A Probabilistic Learning Approach for Document Indexing

NORBERT FUHR
TH Darmstadt

and

CHRIS BUCKLEY
Cornell University

We describe a method for probabilistic document indexing using relevance feedback data that has been collected from a set of queries. Our approach is based on three new concepts: (1) Abstraction from specific terms and documents, which overcomes the restriction of limited relevance information for parameter estimation. (2) Flexibility of the representation, which allows the integration of new text analysis and knowledge-based methods in our approach as well as the consideration of document structures or different types of terms. (3) Probabilistic learning or classification methods for the estimation of the indexing weights making better use of the available relevance information. Our approach can be applied under restrictions that hold for real applications. We give experimental results for five test collections which show improvements over other methods.

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*; I.2.6 [**Artificial Intelligence**]: Learning—*parameter learning*

General Terms: Experimentation, Theory

Additional Key Words and Phrases: Complex document representation, linear indexing functions, linear retrieval functions, probabilistic indexing, probabilistic retrieval, relevance descriptions

1. INTRODUCTION

Document indexing is the task of assigning terms to documents for retrieval purposes. In an early paper on probabilistic retrieval [21], an indexing model was developed based on the assumption that a document should be assigned

This study was supported in part by the National Science Foundation under grant IRI 87-02735.
Authors' addresses: W. Fuhr, Universität Dortmund, Informatic VI, Postfach 500500, W-4600 Dortmund, Germany; C. Buckley, Cornell University, Department of Computer Science, 405 Upson Hall, Ithaca, NY 14853-7501.

those terms that are used by queries to which the document is relevant. With this model, the notion of weighted indexing (instead of binary indexing), that is, the weighting of the index terms with respect to the document, was given a theoretical justification in terms of probabilities. Fuhr [13] generalizes this approach to all models of probabilistic indexing by introducing the concept of "correctness" as the event to which the probabilities relate.

The Maron and Kuhns model assumes that the probabilistic indexing weights for a document can be estimated on the basis of relevance information from a number of queries with relation to the specific document. However, in real applications there is hardly ever enough relevance information for a specific document available in order to estimate the required probabilities. For this reason, retrospective experiments based on this model (or related ones) might show its feasibility [18, 15], but are of little value with regard to real applications. The model described by Kwok [17] overcomes this problem by regarding document components as units to which the index term weights relate. However, experimental evaluations showed that this model is inferior to nonprobabilistic indexing approaches [19]. A different model for using probabilistic indexing weights in retrieval is described by Robertson et al. [26] as the "2-Poisson-independence" model, but also had little success (mainly because of parameter estimation problems). In contrast to these results, the approaches developed by Croft [6, 7] and Wong and Yao [35] show improvements over binary indexing; however, these models lack an explicit notion of an event to which the probabilistic weights relate.

In this paper, we present a radically different approach to probabilistic indexing. We introduce the concept of "relevance description" as an abstraction from specific term-document relationships. As different term-document pairs may have the same relevance description, we overcome the problems of parameter estimation mentioned above by estimating probabilities for relevance descriptions instead of specific term-document pairs. Furthermore, this concept is flexible, with relation to the representation of documents. For the computation of the indexing weights, we use probabilistic classification procedures instead of simple estimation schemes.

In the following, we first show how the basic ideas of our approach differ from other work in IR. For this purpose, we discuss the relationship between representations and models in IR by introducing a new conceptual model, and we regard probabilistic models as machine learning approaches. In Section 3, we give a brief introduction into the binary independence indexing model, which forms the theoretical justification for our probabilistic indexing weights. Then we describe the concepts and procedures of our indexing approach. Section 5 outlines the test setting and the parameters investigated in our experiments, followed by the presentation of the experimental results in Section 6.

2. MODELS AND REPRESENTATIONS IN INFORMATION RETRIEVAL

In this section, we first present a conceptual model for information retrieval (IR). This model helps to classify existing IR models and to describe the role

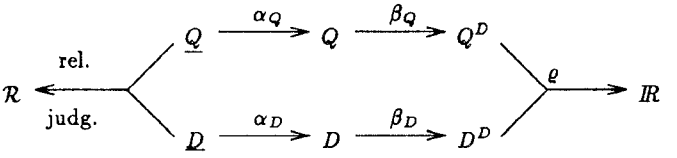

Fig. 1. Conceptual model.

of the representations of queries and documents in these models. In probabilistic IR models, parameters of the models relate to elements of the underlying representations. In order to estimate these parameters, relevance feedback data is used. We show that there have been two general approaches for this kind of parameter learning, whereas the work presented in this paper can be regarded as a new approach that overcomes most of the deficiencies of the older ones.

Our conceptual model is presented in Figure 1: An IR system contains a finite set of documents $\underline{D} = \{\underline{d}_1, \underline{d}_2, \underline{d}_3, \ldots\}$. Let $\underline{Q} = \{\underline{q}_1, \underline{q}_2, \underline{q}_3, \ldots\}$ be the (possibly infinite) set of queries submitted to the system. Here we regard queries as unique events, that is, if two users submit the same query statement, they are treated as different queries. The same approach is taken in the unified model [24], where a single query is termed an "individual use." Between a query and a document, there exists a relevance relationship as specified by the user who submitted the query. Let $\mathcal{R} = \{R, \bar{R}\}$ (relevant/nonrelevant) denote the set of possible relevance judgements[1], then the relevance relationship can be regarded as a mapping r: $\underline{Q} \times \underline{D} a \mathcal{R}$.

As IR systems can only have a limited understanding of documents and queries, they are based on sets of representations D and Q of these objects. With the mapping α_D, document representations D are derived from the original documents $\underline{D}$. In the same way, α_Q maps queries from $\underline{Q}$ onto their representations in Q. With regard to IR models, we can give a more specific description of the concept of representation: a representation of a document or a query denotes the data that is actually used for the retrieval task. For example, in the well-known binary independence retrieval (BIR) model [37, 25], documents are represented as sets of terms. Thus, two documents with the same set of terms will be mapped onto the same representation. In this model, a query representation q_k also consists of a set of terms q_k^T, but in addition, relevance information about some documents (with respect to the current query) is also included in the query representation. The relevance data for a query q_k is a multiset[2] $q_k^J = [(r_{kj}, d_j) \mid r_{kj} \in \mathcal{R} \wedge d_j \in D]$ of pairs of relevance judgements and document representations. Thus, two queries $\underline{q}_1$ and $\underline{q}_2$ have the same representation if $q_1^T = q_2^T$ and $q_1^J = q_2^J$.

