to s is generated. It contains

- the form of occurrence v of t in d,
- the rule $t \rightarrow s$,
- further information about s and t (from the dictionary) and d.

For example, our indexing dictionary contains the rule "ELECTRICAL CONDUCTIVITY" *USE* "ELECTRIC CONDUCTIVITY", so the indexing system produces the corresponding descriptor indication for the example document shown above, where the FOC is P+AB. Another example is the statistical rule "ELECTRICAL COEFFICIENT" → "ELECTRIC CONDUCTIVITY" with a weight of 3.0 (see the definition of the parameter r_{ik} described below), for which an indication with the FOC P-AB is generated. As additional information about d, we might regard the number of different words in the document, because longer documents tend to produce more descriptor indications, but the number of descriptor asssigments to a document should be independent of this factor.

The collection of all descriptor indications from a document leading to the same descriptor s is called the *relevance descripton* $x(s,d)$ of s w.r.t. d. Now a probabilistic *indexing function* $a(x(s,d))$ has to be developed which yields estimates of the probability $P(C|x(s,d))$ that the assignment of a descriptor s to document d will be judged correct, given that the descriptor-document pair is represented by relevance description x. For this purpose, a learning sample with document-descriptor pairs and decisions about the correctness of assignments must be given; then several probabilistic classification methods can be applied for the development of an indexing function (see [Fuhr & Buckley 91] [Fuhr et al. 91]).

In [Fuhr 89a], a new probabilistic model for the DIA is derived. For the description of the indexing functions following from this model, we use the following notations:

T $= \{t_1, \dots, t_n\}$ is the set of terms in our collection, and

V $= \{v_1, \dots, v_g\}$ is the set of forms of occurrence that we distinguish. In addition, let v_0 denote the absence of a term (that is, it is not identified in any form $v_l \in V$ in the current document), and let $V_0 = \{v_0\} \cup V$.

E is the set of document characteristics that we regard. That is, the information describing a document that is independent of a specific relevance description is mapped onto an element $e \in E$ (e.g. the number of words in the document as mentioned above).

For a single document-descriptor pair (d_m, s_k), we have:

- the decision about the correctness (C or $\overline{C}$)
- the document characteristics e_m of d_m
- the terms that occur within d_m together with their forms of occurrence.

The latter information is described by a vector $y_m = \{y_{m_1}, \dots, y_{m_n}\}$ in the following, where $y_{m_i} \in V_0$ has the value of the form of occurrence of t_i in d_m (or v_0, respectively).

With this notation, a relevance description $x(s_k, d_m)$ can be mapped onto a triple (s_k, e_m, y_m), and we seek for the probability $P(C|s_k, e_m, y_m)$. For the formulation of our model, we use odds, where $O(x) = P(x)/P(\overline{x})$. Let $P(y_{m_i}|s_k, C)$ (with $y_{m_i} \in V$) be the probability that term t_i will occur with FOC y_{m_i} in an arbitrary document to which s_k was assigned correctly, whereas $P(y_{m_i} = v_0|s_k, C)$ denotes the probability that t_i will not occur in such a document. $P(y_{m_i}|s_k, \bar{C})$ and $P(y_{m_i} = v_0|s_k, \bar{C})$ denote the corresponding probabilities for documents to which the assigment of s_k is not correct.

With

$$\begin{aligned} w_{ik0} &= \frac{P(y_{m_i} = v_0|s_k, C)}{P(y_{m_i} = v_0|s_k, \bar{C})} \\ w_{ikl_{im}} &= \frac{P(y_{m_i} = v_{l_{im}}|s_k, C)}{P(y_{m_i} = v_{l_{im}}|s_k, \bar{C})} \end{aligned}$$

(where $v_{l_{im}}$ denotes the form of occurrence of t_i in d_m, that is $y_{m_i} = v_{l_{im}}$), our first indexing function yields

$$O(C|s_k, e_m, y_m) = \frac{O(C|s_k) \cdot O(C|e_m)}{O(C)} \cdot \prod_{y_{m_i} \in V} \frac{w_{ikl_{im}}}{w_{ik0}} \prod_{i=1}^{n} w_{ik0} \quad (1)$$

Here $O(C)$ is the odds that an arbitrary document-descriptor relationship is correct, $O(C|e_m)$ denotes the odds that the assigment of an arbitrary descriptor to a document with characteristics e_m will be judged correct, and $O(C|s_k)$ is the odds that the assigment of descriptor s_k to an arbitrary document is correct. In the following, we will refer to this formula as the *FOC dependence model.*

A major disadvantage of the FOC dependence model is the fact that the term descriptor rules (i.e. the probabilities $P(y_{m_i} = v_{lim}|s_k, C)$ and $P(y_{m_i} = v_{lim}|s_k, \bar{C})$ are FOC-specific, so for each term-descriptor pair, parameter values for all $v_{lim} \in V_0$ are required. In order to overcome this problem, we have derived a second probabilistic model in which the distributions of FOCs and term-descriptor pairs are assumed to be independent of each other. The *FOC independence model* based on this assumption uses the parameters $f_{lim} = P(v_{lim}|C)/P(v_{lim}|\bar{C})$ and $r_{ik} = O(C|t_i, s_k)$ instead of $w_{ikl_{im}}$. Here $P(v_{lim}|C)$ denotes the probability that an arbitrary term will occur with FOC v_{lim} in an arbitrary relevance description leading to a correct (incorrect) assignment, whereas $O(C|t_i, s_k)$ is the odds that the assignment of descriptor s_k to an arbitrary document will be correct, given that term t_i occurs in this document. With these parameters, we can derive the indexing function

$$O(C|s_k, e_m, y_m) = \frac{O(C|s_k)O(C|e_m)}{O(C)}$$
$$\cdot \prod_{i=1}^{n} w_{iko} \cdot \prod_{y_{m_i} \in V} O^2(C) \cdot O^3(C|s_k) \cdot f_{l_{im}} \cdot \frac{r_{ik}}{w_{iko}} \, (2)$$

For the discussion in the following, we bring the indexing functions (1) and (2) into their log-linear form. With $\lambda(x) = \log O(x)$, $\omega_{ikj} = \log w_{ikj}$ and $\beta_k = \log\prod_{i=1}^{n} w_{iko}$, the FOC dependence model yields:

$$\begin{aligned} &\lambda(C|s_k, l_m, y_m) = \\ &\quad \lambda(C|e_m) + \lambda(C|s_k) - \lambda(C) + \beta_k \\ &\quad + \sum_{y_{mi}\in V} (\omega_{ikl_{im}} - \omega_{ik0}) \end{aligned} \qquad (3)$$

Furthermore, let $\varphi_{l_{im}} = \log f_{l_{im}}$ and $\varrho_{ik} = \log r_{ik}$, then the FOC independence model can be transformed into

$$\begin{aligned} &\lambda(C|s_k, l_m, y_m) = \\ &\quad \lambda(C|e_m) + \lambda(C|s_k) - \lambda(C) + \beta_k + \\ &\quad \sum_{y_{mi}\in V} (2\lambda(C) + 3\lambda(C|s_k) + \varphi_{l_{im}} + \varrho_{ik} - \omega_{ik0}) (4) \end{aligned}$$

3 Logistic regression

As indexing function $a(x(s,d))$, we regard logistic functions here [Freeman 87] [Fienberg 80]. For this type of functions, the relevance description must be in vector form $\boldsymbol{x}(s,d)$ (also called description vector), with $\boldsymbol{x} = (x_1, \ldots, x_m)^T$. Let $\boldsymbol{b} = (b_1, \ldots, b_m)^T$ denote a parameter vector that is to be estimated, then logistic indexing functions have the form

$$a(\boldsymbol{x}) = \frac{\exp(\boldsymbol{b}^T\boldsymbol{x})}{1+\exp(\boldsymbol{b}^T\boldsymbol{x})}$$

Since we want to vary $\boldsymbol{b}$ in order to find an optimum logistic function, we will include the parameter vector as argument of the indexing function in the following, that is $a(\boldsymbol{x}, \boldsymbol{b})$.

For the development of an indexing function, we need a learning sample of relevance descriptions $x = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$ with corresponding correctness decisions $Y = (y_1, \ldots, y_t)$, where $y_i = 1$ if for $\boldsymbol{x}_i(s,d)$, the assignment of s to d is correct, and $y_i = 0$ otherwise.

As optimizing criterion, maximum likelihood is used here (minimum squared errors also would be possible). For a given learning sample and a specific parameter vector $\boldsymbol{b}$, the likelihood function yields

$$L(\boldsymbol{b}) = \prod_{k=1}^{t} a(\boldsymbol{x}_k, \boldsymbol{b})^{y_k}[1 - a(\boldsymbol{x}_k, \boldsymbol{b})]^{1-y_k}$$

In the following, the log-likehood function $l(\boldsymbol{b}) = \log L(\boldsymbol{b})$ is regarded:

$$l(\boldsymbol{b}) = \sum_{k=1}^{t} y_k \log a(\boldsymbol{x}_k, \boldsymbol{b}) + (1-y_k)\log[1 - a(\boldsymbol{x}_k, \boldsymbol{b})].$$

The maximum likelihood estimate of the parameter vector is the value at which l takes its maximum, i.e., the value of $\boldsymbol{b}$ at which

$$\frac{\delta l}{\delta b_i} = \sum_{k=1}^{t}[y_k - a(\boldsymbol{x}_k, \boldsymbol{b})]x_{k_i} = g_i(\boldsymbol{b}) = 0. \qquad (5)$$

In general, this equation cannot be solved in closed form, so an iterative method has to be applied. Here we use the Newton-Raphson method, for which we need the derivations

$$\begin{aligned} \frac{\delta^2 l}{\delta b_i \delta b_j} &= \frac{\delta g_i}{\delta b_j} = \sum_{k=1}^{t} a(\boldsymbol{x}_k, \boldsymbol{b})[1 - a(\boldsymbol{x}_k, \boldsymbol{b})]x_{k_i}x_{k_j} \\ &= H_{ij}(\boldsymbol{b}). \end{aligned} \qquad (6)$$

With $\boldsymbol{g}(\boldsymbol{b})$ and the matrix $H(\boldsymbol{b})$ computed this way, we get the iteration formula

$$\boldsymbol{b}^{(n+1)} = \boldsymbol{b}^{(n)} - H^{-1}(\boldsymbol{b}^{(n)}) \cdot \boldsymbol{g}(\boldsymbol{b}^{(n)}) \qquad (7)$$

So we first have to choose a starting vector $\boldsymbol{b}^{(0)}$, and then $H(\boldsymbol{b}^{(n)})$, $\boldsymbol{g}(\boldsymbol{b}^{(n)})$ and $\boldsymbol{b}^{(n+1)})$ are computed in each iteration step n, until a stopping criterion is fulfilled. In the computation process, the matrix H is not inverted, instead we solve the linear equation $H(\boldsymbol{b}^{(n)}) \cdot \Delta^{(n)} = \boldsymbol{g}(\boldsymbol{b}^{(n)})$ with $\Delta^{(n)} = \boldsymbol{b}^{(n)} - \boldsymbol{b}^{(n+1)}$. As stopping criterion, one can compare $|\Delta^{(n)}|$ with a predefined value ε. Since we performed experiments with different vector lengths m, we choose to regard the value

$$\beta^{(n+1)} = \max_i \left| \frac{b_i^{(n+1)} - b_i^{(n)}}{b_i^{(n+1)}} \right|$$

instead. In our experiments, we used description vectors with up to 40 components, and we always got good solutions within 5 to 6 iterations (that is, further iterations did not improve the indexing quality).