For retrieval, the sets of representations Q and D are not compared directly because this would in many cases be too complex. For this reason,

[1] For multivalued relevance scales, see Fuhr [14].

[2] Different documents may have the same representation, but they should occur as separate elements in q_k^J, even if they are given the same relevance judgement.

the representations Q and D are transformed by the mappings β_Q and β_D onto the descriptions Q^D and D^D. These descriptions are the arguments of the retrieval function ϱ: $Q^D \times D^D \rightarrow \mathbb{R}$ which maps descriptions of query-document pairs onto the set of real numbers, where $\varrho(q_k^D, d_m^D)$ is called the relevance status value. In response to a query q_k, documents $\underline{d}_j \in \underline{D}$ are ranked according to descending values $\varrho(q_k^D, d_j^D)$. In the case of the binary independence retrieval model, the description and the representation of a document are identical (a set of terms), while the description of a query is a set of weighted terms.

For the comparison of different IR models, the underlying representations play an important role: only models based on identical representations are directly comparable. So a major research goal is the development of a model that produces the best retrieval quality for a given representation. On the other hand, one may use more detailed representations of documents and queries in order to improve retrieval quality (e.g., by regarding the within-document or within-query frequency of terms or by using phrases in addition to single words as terms). Most approaches in this direction require the development of new models that fit to the underlying representations. In this paper we describe a model that is representation-independent to a certain extent, and thus it can be combined with the best representation available.

IR models also can be classified according to the mappings of our conceptual model that they attempt to improve. In addition to the model-specific retrieval function, most IR models aim at optimizing either β_D or β_Q, while the other mapping is fixed. The BIR model and many refinements of this model try to optimize β_Q. The approach taken in this paper concentrates on the improvement of β_D.

As mentioned in the beginning of this section, probabilistic IR models can be regarded as parameter learning methods: in order to estimate the probabilistic parameters of a model, relevance feedback data is needed. Figure 2 shows three different learning approaches that are used in IR. The three axes indicate to what kinds of objects probabilistic parameters may relate: documents, queries and terms (that is, elements of the representation). In each of the three approaches, we can distinguish a learning phase and an application phase: In the learning phase, we have relevance feedback data for a certain subset $Q_L \times D_L \times T_L$ of $Q \times D \times T$ (where T denotes the set of terms in the collection) from which we can derive probabilistic parameters. These parameters can be used in the application phase for the improvement of the descriptions of documents and queries.

In the first type of learning, relevance feedback data is used for weighting of search terms (e.g., in the BIR model) with respect to a single query (representation) q_k. Here we have relevance information from a set of documents D_L, and we can estimate parameters for the set of terms T_L occurring in these documents. In the application phase, we are restricted to the same query q_k and the set of terms T_L, but we can apply our model to all documents in D.

The second type of learning is orthogonal to the first approach: probabilistic indexing models (e.g., the one described by Maron and Kuhns [21]) collect

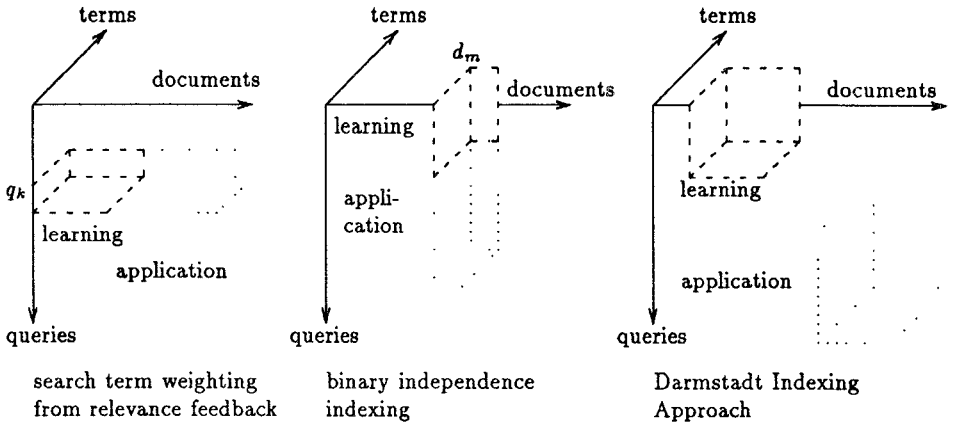

Fig. 2. Learning approaches in IR.

relevance feedback data for a specific document d_m from a set of queries Q_L with the set of terms T_L occurring in these queries. The parameters derived from this data can be used for the same document and the same set of terms T_L (occurring in queries) only, but for all queries submitted to the system. The major problem with this approach, however, is the fact that there are not enough relevance judgements for a single document in real databases, so it is almost impossible to estimate the parameters in this approach.

The major drawback of these two learning approaches is their limited application range: in the case of search term weighting from relevance feedback, the relevance information collected for one query is worthless for any other query. In the same way, the probabilistic indexing approach restricts the use of relevance data to a single document. The Darmstadt Indexing Approach [13, 3] overcomes these deficiencies by introducing the concept of relevance descriptions: a relevance description is an abstraction from specific queries, documents and terms. Like in pattern recognition methods, a relevance description contains values of features of the objects under consideration (queries, documents and terms). In the learning phase, parameters relating to these features are derived from the learning sample $Q_L \times D_L \times T_L$. For the application phase, there are no restrictions concerning the subset $Q_A \times D_A \times T_A$ of objects to which these parameters can be applied: new queries as well as new documents and new terms can be considered. This strategy is a kind of long-term learning method, since feedback data can be collected from all queries submitted to the IR system, thus increasing the size of the learning sample over time; as a consequence, the probability estimates can be improved. This approach [14] has been taken for the development of retrieval functions.[3] In the following section, we describe the application of this learning approach to the task of document

[3]Another long-term learning method has been presented by Yu and Mizuno [36], but instead of a general abstraction only a single feature is regarded in this work.

indexing, that is, the improvement of the mapping β_D of our conceptual model.