4 Combination of logistic regression and log-linear models

Logistic functions have in fact the same form as log-linear models, which can be seen easily when we transform the approximation that is given by the logistic function

$$P(C|\boldsymbol{x}) \approx \frac{\exp(\boldsymbol{b}^T\boldsymbol{x})}{1+\exp(\boldsymbol{b}^T\boldsymbol{x})}$$

into $O(C|\boldsymbol{x}) \approx \exp(\boldsymbol{b}^T\boldsymbol{x})$, and thus $\lambda(C|\boldsymbol{x}) \approx \boldsymbol{b}^T\boldsymbol{x}$. For the discussion below, let us assume that we have a constant element in addition to the description vector, so the logistic functions that we want to consider are of the form

$$\lambda(C|\boldsymbol{x}) \approx b_0 + \sum_{i=1}^{m} b_i x_i$$

Obviously, most probabilistic models can be brought into such a log-linear form, where the b_is are some coefficients (mostly -1 or +1) and the x_is are certain probabilities (or products or quotients of probabilities). This feature suggests the combination of probabilistic models with logistic regression[1]. In [Robertson & Bovey 82], this kind of combination (which they call logistic model) has been applied to the binary independence retrieval model. However, the experimental results of this work showed no strong evidence for the logistic model in comparison to the original probabilistic model, even for half-collection experiments. We think that these results are caused by the limited size of the learning data. From experiments with least square polynomial functions described in [Fuhr 89b], [Knorz 83a] and [Fuhr & Buckley 91], we know that regression methods require significantly larger learning samples than simple parameter estimation schemes. For our indexing experiments described here, learning sample size is no problem, since we work with samples of at least 3000 elements.

Robertson and Bovey mentioned the following important properties of logistic models:

- In comparison to the original probabilistic model, the independence assumptions of the logistic model are weaker. In the logistic model, it is only assumed that the degree of dependence of any set of elements is the same in the set of correct relevance descriptions as in the set of incorrect ones.
- A major advantage of the logistic model is that the estimation process does not require the learning sample being a random sample from the whole event space. It is only assumed that the learning and the test samples are representative for each other. For our indexing task, we only consider relevance descriptions with at least one descriptor indication, whereas the probabilistic model regards all possible document-descriptor pairs.
- In the logistic model, prior distributions for the parameters of the model can be considered, so the model yields Bayesian estimates of these parameters. This feature is important for small learning samples.

Now we discuss some aspects of the application of logistic repression to our probabilistic indexing model, especially the definition of the description vector $\boldsymbol{x}$:

- The mapping of the elements of the probabilistic formula onto elements of the description vector can be defined in two different ways, depending on the fact whether the number of elements of the formula is fixed or variable (for one logistic regression formula): For the development of query-specific retrieval functions as described in [Robertson & Bovey 82], the dimension of $\boldsymbol{x}$ can be set equal to the number of query terms. In our probabilistic indexing function, we have a varying number of factors for different descriptors (depending on the number of term-descriptor rules stored in the indexing dictionary for the specific descriptor), and we want to develop descriptor-independent indexing functions (otherwise our learning samples would become to small). In this case, we have to map variable numbers of elements of the formula onto one description vector element. For example, we can split the sum in formula (6) according to the different forms of occurrence:

  $$\begin{aligned}\lambda(C|s_k, e_m, y_m) = \\ \lambda(C|e_m) + \lambda(C|s_k) - \lambda(C) + \beta_k \\ + \sum_{j=1}^{|V|} \left(\sum_{y_{m_i}=v_j} \omega_{ikj} \right) - \sum_{y_{m_i}\in V} \omega_{ik0}\end{aligned} \qquad (8)$$

  and define $|V| + 5$ vector elements for this formula.
- In many cases, not all of the parameters of the probabilistic formula are known in advance. For this problem, logistic regression can be used in different ways for estimating probabilistic parameters:
  - For binary features, (e.g. presence or absence of a term), we can set $x_i = 0/1$, and the logistic approach will (in the ideal case) yield the coefficients $b_i = \lambda(x_i = 1) - \lambda(x_i = 0)$ and $b_0 = b'_0 + \lambda(x_i = 0)$, where b'_0 is the value of the constant element before the inclusion of x_i. Similarly, we can set $x_i = n$ for the case when we need $\log O^n(.)$.
  - In the case of our indexing function, we do not have the specific values c_{ikj} for different FOCs in our indexing dictionary, only general values c_{ik} relating to all FOCs included in V. In order to derive estimates for the different forms of occurrence, we can assume general numeric relationships between the values ω_{ik} and ω_{ikj}: For that, we assume the existence of a value α_j for each form of occurrence $v_j \in V$ and test the relationships $\omega_{ikj} = \alpha_j + \omega_{ik}$ and $\omega_{ikj} = \alpha_j \cdot \omega_{ik}$. In the former case, we define one vector component for each FOC v_j, counting in this element the number of indications with v_j, and sum up the values ω_{ik} in an additional vector element. In the latter case, we sum up the ω_{ik}'s separately for each v_j in a specific vector element.
- Finally, there may be cases where we want to perform a general regression, because we assume that the value of a missing probabilistic parameter is a monotonic function of a known non-probabilistic parameter: In our indexing application, we do not have the values $P(C|e_m)$. However, we can assume that this

[1] In [Fuhr 89c], we have discussed the relationship between log-linear models and least square polynomials, and it is shown that this type of function is not suited for a combination of both.

probability is a function of the number of descriptor indications of the document, and therefore we include this information in the vector.