3. THE BINARY INDEPENDENCE INDEXING MODEL

As described in the previous section, let $\underline{Q}$ denote the set of queries and $\underline{D}$ the set of documents in the collection, and Q and D are the corresponding sets of representations. Then the event space of the BII model is $Q \times \underline{D}$, and the query representations are sets of terms. As a consequence, the BII model will yield the same ranking for two different queries which use the same set of terms. With $T = \{t_1, \ldots, t_n\}$ as the set of index terms in our collection, the query representation q_k of a query $\underline{q_k}$ is a subset $q_k^T \subset T$. Below, we will also use a binary vector $\vec{z}_k = (z_{k1}, \ldots, z_{kn})$ instead of q_k^T, where $z_{k\iota} = 1$, if $t_\iota \in q_k^T$, and $z_{k\iota} = 0$ otherwise. The document representation is not further specified in the BII model, and below we will show that this is a major advantage of this model. In the following, we will assume that there exists a set $d_m^T \subset T$ of terms which are to be given weights with relation to the document. For brevity, we will call d_m^T "the set of terms occurring in the document" in the following, although the model also can be applied in situations where the elements of d_m^T are derived from the document text with the help of a dictionary or knowledge base (see Fuhr [13]). Let us further assume that we have a binary relevance scale $\mathcal{R} = \{R, \bar{R}\}$ denoting relevant/nonrelevant query-document relationships. Then each element $(\underline{q_k}, \underline{d_m})$ of the event space has associated with it the sets q_k^T, d_m^T and a relevance judgement $r_{km} = r(\underline{q_k}, \underline{d_m}) \in \mathcal{R}$.

The BII model now seeks for an estimate of the probability $P(R \mid q_k, d_m) = P(R \mid z_k, d_m)$ that a document with the representation d_m will be judged relevant w.r.t. a query with the representation $q_k = q_k^T$. Applying Bayes' theorem, we first get

$$P(R \mid \vec{z}_k, d_m) = P(R \mid d_m) \cdot \frac{P(\vec{z}_k \mid R, d_m)}{P(\vec{z}_k \mid d_m)}. \tag{1}$$

Here $P(R \mid d_m)$ is the probability that document d_m will be judged relevant to an arbitrary request. $P(\vec{z}_k \mid R, d_m)$ is the probability that d_m will be relevant to a query with representation $\vec{z}_k$, and $P(\vec{z}_k \mid d_m)$ is the probability that such a query will be submitted to the system.

Regarding the restricted event space consisting of all documents with the same representation d_m and all queries in the collection, we assume that the distribution of terms in all queries to which a document with representation d_m is relevant is independent.[4]

$$P(\vec{z}_k \mid R, d_m) = \prod_{\iota=1}^{n} P(z_{k_\iota} \mid R, d_m).$$

[4]Fuhr [13] used an additional assumption for the derivation of the BII model. However, as Bill Cooper mentioned to us, it can be shown that these two assumptions are incompatible (see also Robertson et al. [24] for a similar problem)

With this assumption, (1) can be transformed into

$$P(R \mid \vec{z}_k, d_m) = \frac{P(R \mid d_m)}{P(\vec{z}_k \mid d_m)} \cdot \prod_{\iota=1}^{n} P(z_{k_\iota} \mid R, d_m)$$

$$= \frac{P(R \mid d_m)}{P(\vec{z}_k \mid d_m)} \cdot \prod_{\iota=1}^{n} \frac{P(R \mid z_{k_\iota}, d_m)}{P(R \mid d_m)} \cdot P(z_{k_\iota} \mid d_m).$$

Since we always regard all documents with relation to a query, the probabilities $P(\vec{z}_k \mid d_m)$ and $P(z_{k\iota} \mid d_m)$ are independent of a specific document, to get

$$P(R \mid \vec{z}_k, d_m) = \frac{\prod_{\iota=1}^{n} P(z_{k_\iota})}{P(\vec{z}_k)} \cdot P(R \mid d_m) \cdot \prod_{\iota=1}^{n} \frac{P(R \mid z_{k_\iota}, d_m)}{P(R \mid d_m)}$$

$$= \frac{\prod_{\iota=1}^{n} P(z_{k_\iota})}{P(\vec{z}_k)} \cdot P(R \mid d_m) \cdot \prod_{z_{k_\iota}=1} \frac{P(R \mid z_{k_\iota} = 1, d_m)}{P(R \mid d_m)}$$

$$\cdot \prod_{z_{k_\iota}=0} \frac{P(R \mid z_{k_\iota} = 0, d_m)}{P(R \mid d_m)}. \tag{2}$$

Now we make an additional simplifying assumption that is also used by Maron and Kuhns [21]: The relevance of a document with representation d_m with respect to a query q_k depends only on the terms from q_k^T, and not on other terms. This assumption means that the last product in formula (2) has the value 1 and thus it can be omitted.

The value of the first fraction in this formula is a constant c_k for a given query q_k, so there is no need to estimate this parameter for a ranking of documents with respect to q_k.

$P(R \mid z_{k\iota} = 1, d_m) = P(R \mid t_\iota, d_m)$ is the probabilistic index term weight of t_ι with relation to d_m, the probability that document d_m will be judged relevant to an arbitrary query, given that it contains t_ι. From our model, it follows that d_m^T should contain at least those terms from T for which $P(R \mid t_\iota, d_m) \neq P(R \mid d_m)$. Assuming that $P(R \mid t_\iota, d_m) = P(R \mid d_m)$ for all $t_\iota \notin d_m^T$, we get the final BII formula[5]

$$P(R \mid q_k, d_m) = c_k \cdot P(R \mid d_m) \cdot \prod_{t_\iota \in q_k^T \cap d_m^T} \frac{P(R \mid t_\iota, d_m)}{P(R \mid d_m)}. \tag{3}$$

In this form it is nearly impossible to apply the BII model, because there hardly will be enough relevance information available to estimate the

[5]In contrast to this assumption, experiments described by Turtle [32, pp. 127–132] with indexing weights also assigned to query terms not occurring in the document have shown significant improvements in comparison to the case where these terms are ignored. For the experiments described in this paper, this is a pragmatic assumption. We could apply our indexing approach to terms not occurring in the documents as well.

probabilities $P(R \mid t_i, d_m)$ for specific term-document pairs. All attempts in this direction are doomed to fail ([20, 18]).

4. NEW INDEXING CONCEPTS

The basic ideas for our new approach stem from the Darmstadt Indexing Approach (DIA) [13, 2]. This approach has been developed for automatic indexing with a prescribed indexing vocabulary. We will show how the concepts developed within the DIA can be applied to all kinds of probabilistic indexing.

In the DIA, the indexing task is subdivided in a description step and a decision step. In the description step, *relevance descriptions* for term document pairs (t_i, d_m) are formed. Similar to pattern recognition approaches, a relevance description comprises a set of features that are considered to be important for the task of assigning weights to terms with relation to documents. So a relevance description $x(t_i, d_m)$ contains values of attributes of the term t_i, the document d_m and their relationship. Our approach makes no additional assumptions about the choice of the attributes and the structure of x. For this reason, the concrete definition of relevance descriptions can be adapted to the specific application context. Examples for possible elements of x are

—dictionary information about t_i, e.g. its inverse document frequency,
—parameters describing d_m, e.g., its length or the number of different terms in it,
—information about the *form of occurrence* of t_i in d_m (see Fuhr [13]), e.g., the parts of the document in which t_i occurs (*title vs. abstract*), the within-document-frequency of t_i in d_m, or in the case of t_i being a noun phrase, the word distance in d_m between the first and the last component of t_i.

In the decision step, a probabilistic index term weight based on this data is assigned. This means that we estimate instead of $P(R \mid t_i, d_m)$ the probability $P(R \mid x(t_i, d_m))$. In the former case, we would have to regard a single document d_m with respect to all queries containing t_i in order to estimate $P(R \mid t_i, d_m)$. Now we regard the set of all query-document pairs in which the same relevance description x occurs. Here the probability $P(R \mid x(t_i, d_m))$ is the probability that a document will be judged relevant to an arbitary query, given that one of the document's index terms which also occurs in the query has the relevance description x.

There are two advantages from the introduction of the concept of relevance description:

(1) By abstracting from specific document-term pairs, we do not need relevance information about the specific document d_m or the specific term t_i for the estimation of $P(R \mid x(t_i, d_m))$. According to the definition of the relevance description, document-term pairs with different documents or terms can be mapped onto the same relevance description. For this reason

we can use relevance information from other documents or even from queries q_k with $t_\iota \notin q_k^T$ for the estimation of $P(R \mid x(t_\iota, d_m))$, too. This way, the amount of relevance data that is available for the estimation of a specific indexing weight is not restricted by the number of queries for the specific document (or documents for the specific query) for which we have relevance information. In a system running in an application, the amount of relevance data from which the indexing weights are computed will always increase and therefore improve the probability estimates.

(2) Relevance descriptions can be defined for different forms of representation. Since most other probabilistic IR models are based on a specific form of representation of documents or queries, for every new form of representation a different model has to be developed. In our approach, the independence from a specific form of representation offers the following possibilities:
   —The representations can be adapted to the amount of relevance information that is currently available: the more data we have, the more detailed we can choose our representations.
   —We can consider new forms of representations that are based on techniques from artificial intelligence or computational linguistics. Now the restricted view of regarding a document as a set of terms with multiple occurrences can be abandoned (some concepts for a more detailed document representation are described by Fuhr [13]). On the other hand, our approach provides a solid theoretical background and an easy-to-apply method for the effective integration of these new types of representation in IR.
   —We can develop relevance descriptions for different types of terms or documents. Several authors have investigated the benefit of using noun phrases in addition to single words as index terms [28, 5, 30, 8, 9, 29]. However, none of them could devise a theoretical basis for the computation of document-oriented probabilistic index term weights for this new type of terms. The probabilistic foundation of our approach gives us a kind of objective weighting scheme for all types of terms. In a similar way, one could differentiate between several types of documents that are stored in the same database. This possibility of handling heterogeneous document collections becomes important in new application areas of IR systems, e.g., in the office environment.

In the decision step, estimates of the probabilistic index term weights $P(R \mid t_i, d_m)$ are computed. These estimates are derived from a learning sample $L \subset \underline{Q} \times \underline{D} \times \mathcal{R}$ of query-document pairs for which we have relevance judgements, so $L = \{(\underline{q}_k, \underline{d}_m, r_{km})\}$. By forming relevance descriptions for the terms common to query and document for every query-document pair in L, we get a multiset of relevance descriptions with relevance judgements $L^x = [(x(t_i, d_m), r_{km}) \mid t_\iota \in q_k^T \cap d_m^T \wedge (\underline{q}_k, \underline{d}_m, r_{km}) \in L]$. This set with multiple occurrences of elements forms the basis for the estimation of the probabilistic index term weights. However, there is a minor problem with the definition of the event space in the probability estimation process: according to the

Table I. List of Symbols

$\underline{Q}$	set of queries submitted to the system
$\underline{D}$	set of documents in the system
Q	set of query representations
D	set of document representations
T	set of terms
$\underline{q}_k$	a specific query
$\underline{d}_m$	a specific document
q_k	representation of query $\underline{q}_k$
d_m	representation of document $\underline{d}_m$
t_i	term
q_k^T	set of terms of the query q_k
d_m^T	set of terms of the document d_m
$\mathcal{R}$	binary relevance scale $\{R, \bar{R}\}$ (relevant/nonrelevant)
r_{km}	relevance judgement of query-document pair $(\underline{q}_k, \underline{d}_m)$
$x(t_i, d_m)$	relevance description of term-document pair (t_i, d_m)
u_{mi}	indexing weight of term t_i for docment d_m
E_x	event space with equiprobable relevance description
E_{BII}	event space with equiprobable query-document pairs
ϱ	retrieval function
e	probabilistic indexing function
$tf \times idf$	SMART indexing function

definition of the BII model, a single event is a query-document pair, so all query-document pairs should be equiprobable. We will denote this event space by E_{BII} in the following. On the other hand, the definition of L^x suggests a different event space E_x in which the triples (query, document, term) are equiprobable events. As different query-document pairs will have different numbers of relevance descriptions, it is obvious that the equiprobability assumption on L implies nonequiprobability on L^x. So there is an error in using E_x instead of E_{BII}. However, the choice of E_x eases the process of probability estimation (see below); we will therefore regard both definitions in the following and investigate whether this difference has any influence on the experimental results.