5 Test settings

For our experiments, we used documents from the physics data base PHYS of the Fachinformationszentrum Karlsruhe, Germany. For this application, an indexing dictionary named PHYS/PILOT was developed. As probabilistic term-descriptor rules, the association factors $z(t,s) = h(t,s)/f(t)$ were computed from 392 000 manually indexed documents, where $f(t)$ denotes the number of documents containing the term t and $h(t,s)$ is the number of those among the $f(t)$ documents to which the descriptor s was assigned manually (terms can be words, phrases, or terms derived from formulas). So $z(t,s)$ yields an approximation of the probability $P(C|s,t)$. With the additional criteria $z(t,s) \geq 0.3$ and $h(t,s) \geq 3$, more than 800 000 term-descriptor pairs were obtained. Because of performance reasons, only 350 000 of these pairs (chosen by some heuristic criterica) have been included as relation Z in PHYS/PILOT[2].

In addition to these term-descriptor pairs with probabilistic weights, the PHYS/PILOT dictionary also contains the following relations:

- the USE relation of the PHYS thesaurus,
- the IDENTITY relation connecting each of the 22 683 descriptors with itself,
- the FORMULA IDENTITY relation mapping identifiers derived from formulas onto the corresponding descriptors (In our experiments, we do not distinguish between IDENTITY and FORMULA IDENTITY, and not between formula identifiers and words.).

The size of the PHYS/PILOT dictionary is illustrated in table 1.

Since the parameters $O(C|s_k)$ were not contained in the original PHYS/PILOT dictionary, we computed these factors from a sample of 20 000 documents, where only those descriptors were considered which were assigned to at least 3 documents. Due to this procedure, some of our experiments are restricted to relevance descriptions where the parameter $O(C|s_k)$ is available; furthermore, we consider only descriptor indications based on relation Z in these experiments.

For our indexing experiments, we used three samples of 1000 documents each (disjoint from the material from which the indexing dictionary or the parameters $O(C|s_k)$ were derived). These samples with the numbers 5, 6 and 7 were drawn randomly from the input to the PHYS database. Sample 5 is always used as learning sample for the estimation of the parameters of the indexing functions. All indexing functions developed were tested with sample 6. Due to the large number of experiments performed, this sample also can be regarded as a kind of learning sample. For this reason, we finally tested a few functions with sample 7, in order to get statistically valid tests for our major findings.

Descriptors (with classifications)	22 683
Other terms	179 675
Words	85 017
Phrases	94 658
Pairs (t,s) in the relations:	
Relation Z	355 933
t = word	159 930
t = phrase	170 697
t = formula identifier	25 306
Relation USE	50 138
IDENTITY relation $t = s$	22 683
FORMULA IDENTITY relation	15 214

Table 1: Survey of the PHYS/PILOT dictionary used for our experiments

For evaluating the quality of automatic indexing, we regard the coincidence with manual indexing[3]. Since our indexing functions produce a weighted indexing (whereas the manual indexing is a binary one), we can either use measures that consider these weights, or we can regard measures for binary indexing, after application of a cutoff value. As a measure of the first type, we regard the average square error

$$s^2 = \frac{1}{t}\sum_{k=1}^{t}(y_k - a(\boldsymbol{x}_k))^2.$$

For the comparison of two binary indexings, the consistency factor

$$q = \frac{|AUT \cap MAN|}{|AUT \cup MAN|}.$$

is computed, where AUT and MAN denote the set of all automatic resp. manual descriptor assignments for a given test set of documents. By varying the cutoff values, we take the maximum q_{max} of these q values.

As test of significance, the sign test is used in the following way: For a certain test set of documents with relevance descriptions, we compare the performance of two indexing functions $a_1(\boldsymbol{x})$ and $a_2(\boldsymbol{x})$. Now we can either regard weighted indexing or binary indexing. In

[2] Some experiments with the complete set of pairs described in [Fuhr et al. 91] showed almost no difference in terms of indexing quality.

[3] Of course, retrieval results are the final yardstick for indexing quality. This approach was taken in the AIR retrieval test [Fuhr & Knorz 84]. The results of the AIR retrieval test revealed that the comparison with manual indexing is also a good measure of indexing quality, so we prefer this method because it requires substantially less effort.

the case of weighted indexing, we compare the differences $|y_k - a_1(\boldsymbol{x}_k)|$ and $|y_k - a_2(\boldsymbol{x}_k)|$ for $k = 1 \ldots t$. In order to compare binary indexing, first an average indexing depth δ (number of descriptor assignments per document) has to be chosen. From this parameter, the corresponding cutoff values $\alpha_1(\delta)$ for $a_1(\boldsymbol{x})$ and $\alpha_2(\delta)$ for $a_2(\boldsymbol{x})$ can be derived. In the test set, only those relevance descriptions $\boldsymbol{x}_k$ are regarded where the two indexing functions lead to different assignment decisions, that is either $a_1(\boldsymbol{x}_k) \geq \alpha_1(\delta) \wedge a_2(\boldsymbol{x}_k) < \alpha_2(\delta)$ or $a_1(\boldsymbol{x}_k) < \alpha_1(\delta) \wedge a_2(\boldsymbol{x}_k) \geq \alpha_2(\delta)$, and then these decisions are compared with the intellectual decision denoted by y_k. In our experiments, we varied the indexing depth from 0 to the maximum (21.9 or 27.7, respectively) in steps of 0.1 and compared the resulting binary indexings at a significance level of 99%. Then we give the indexing depths for which the difference is significant.