Following the concepts of other probabilistic IR models, we would estimate the probability $P(R \mid x(t_i, d_m))$ as the relative frequency from those elements of L^x that have the same relevance description (in the case of E_x). (Attributes with continuous values have to be discretized for this purpose, see Wong and Chiu, for example, [34]). As a simple example, assume that the relevance description consists of two elements, defined as

$$x_1 = \begin{cases} 1, & \text{if } t_i \text{ occurs in the title of } d_m \\ 0, & \text{otherwise} \end{cases}$$

$$x_2 = \begin{cases} 1, & \text{if } t_i \text{ occurs once in } d_m \\ 2, & \text{if } t_i \text{ occurs at least twice in } d_m. \end{cases}$$

Table II. Example for Learning Sample

query	doc.	judg.	term	$\vec{x}$
q_1	d_1	R	t_1	$(1,1)$
			t_2	$(0,1)$
			t_3	$(1,2)$
q_1	d_2	$\bar{R}$	t_1	$(0,2)$
			t_3	$(1,1)$
			t_4	$(0,1)$
q_2	d_1	R	t_2	$(0,2)$
			t_5	$(0,2)$
			t_6	$(1,1)$
			t_7	$(1,2)$
q_2	d_3	$\bar{R}$	t_5	$(0,1)$
			t_7	$(0,1)$

Table III. Probability Estimates for the Example from Table II

	$P(R\|\vec{x})$	
$\vec{x}$	E_X	E_{BII}
$(0,1)$	1/4	1/3
$(0,2)$	2/3	1/2
$(1,1)$	2/3	2/3
$(1,2)$	1	1

Table II shows a small learning sample with two queries and two relevance judgements for each query. From this data, the probability estimates depicted in Table III can be derived. Here it can be seen that the different event spaces produce different parameter estimates.

Now, the second important concept of the DIA comes into play: It is the task of an *indexing function* $e(x(t_i, d_m))$ to estimate the probabilities $P(R \mid x(t_i, d_m))$. As indexing functions, different probabilistic classification (or learning) algorithms can be applied. The general advantage of these probabilistic algorithms over simple estimation from relative frequencies is that they yield better estimates, because they use additional (plausible) assumptions about the indexing function.

Within the application of the DIA for indexing with a controlled vocabulary, we have investigated several probabilistic classification algorithms as indexing functions. (Most of these algorithms are restricted to a vector form $\vec{x}$ of the relevance description):

—The so-called Boolean approach developed by Lustig [1] exploits prior knowledge about the relationship between single elements of the relevance description x and the corresponding probability $P(R \mid x)$ for the development of a discrete indexing function.

—The probabilistic learning algorithm ID3 developed by Quinlan [23] seeks for significant components of $\vec{x}$ that form a probabilistic classification tree [10].

—By assuming only pair-wise dependencies among the components of $\vec{x}$, one can apply the tree dependence model [4, 33] as indexing function [31].

—Using logistic regression [11] the indexing function yields $e(\vec{x}) = \frac{exp(\vec{a}^T \cdot \vec{x})}{1 + exp(\vec{a}^T \cdot \vec{x})}$, where $\vec{a}$ is a coefficient vector that is estimated based on the maximum likelihood method [22].

—In this paper, we will use least square polynomials (LSP) [16, 13] as indexing functions. This method is described in more detail in the following.

For the LSP approach, we first have to choose the class of polynomials from which the indexing function is to be selected. Based on the relevance description in vector form $\vec{x}$, a polynomial structure

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

has to be defined (where N denotes the number of dimensions of $\vec{x}$). Then our indexing function yields $e(\vec{x}) = \vec{a}^T \cdot \vec{v}(\vec{x})$, where $\vec{a}$ is the coefficient vector to be estimated.

Let $y(q_k, \underline{d}_m) = y_{km}$ denote a class variable for each element of L with $y_{km} = 1$ if $r_{km} = R$ and $y_{km} = 0$ otherwise. Then the coefficient vector $\vec{a}$ is estimated such that it minimizes the squared error $E((y - \vec{a}^T \cdot \vec{v}(\vec{x}))^2)$. Here $E(.)$ denotes the expectation based on a uniform distribution within E_x or E_{BII}, respectively. The coefficient vector $\vec{a}$ can be computed by solving the linear equation system [14].

$$E(\vec{v} \cdot \vec{v}^T) \cdot \vec{a} = E(\vec{v} \cdot y). \tag{4}$$

As an approximation for the expectations, the corresponding arithmetic means from the learning sample are taken. The momental matrix M which contains both sides of the equation system (4) is computed according to the underlying event space:

— In the case of E_{BII}, we have

$$M_{BII} = \frac{1}{|L|} \sum_{(\underline{q}_k, \underline{d}_m, r_{km}) \in L} \frac{1}{|q_k^T \cap d_m^T|} \sum_{t_i \in q_k^T \cap d_m^T} (\vec{v}_{im} \cdot \vec{v}_{im}^T, \vec{v}_{im} \cdot y_{km})$$

where $\vec{v}_{im} = \vec{v}(\vec{x}(t_i, d_m))$.

— For the event space E_x, the matrix M_x is computed as

$$M_x = \frac{1}{|L^x|} \sum_{(x_{im}, r_{km}) \in L^x} (\vec{v}_{im} \cdot \vec{v}_{im}^T, \vec{v}_{im} \cdot y_{km}).$$

The momental matrix M can then be solved to yield the coefficient vector $\vec{a}$.

For most of the experiments described here, we used a relevance description of four elements and a polynomial structure $\vec{v}(\vec{x})$ of length five (i.e., an additional constant for a linear function). So we had to compute five coefficients $a_1, \ldots, a_5$. Each of these parameters is estimated for a collection rather than a particular query term (as in conventional probabilistic retrieval), and is therefore based on much more evidence. In our experiments, the smallest learning sample L has about 400 elements. In comparison, in conventional probabilistic retrieval, a typical feedback query might be 20 terms long, and thus you must estimate 40 probabilistic parameters, each one based on perhaps 15 elements. On the other hand, our approach considers interdependencies between all the parameters, and other experiments [16, 12] have shown that we need about 50-100 elements per parameter in order to achieve reliable estimates.