6 Experiments

For testing the logistic models, we performed three series of experiments:

1) Tests of the pure model with relevance descriptions where all parameters required by the model are available.
2) Extensious of the pure models that consider all the information stored in the indexing dictionary.
3) Heuristic definitions of the description vector for the comparison of logistic indexing functions and linear functions based on least square polynomials.

These experiments are described in the following. A more detailed presentation of the experimental results can be found in [Pfeifer 90].

6.1 Tests of the pure model

For this series of experiments, we only considered the descriptor indications and the relevance descriptions where all the parameters ω_{ik}, ω_{ik0}, ϱ_{ik}, β_k and $\lambda(C|s_k)$ were available (about 21 900 of a total of 27 700 relevance descriptions for a sample of 1000 documents).

name	m	description
$FD+$	11	FOC dependence model with $\omega_{ikj} = \alpha_j + \omega_{ik}$
$FD*$	14	FOC dependence model with $\omega_{ikj} = \alpha_j \cdot \omega_{ik}$
FI	14	FOC independence model
FIM	14	FOC independence model, but with $\max z(t,s)$
FIT	30	like FI, plus ID and USE relations
$FIMT$	30	like FIM, plus ID and USE relations
$FIT3$	30	like FIT, but 3 classes of rel. descr.
$FIT5$	30	like FIT, but 5 classes of rel. descr.
$FIZ-DO$	42	FIZ vector plus descriptor weights $O(C|s)$
$FIZ-DO5$	42	like FIZ-DO, but 5 classes of rel. descr.
$FIZ-O$	40	FIZ vector
$FIZ-L$	40	FIZ vector with linear function (MSE)
$FIZ-LI$	40	like FIZ–L, but iterated MSE

Table 2: Survey of indexing functions tested

no	name	description
1	$\#DIDOC$	number of descriptor indications in the document
2	$\#WODOC$	number of different words in the document giving indications
3	$\sum STI$	$\sum \omega_{ik}$ for words in the title
4	$\sum SAB$	$\sum \omega_{ik}$ for words in the abstract
5	$\sum P{+}TI$	$\sum \omega_{ik}$ for phrases (P+) in the title
6	$\sum P{-}TI$	$\sum \omega_{ik}$ for phrases (P+) in the abstract
7	$\sum P{+}AB$	$\sum \omega_{ik}$ for phrases (P+) in the title
8	$\sum P{-}AB$	$\sum \omega_{ik}$ for phrases (P+) in the abstract
9	$DESKWT$	$\lambda(C|s_k) = \log O(C|s_k)$
10	$\sum RULEAWT$	$\sum_{y_{mi} \in V} \omega_{ikO}$
11	$DESKRAW$	$\beta_k = \log \prod_{i=1}^{n} \omega_{ikO}$

Table 3: Description vector for $FD*$

We compare the two variants fo the FOC dependence model with the independence model. Table 3 shows the description vector for the FOC dependence model with the heuristic assumption $\omega_{ikj} = \alpha_j \cdot \omega_{ik}$. This indexing function is called $FD*$ here. For the alternative heuristic assumption $\omega_{ikj} = \alpha_j + \omega_{ik}$ (function $FD+$), the description vector contains elements no. 1-12 from table 4, but with the element $\sum ZWT$ defined as $\sum y_{m_i \in V}(\omega_{ik})$. In both cases (as well as with all functions for the FOC independence model described below), the first two elements of the vector serve for the estimation of the document-oriented parameter $\lambda(C|e_m)$, whereas all other elements relate to a specific relevance description. Furthermore, it should be noted that in all experiments, the description vector contains a constant element (which is not shown here). The description vector for the FOC independence model consists of elements no. 1-14 from table 4. For the sum in eqn(5), we have defined $\sum ZWT$ for ϱ_{ik}, $\sum RULEAWT$ gives ω_{ik0} and the elements $\#ZSTI$ thru $\#ZP{-}AB$ serve for the FOC-specific estimation of the parameters $\varphi_{lim} = P(v_{lim}|C)/P(v_{lim}|\bar{C})$. In order to compute $\sum 2\lambda(C)$, $\#Z$ counts the number of indications (with Z relations) in the relevance description, so the corresponding coefficient is assumed to approximate $2\lambda(C)$. In a similar way, $\#Z * DW$ computes $\sum \lambda(C|s_k)$, so its coefficient should be about 3. The results for these three indexing functions are shown in table 5. Although the differences for the evaluation parameters s^2 and q_{max} are fairly small (as for all our ex-

no	name	description
1	$\#DIDOC$	number of descriptor indications in the document
2	$\#WODOC$	number of different words in the document giving indications
3	$\sum ZWT$	$\sum_{y_{m\imath} \in V} \varrho_{ik} = \log \prod_{y_{m\imath} \in V} O(C\|t_\imath, s_k)$
4	$\#ZSTI$	n.i.w. Z relation from word in the title
5	$\#ZSAB$	n.i.w. Z relation from word in the abstract
6	$\#ZP{+}TI$	n.i.w. Z relation from phrase (P+) in the title
7	$\#ZP{-}TI$	n.i.w. Z relation from phrase (P-) in the title
8	$\#ZP{+}AB$	n.i.w. Z relation from phrase (P+) in the abstract
9	$\#ZP{-}AB$	n.i.w. Z relation from phrase (P-) in the abstract
10	$\sum RULEAWT$	$\sum_{y_{m\imath} \in V} \omega_{\imath k0}$
11	$DESCWT$	$\lambda(C\|s_k) = \log O(C\|t_\imath, s_k)$
12	$DESCRAW$	$\beta_k = \sum_{\imath=1}^{n} \omega_{\imath k0}$
13	$\#Z$	n.i.w. Z relations
14	$\#Z{*}DW$	$= \#Z \cdot DESCWT = \sum_{y_{m\imath} \in V} \lambda(C\|s_k)$
15	$?DESCWT$	=1, if $DESCWT$ available, =0 otherwise
16	$\#NRA$	# indications without weight $\omega_{\imath k0}$
17	$?DESCRAW$	=1, if $DESCRAW$ available, =0 otherwise
18	$?\sum RAWT$	=1, if $\sum RULEAWT \neq 0$, =0 otherwise
19	$\#ISTI$	n.i.w. ID relation from word in the title
20	$\#ISAB$	n.i.w. ID relation from word in the abstract
21	$\#IP{+}TI$	n.i.w. ID relation from (P+) in the title
22	$\#IP{-}TI$	n.i.w. ID relation from (P-) in the title
23	$\#IP{+}AB$	n.i.w. ID relation from (P+) in the abstract
24	$\#IP{-}AB$	n.i.w. ID relation from (P-) in the abstract
25	$\#USTI$	n.i.w. USE relation from word in the title
26	$\#USAB$	n.i.w. USE relation from word in the abstract
27	$\#UP{+}TI$	n.i.w. USE relation from (P+) in the title
28	$\#UP{-}TI$	n.i.w. USE relation from (P-) in the title
29	$\#UP{+}AB$	n.i.w. USE relation from (P+) in the abstract
30	$\#UP{-}AB$	n.i.w. USE relation from (P-) in the abstract

(n.i.w. = number of indications with)

Table 4: Description vector for the functions FI, FIT, $FIT3$ and $FIT5$

name	sample 6		sample 7	
	s^2	q_{max}	s^2	q_{max}
$FD+$	0.134	0.422	0.131	0.414
$FD*$	0.134	0.419	0.131	0.412
FI	0.128	0.429	0.126	0.412
FIM	0.126	0.432	0.124	0.428
FIT	0.113	0.452	0.111	0.449
$FIMT$	0.112	0.451	0.109	0.451
$FIT3$	0.106	0.470	0.103	0.469
$FIT5$	0.102	0.476	0.100	0.476
$FIZ{-}DO$	0.110	0.470	0.109	0.465
$FIZ{-}DO5$	0.102	0.476	0.101	0.471
$FIZ{-}O$	0.113	0.463	0.111	0.459
$FIZ{-}L$	0.117	0.452	0.113	0.451
$FIZ{-}LI$	0.113	0.464	0.111	0.458

Table 5: Indexing results

periments described here), the statistical tests show that most of these differences are significant. With the FOC dependence model, $FD+$ performs significantly better (> 99.99%) for weighted indexing, but only for a few indexing depths ($\delta = 7.3, 7.4, 9.0, 9.8$) when we regard binary indexing. The FOC independence model outperforms both variants of the dependence model, with a significance level > 99.99% for weighted indexing and at most indexing depths in the case of binary indexing. As a heuristic variant of the FI function, the function FIM was developed. This function is based on the same decription vector as FI. However, $\sum ZWT$ was redefined here in order to get the maximum value $\max_{y_{m\imath} \in V} \varrho_{ik}$ instead of the sum of these values. Furthermore, $\sum RULEAWT$ was assigned only the weight ω_{ik0} from the indication with the maximum ϱ_{ik} value. This heuristic strategy has been applied successfully for the development of least square polynomial indexing functions ([Knorz 83b] [Knorz 86]; see also below). One can regard this method as a means for coping with the problem of statistical dependence of the descriptor indications within a relevance description. In fact, the experimental results show a slight improvement for weighted indexing, and for binary indexing with indexing depths in the intervals [7.1 . . . 9.9] [10.1 . . . 10.3] and [10.5 . . . 11.5].

6.2 Extensions of the pure model

Since our dictionary contains the USE and the ID relation besides the Z relation, we have to extend the description vector in order to consider this information, too. In contrast to the relation Z, there is no weight associated with these thesaurus relations. For this reason, we assume that the weights ω_{ikj} are the same for all term-descriptor pairs (t_i, s_k) in one of these relations. So we have six different weights (for our six FOCs) for the relation USE and another six weights for the ID relation. Furthermore, we assume that $\omega_{ik0} = 0$ for these relations. These assumptions lead us to the description vector shown in table 4, where we have six elements for either of these relations. Furthermore, the elements $?DESKWT, \#NRA, ?DESKRAW$ and $\#Z{-}ABS$ have been defined in order to cope with indications based on the relation Z for which the parameters ω_{ik0} or β_k are not available. So we have a combination of the FOC dependence and the FOC independence model here: For the relation Z, we assume independence, since we do not have the FOC-specific weights. With the relations ID and USE, no term-descriptor-specific weights are available, but we can aim at estimating FOC-specific weights here. The results of this indexing function called FIT are shown in table 5. In comparison to the experiments based on the relation Z only, we get a clear improvement.

The function $FIMT$ is heuristic variation of FIT, where we regard the maximum of the weights ϱ_{ik} instead of their sum (like in the case of FIM). Again, we achieve slight improvements which are significant (> 95%) for weighted indexing.

The next two experiments in this series are based on a subdivision of the relevance descriptions into 3 resp. 5 classes. According to the relations on which the indications of a relevance descriptions are based, we defined 5 classes:

T : 38% of all relevance descriptions are based exclusively on thesaurus relations.

$Z1$: 23% of all relevance descriptions consist of exactly one indication, which is based on the relation Z.

$Z2$: 11% of all relevance descriptions consist of at least two indications, which are all based on the relation Z.

$TZ1$: 12% of all relevance descriptions contain exactly one indication derived from the relation Z, plus one or more indications based on thesaurus relations.

$TZ2$: 15% of all relevance descriptions contain at least two indications derived from the relation Z, plus one or more indications based on thesaurus relations.