5. TEST SETTING

Some experiments with a preliminary version of our approach in combination with controlled vocabulary indexing have been described by Fuhr [12, pp. 146–150]. In this paper, we apply our approach to the task of free term indexing and compare it with the standard SMART indexing procedures as described by Salton and Buckley [27]. In most of our experiments, we use the same representation of queries and documents in the SMART approach. For this reason, our evaluation should be regarded as a starting point for further experiments in which improved representations of documents (e.g., with noun phrases as index terms) are considered.

For our experiments, we used the five experimental collections shown in Table IV. In order to perform predictive experiments, the set of queries of each collection was split into halves. Because of the limited number of queries in our collections, a random sampling technique might have split the queries into two very different samples; therefore we used the number of relevant documents for a query as a criterion to get two disjoint, but similar query sets for each collection. Table IV shows for both sets the number of queries and the average number of terms as well as the average number of relevant documents per query. From these two query sets, we used one for the estimation of the probabilistic indexing function, which is called learning sample in the following. With the second set, called test sample below, only predictive retrieval runs were performed; that is, no relevance information from this set has been used for the estimation of the indexing function. In additional retrospective experiments the learning sample was used for retrieval runs, too.

Besides the choice of the query set, we also had to decide which documents should be considered in the learning set L. In our experiments, we investigated two possibilities:

(1) Full relevance information: All documents retrieved for the queries from the learning sample are considered. A document d_m is retrieved with respect to a query q_k if $d_m^T \cap q_k^T \neq \emptyset$.

Table IV. Collections Used for Experiments

collection	CACM	CISI	CRAN	INSPEC	NPL
#documents	3204	1460	1398	12684	11429
#learning queries	26	38	113	39	47
#test queries	26	38	112	38	46
avg. length learning	11.1	25.7	9.1	15.8	7.2
avg. length test	10.5	19.9	9.2	15.8	7.1
avg. rels. learning	14.8	39.8	8.3	33.2	22.8
avg. rels. test	15.8	42.1	8.1	32.8	22.0

(2) Top 15 documents: Only the top 15 documents for each query (by applying the retrieval function $\varrho_{tf\imath df}$ with $tf \times idf$ indexing weights, see below) are included in L.

The first variant follows from the BII model which is based on the event space $|\underline{Q}| \times |\underline{D}|$; the additional assumptions restrict this event space to a set of all query-document pairs which have at least one term in common. The second case is more realistic for applications, because mostly a user will only judge the top ranking documents.

For the development of the LSP indexing functions, we first had to define a relevance description $\vec{x}$, for which we used the following parameters.

$tf_{m\imath}$: within-document frequency (wdf) of $t_\imath$ in d_m.

$\max tf_m$: maximum wdf $\mathrm{tf}_{m\imath}$ of all terms $t_\imath \in d_m^T$.

$n_\imath$: number of documents in which $t_\imath$ occurs.

$|\underline{D}|$: number of documents in the collection.

$|d_m^T|$: number of different terms in d_m.

$ta_{m\imath}$: $= 1$, if $t_\imath$ occurs in the title of d_m, and 0 otherwise.

With the exception of the parameter $ta_{m\imath}$, we consider only information that is also used in the standard SMART indexing procedures [27]. With these parameters, the components of the relevance description were defined as

$$x_1 = tf_{m\imath}$$

$$x_2 = 1/\max tf_m$$

$$x_3 = \log(n_\imath / |\underline{D}|)$$

$$x_4 = \log |d_m^T|$$

$$x_5 = ta_{m\imath}.$$

Based on this relevance description, four different indexing functions e_L, e_Q, e_{tfidf}, and e_{ta} were developed by defining the polynomial structures

$$\vec{v}_L = (1, x_1, x_2, x_3, x_4)$$

$$\vec{v}_Q = (1, x_1, x_2, x_3, x_4, x_1^2, x_1x_2, x_1x_3, x_1x_4, x_2^2, x_2x_3, x_2x_4, x_3^2, x_3x_4, x_4^2)$$

$$\vec{v}_{tfidf} = (1, x_1x_2x_3, x_1x_2, x_3, x_4)$$

$$\vec{v}_{ta} = (1, x_1, x_2, x_3, x_4, x_5).$$

So we have the indexing functions

$$e_L = a_0 + a_1 tf_{mi} + a_2/\max tf_m + a_3 \log(n_i/|\underline{D}|) + a_4 \log|d_m^T|,$$

$$\begin{aligned} e_Q = {} & a_0 + a_1 tf_{mi} + a_2/\max tf_m + a_3 \log(n_i/|\underline{D}|) + a_4 \log|d_m^T| \\ & + a_5(tf_{mi})^2 + a_6 tf_{mi}/\max tf_m + a_7 tf_{mi} \cdot \log(n_i/|\underline{D}|) \\ & + a_8 tf_{mi} \cdot \log(|d_m^T|) + a_9/(\max tf_m)^2 \\ & + a_{10}/\max tf_m \cdot \log(n_i/|\underline{D}|) + a_{11}/\max tf_m \cdot \log(|d_m^T|) \\ & + a_{12}(\log(n_i/|\underline{D}|))^2 + a_{13} \log(n_i/|\underline{D}|) \log|d_m^T| + a_{14}(\log|d_m^T|)^2, \end{aligned}$$

$$\begin{aligned} e_{tfidf} = {} & a_0 + a_1 tf_{mi} \log(n_i/|\underline{D}|)/\max tf_m + a_2 tf_{mi}/\max tf_m \\ & + a_3 \log(n_i/|\underline{D}|) + a_4 \log|d_m^T|, \end{aligned}$$

$$e_{ta} = a_0 + a_1 tf_{mi} + a_2/\max tf_m + a_3 \log(n_i/|\underline{D}|) + a_4 \log|d_m^T| + a_5 ta_{mi}.$$

e_L and e_{ta} are linear functions of $\vec{x}$, while e_Q is a so-called "complete quadratic polynomial" of $\vec{x}$. e_{tfidf} was defined in order to get a function similar to the best SMART indexing function, called $tf \times idf$ [27].

In our experiments, it turned out that the indexing functions may yield negative indexing weights for some relevance descriptions. In these cases, the weight was set to 0. The experiments described by Pfeifer [22] dealing with the problem of negative estimates indicated that this is a weakness of LSP indexing functions: Even if $y_{km} = 0$, a negative indexing weight is regarded as an error (and similarly for $y_{km} = 1$ and a weight > 1). It turned out that slight improvements in terms of indexing quality can be achieved when relevance descriptions which would get negative indexing weights are removed from the learning sample, and then the coefficient vector is recomputed. This procedure can be repeated several times, with smaller samples and diminishing improvements from step to step.

The retrieval results for the LSP indexing functions are compared with those of the $tf \times idf$ indexing function described in the following (for further details, see Salton and Backley [27]). In contrast to our indexing method, the SMART approach does not consider any relevance information for the computation of the indexing weights. With the parameters as defined above, first a

preliminary indexing weight α_{mt} for each term in a document is computed:

$$\alpha_{mt} = \left(0.5 + 0.5\frac{tf_{mt}}{\max tf_m}\right) \cdot \log\frac{n_t}{|\underline{D}|}.$$

These weights are further normalized by the factor $w_m = \sqrt{\sum_{t_t \in d_m^T} \alpha_{mt}^2}$ So the final indexing weight for a term t_t in a document d_m according to the $tf \times idf$ formula yields $u_{mt} = \alpha_{mt} / w_m$.

In the retrieval process, the indexing weights u_{mt} are used by the retrieval function $\varrho(q_k, d_m)$ which computes a relevance status value for each query-document pair. Then the documents are ranked by decreasing relevance values.

We performed a few experiments with the BII formula (3). As experiments described by Fuhr [13] have shown, the assumption of a constant β for the probabilities $P(R \mid d_m)$ yields significantly better retrieval results than document-specific estimates. So the retrieval function based on the BII model is

$$\varrho_{BII}(q_k^D, d_m^D) = \sum_{t_t \in q_k^T \cap d_m^T} \log\frac{u_{mt}}{\beta}.$$

In most of our experiments, we considered the scalar product as retrieval function with

$$\varrho(q_k^D, d_m^D) = \sum_{t_t \in q_k^T \cap d_m^T} c_{kt} \cdot u_{mt}.$$

Here c_{kt} denotes the weight of the term t_t with respect to the query q_k. As mentioned by Wong and Yao [35], this retrieval function can be given a utility theoretic interpretation in the case of probabilistic indexing weights u_{mt}: The weight c_{kt} can be regarded as the utility of the term t_t, and the retrieval function gives the expected utility of the document with respect to the query.

For the computation of the query term weights c_{kt}, three different possibilities were considered in our experiments. In the following, we denote these weighting schemes as subscript of the retrieval function:

— ϱ_{bin}: Binary query term weights are used with $c_{kt} = 1$ for all $t_t \in q_k^T$.
— ϱ_{tf}: The query terms weight c_{kt} is set equal to the number of occurrences tf_k, of t_t in the query formulation q_k.
— ϱ_{tfidf}: The query term weights are computed in the same way as the $tf \times idf$ document term weight, except that the within-query frequencies tf_{kt} (and max tf_k) are regarded instead of the within-document frequencies.

For evaluation, the standard SMART routines were taken for computing the average precision of a set of queries at a certain recall point. From these precision values at the recall points 0.25, 0.50, and 0.75, we took the average as global measure of retrieval quality. In addition, significance test were performed with the Wilcoxon signed-ranks test. For this purpose, the

Table V. Average Precision Values for Learning and Test Samples

collection	learn. sample	test sample	relative difference
CACM	0.3046	0.2963	- 2.7%
CISI	0.1358	0.2099	+ 54.6%
CRAN	0.3634	0.3816	+ 5.0%
INSPEC	0.2214	0.2489	+ 12.4%
NPL	0.1505	0.2138	+ 42.1%

precision values at a certain recall point were compared query-wise for two different combinations of indexing and retrieval function (like the above, the recall points 0.25, 0.50, and 0.75 were regarded here). However, due to the relatively small number of queries in most of our test samples, only a few significant differences were found (see next section).

6.EXPERIMENTAL RESULTS

With the test parameters described before, we performed a number of retrieval runs according to a factorial test plan; that is, we tested (almost) all possible parameter combinations. In the following, we will present the experimental results grouped by the different parameters, in order to show the influence of each parameter on the final retrieval quality. Unless mentioned otherwise, all probabilistic indexing functions are based on the event space E_x.

Learning vs. Test Sample

Before presenting results of predictive retrieval runs for probabilistic indexing, we want to discuss the sampling problem: our approach requires a representative sample of the collection as a learning sample. With the limited number of queries available in our collections, we had to split the query sets into similar halves instead. Now we want to investigate how similar these two samples really are. It is obvious that this is still an open research problem in IR: having experimental results for a collection A, for which other collections is A representative (so that one can conclude that the experimental results hold for this set of collections)?

As a very simple measure of the similarity of two collections, we use the results of the retrieval function ϱ_{tfidf} in combination with $tf \times idf$ indexing weights here. Table V shows the average precision values for the learning and the test samples of each collection, and the relative difference between the two results. It can be seen that we have the best sampling for the CACM collection, and for the CRAN and INSPEC collection, the two query sets also seem to be quite similar. In the case of the CISI collection, the difference is much larger (see also the average query lengths in Table IV); in the following, we will see that this may account for some strange results that we got for the CISI collection. We have the biggest difference for the NPL collection; however, as claimed by Salton and Buckley [27], the combination of $\varrho tfidf$

Table VI. Retrieval Results Using Either the Top 15 Ranked Documents or Full Relevance Information (learning sample, ϱ_{bin}, E_x)

collection	e_L full	e_L top	e_Q full	e_Q top	$e_{tf \cdot idf}$ full	$e_{tf \cdot idf}$ top
CACM	0.2954	0.3066	0.3249	0.3210	0.3021	0.3228
		+ 3.8%		- 1.2%		+ 6.9%
CISI	0.1021	0.1160	0.1108	0.1236	0.1033	0.1193
		+ 13.6%		+ 11.6%		+ 15.5%
CRAN	0.3710	0.3776	0.3532	0.3389	0.3504	0.3405
		+ 1.8%		- 4.0%		- 2.8%
INSPEC	0 1982	0.2094	0.2228	0.1875	0.2096	0.2139
		+ 5 7%		- 15.8%		+ 2.1%
NPL	0.2110	0 1729	0.1750	0.1295	0.1975	0.1243
		- 18.1%		- 26.0%		- 37.1%