For each of these classes, separate indexing functions (based on the description vector of the function FIT) were developed. This combined function is called $FIT5$. We also tested a variant ($FIT3$) with 3 classes only, where the distinctions between the classes $Z1$ and $Z2$ and between $TZ1$ and $TZ2$ were dropped. The experimental results show large improvements for $FIT3$ over FIT, and a further slight improvement for $FIT5$. In both cases, the difference is highly significant for weighted indexing (> 99.99%). For binary indexing, $FIT3$ performs significantly better than FIT at the indexing depths 2.3, 2.5 and 2.8 thru 16.9. 5 classes are better than 3 classes at $\delta \in [5.9\ldots6.9], [7.2\ldots10.2], [10.9\ldots16.4]$. These results show that – in comparison to the other strategies tested – using class-specific instead of general indexing functions is the best method for improving indexing quality.

6.3 Heuristic description vectors

The indexing functions described so far are based more or less on the probabilistic indexing model as described in section 2. Here we want to consider a heuristic strategy for the description vector, in order to see whether there are significant differences between the indexing results of these two strategies.

Instead of developing a completely new description vector, we choose to use the description vector that had been developed heuristically for the least square polynomials indexing function, as described in [Knorz 86]. This description vector is shown in table 6. With the exception of the elements $DESKWT$ and $?DESKWT$, this vector is currently used in the linear indexing function $FIZ{-}L$ for the input production of the database PHYS. The logistic indexing function $FIZ{-}DO$ based on this vector performs significantly better than our more theoretic function FIT. With class-specific functions, however, $FIZ{-}DO5$ produces results that are about identical with those of $FIT5$. In the following experiments, we compare logistic and linear indexing functions. Here the elements $DESKWT$ and $?DESKWT$ were excluded from the description vector. The logistic function $FIZ{-}O$ shows significantly better results than the linear function $FIZ{-}L$. A closer examination of the properties of the least square polynomials procedure revealed the following weakness of this approach: in contrast to the logistic functions, linear functions may yield estimates outside the interval $[0, 1]$. Even if these estimates are on the "correct side" of the interval (that is, $a(\boldsymbol{x}_k) > 1$ and $y_k = 1$ or $a(\boldsymbol{x}_k) < 0$ and $y_k = 0$), these estimates are treated as errors, and the LSP procedure aims at minimizing the error $(a(\boldsymbol{x}_k) - y_k)^2$. In order to overcome this problem, we first developed a linear function $a_1(\boldsymbol{x}_k)$, then we removed all relevance descriptions with $a_1(\boldsymbol{x}_k) \notin [0, 1]$ from the learning sample, and developed a new linear function. This process of reducing the learning sample and developing a new function can be repeated several times, and we achieved improvements for up to 10 iterations[4] [Pfeifer 91]. The experimen-

[4] The computational effort for each of these iterations is about

no	name	description
1	*#WODOC*	number of different words in the document giving indications
2	*#RDDOC*	number of relevance descriptions of the document
3	*#DI*	number of descriptor indications of the RD
4	*#WORDS*	number of different words in the RD
5	*#PHRASES*	number of different phrases in the RD
6	*#ID*	n.i.w. *ID* relation
7	*#ZS5*	n.i.w. *Z* relation from word, where $h(t,s) \geq 5$
8	*Z5*	maximum $z(t,s)$, where $h(t,s) \geq 5$
9	*ZS*	max $z(t,s)$ from word or phrase, where $h(t,s) \geq 5$
10	*Z2S*	2nd largest $z(t,s)$ from word or phrase, where $h(t,s) \geq 5$
11	*?IDONLY*	=1, if *RD* contains only *ID* relation, =0 otherwise
12	*?P+*	=1, if indication with *FOC P+* occurring in title on abstract
13	*#ZFO*	n.i.w. *Z* relation from formula identifier
14	*Z5T+AB*	max. $z(t,s)$ from word or *P+* in the abstract, where $h(t,s) \geq 5$
15	*Z5T+*	max. $z(t,s)$ from word or *P+*, where $h(t,s) \geq 5$
16	*Z3T+*	max. $z(t,s)$ from word or *P+*, where $h(t,s) < 5$
17	*ZT+*	max. $z(t,s)$ from word or *P+*
18	*HZT+*	$h(t,s)$ for $z(t,s)$ stored in *ZT+*
19	*#ZT+*	n.i.w. *Z* relation from word or *P+*
20	*#Z5PTI*	n.i.w. *Z* relation from phrase in the title
21	*#ISTI*	n.i.w. *ID* relation from word in the title
22	*#IFTI*	n.i.w. *ID* relation from formula in the title
23	*#ISAB*	n.i.w. *ID* relation from word in the abstract
24	*#IFAB*	n.i.w. *ID* relation from formula in the abstract
25	*#ID+TI*	n.i.w. *ID* relation from phrase *P+* in the title
26	*#IPTI*	n.i.w. *ID* relation from phrase in the title
27	*#IP+AB*	n.i.w. *ID* relation from phrase *P+* in the abstract
28	*#IPAB*	n.i.w. *ID* relation from phrase in the abstract
29	*Z3STI*	max. $z(t,s)$ from word in the title, where $h(t,s) < 5$
30	*Z5STI*	max. $z(t,s)$ from word in the title, where $h(t,s) \geq 5$
31	*Z3SAB*	max. $z(t,s)$ from word in the abstract, where $h(t,s) < 5$
32	*Z5SAB*	max. $z(t,s)$ from word in the abstract, where $h(t,s) \geq 5$
33	*Z3P+TI*	max. $z(t,s)$ from *P+* in the title, where $h(t,s) < 5$
34	*Z5P+TI*	max. $z(t,s)$ from *P+* in the title, where $h(t,s) \geq 5$
35	*Z3PTI*	max. $z(t,s)$ from phrase in the title, where $h(t,s) < 5$
36	*Z5PTI*	max. $z(t,s)$ from phrase in the title, where $h(t,s) \geq 5$
37	*Z3P+AB*	max. $z(t,s)$ from *P+* in the abstract, where $h(t,s) < 5$
38	*Z5P+AB*	max. $z(t,s)$ from *P+* in the abstract, where $h(t,s) \geq 5$
39	*Z3PAB*	max. $z(t,s)$ from phrase in the abstract, where $h(t,s) < 5$
40	*Z5PAB*	max. $z(t,s)$ from phrase in the abstract, where $h(t,s) \geq 5$
41	*DESKWT*	$\lambda(C\|s_k)$
42	*?DESKWT*	=1, if value for *DESKWT* available, =0 otherwise

Table 6: Description vector for indexing functions *FIZ* ...

tal results for this iterated linear function *FIZ−LI* are about the same as for the logistic function *FIZ−O*.

7 Summary and conclusions

In this paper, we have derived a new probabilistic indexing model, which serves as a basis for developing logistic indexing functions. We have shown that logistic functions can be applied as indexing functions, and that the definition of description vectors based on the theoretical model is a partially successful strategy. However, additional heuristic strategies – such as the development of class-specific functions – may yield large improvements. In comparison to a purely heuristic strategy, only the combination of heuristics and theory produces about the same indexing quality. No significant differences were found between logistic and linear (iterated) functions.

All these findings, however, refer to the specific application tested here. In contrast to the experiments with logistic models described in [Robertson & Bovey 82], we have very large learning samples, even in relation to the number of parameters to be estimated. The number of elements of the description vector also may affect the almost identical performance of linear and logistic functions. For smaller numbers of parameters to be estimated, the difference between linear and logistic functions may be important. For example, in the experiments described in [Fuhr & Buckley 91], indexing functions are based on about 5 parameters, and the learning samples available do not allow a larger number of parameters. So more experimental work is needed on

the same as for one iteration with the Newton-Raphson method for logistic functions, so these two approaches require about the same effort logistic functions in order to fill the gap between the results of Robertson and Bovey and the work described here.

Acknowledgement

We wish to thank Abe Bookstein who pointed us towards logistic functions.

References

Fienberg, S. (1980). *The Analysis of Cross-Classified Categorial Data.* MIT Press, Cambridge, Mass., 2. edition.

Freeman, D. (1987). *Applied Categorial Data Analysis.* Dekker, New York.

Fuhr, N.; Buckley, C. (1990). Probabilistic Document Indexing from Relevance Feedback Data. In: Vidick, J.-L. (ed.): *Proceedings of the 13th International Conference on Research and Development in Information Retrieval,* pages 45–61. ACM, New York.

Fuhr, N.; Buckley, C. (1991). A Probabilistic Learning Approach for Document Indexing. *ACM Transactions on Information systems 9(2).*

Fuhr, N.; Knorz, G. (1984). Retrieval Test Evaluation of a Rule Based Automatic Indexing (AIR/PHYS). In: Van Rijsbergen, C. (ed.): *Research and Development in Information Retrieval,* pages 391–408. Cambridge University Press, Cambridge.

Fuhr, N. (1989a). *Log-Linear Indexing Functions and the Darmstadt Indexing Approach.* Internal Report DV II 89-4, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II.

Fuhr, N. (1989b). Models for Retrieval with Probabilistic Indexing. *Information Processing and Management 25(1),* pages 55–72.

Fuhr, N. (1989c). Optimum Polynomial Retrieval Functions Based on the Probability Ranking Principle. *ACM Transactions on Information Systems 7(3),* pages 183–204.

Fuhr, N.; Hartmann, S.; Knorz, G.; Lustig, G.; Schwantner, M.; Tzeras, K. (1991). AIR/X - a Rule-Based Multistage Indexing System for Large Subject Fields. In: *Proceedings of the RIAO'91, Barcelona, Spain, April 2-5, 1991.*

Knorz, G. (1983a). *Automatisches Indexieren als Erkennen abstrakter Objekte.* Niemeyer, Tübingen.

Knorz, G. (1983b). *Development of Automatic Indexing for the AIR Retrieval Test. Experiments by means of ALIBABA.* Report DVII 83-3, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II.

Knorz, G. (1986). Die Anwendung von Polynomklassifikatoren für die automatische Indexierung. In: Lustig, G. (ed.): *Automatische Indexierung zwischen Forschung und Anwendung,* pages 98–126. Olms, Hildesheim.

Pfeifer, U. (1990). *Development of Log-Linear and Linear-Iterative Indexing Functions (in German).* Diploma thesis, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II.

Pfeifer, U. (1991). Entwicklung linear iterativer und logistischer Indexierungsfunktionen. In: Fuhr, N. (ed.): *Information Retrieval.* Springer, Berlin et al.

van Rijsbergen, C. (1977). A Theoretical Basis for the Use of Co-Occurrence Data in Information Retrieval. *Journal of Documentation 33,* pages 106–119.

Robertson, S.; Bovey, J. (1982). *Statistical Problems in the application of probabilistic models to information retrieval.* Report 5739, British Library, London.

Robertson, S.; Sparck Jones, K. (1976). Relevance Weighting of Search Terms. *Journal of the American Society for Information Science 27,* pages 129–146.

Yu, C.; Salton, G. (1976). Precision Weighting. An Effective Automatic Indexing Method. *Journal of the ACM 23,* pages 76–88.

Yu, C.; Buckley, C.; Lam, K.; Salton, G. (1983). A Generalized Term Dependence Model in Information Retrieval. *Information Technology: Research and Development 2,* pages 129–154.