Table VII. Retrieval Results Using Either the Top 15 Ranked Documents or Full Relevance Information (test sample, ϱ_{bin}, E_x)

collection	e_L full	e_L top	e_Q full	e_Q top	$e_{tf \cdot idf}$ full	$e_{tf \cdot idf}$ top
CACM	0 3103	0.3046	0.3688	0.3566	0.3347	0.3324
		- 1.8%		- 3.3%		- 0.7%
CISI	0 1405	0.1731	0.1571	0.1949	0.1457	0.1745
		+ 23.2%		+ 24.1%		+ 19.8%
CRAN	0.4122	0.4265	0.4065	0.3925	0.3914	0.3764
		+ 3.5%		- 3.4%		- 3.8%
INSPEC	0 2331	0 2307	0.2452	0.2141	0.2036	0.1929
		- 1.0%		- 12.7%		- 5.3%
NPL	0.2393	0.2834	0.2068	0.1749	0.2710	0.1943
		+ 18.4%		- 15.4%		- 28.3%

and $tf \times idf$ is not appropriate for the NPL collection, since terms occur at most once in the queries of this collection. Therefore, our measure of similarity may be invalid for the NPL collection. This assumption is also supported by the results presented in the following section.

Documents in the Learning Set

Using either the top 15 ranked documents or all documents retrieved as elements of L, we show the retrieval results for the different indexing functions in Tables VI and VII. It can be seen that the differences in the retrieval results caused by the choice of L are the smallest for the indexing function e_L; this may be due to the fact that the estimation of the coefficient vector $\vec{a}$ is less crucial for e_L than for e_Q and $e_{tf \cdot idf}$, since e_L is the only linear function presented in these tables. With the exception of the CISI collection, most of the results for the indexing functions based on the top 15 documents are worse than those based on full relevance information. On the other hand,

Table VIII. Retrieval Results Using Either E_x or E_{BII} (learning sample, top, ϱ_{bin})

collection	e_L		e_Q		e_{tfidf}	
	E_x	E_{BII}	E_x	E_{BII}	E_x	E_{BII}
CACM	0.3066	0.3110	0.3210	0.3036	0.3228	0.3122
	+ 1.4%		- 5.4%		- 3.3%	
CISI	0.1160	0.1166	0.1236	0.1227	0.1193	0.1172
	+ 0.5%		- 0.7%		- 1.8%	
CRAN	0.3776	0.3766	0.3389	0.3302	0.3405	0.3271
	- 0.3%		- 2.6%		- 3.9%	
INSPEC	0.2094	0.2160	0.1875	0.1849	0.2139	0.2077
	+ 3.2%		- 1.4%		- 2.9%	
NPL	0.1729	0.1706	0.1295	0.1224	0.1243	0.1197
	- 1.3%		- 5.5%		- 3.7%	

Table IX. Retrieval Results Using Either E_x or E_{BII} (test sample, top, ϱ_{bin})

collection	e_L		e_Q		e_{tfidf}	
	E_x	E_{BII}	E_x	E_{BII}	E_x	E_{BII}
CACM	0 3046	0.3049	0.3566	0.3263	0.3324	0.3277
	+ 0.1%		- 8.5%		- 1.4%	
CISI	0.1731	0.1693	0.1949	0.1886	0.1745	0.1636
	- 2.2%		- 3.2%		- 6.2%	
CRAN	0.4265	0.4178	0.3925	0.3719	0.3764	0.3598
	- 2.0%		- 5.2%		- 4.4%	
INSPEC	0.2307	0.2398	0.2141	0.2101	0.1929	0.1810
	+ 3.9%		- 1.9%		- 6.2%	
NPL	0.2834	0.2815	0.1749	0.1664	0.1943	0.1846
	- 0.7%		- 4.9%		- 5.0%	

the loss in retrieval quality by restricting to the top 15 documents is not too large to make our approach infeasible for practical applications. Following this point of view, we will discuss only results of indexing functions based on the top 15 ranked documents in the following section.

Event Space

Tables VIII and IX show the difference in the retrieval quality by using either the event space E_x or E_{BII}. For e_L, the differences are negligible, while the other indexing functions are again more sensitive to small changes in the learning samples. In general, one can say that the choice of the event space is not crucial for the development of probabilistic indexing functions.

BII Model

Only a few experiments were performed with the retrieval function ϱ_{BII}, since it turned out that better retrieval results can be achieved with the other retrieval functions. In order to derive an estimate for the parameter β

Table X. Retrieval Results Using e_L or e_Q with the BII Retrieval Function (ϱ_{BII}, E_{BII})

	learning sample				test sample			
	full		top		full		top	
collection	e_L	e_Q	e_L	e_Q	e_L	e_Q	e_L	e_Q
CACM	0.2373	0.3169	0.2552	0.2894	0.2606	0.3352	0.2680	0.3180
	+ 33.5%		+ 13.4%		+ 28.6%		+ 18.7%	
CISI	0 1020	0.1043	0.1146	0.1138	0.1347	0 1390	0.1402	0.1446
	+ 2.3%		- 0.7%		+ 3.2%		+ 3.1%	
CRAN	0.3427	0.3587	0.3662	0.3486	0.3902	0.4096	0.4010	0.4029
	+ 4.7%		- 4.8%		+ 5.0%		+ 0.5%	
INSPEC	0.1597	0.1870	0.1677	0.1796	0.1740	0.2078	0.1656	0.2001
	+ 17.1%		+ 7.1%		+ 19.4%		+ 20.8%	
NPL	0.2333	0.2355	0.2371	0.2096	0.2510	0.2561	0.2831	0.2557
	+ 0.9%		- 11.6%		+ 2.0%		- 9.7%	

for each collection, we performed series of retrieval runs on the learning samples with different values for this parameter. Then we choose the value that gave us the best retrieval results. Different values of β were derived this way for e_L and e_Q as well as for full relevance information and feedback from the top 15 ranked documents. The retrieval results for the BII model are shown in Table X. For the learning sample and full relevance feedback, e_Q gives better retrieval results than e_L. Since e_Q contains all the parameters of e_L plus all quadratic combinations of elements of $\vec{x}$, it is theoretically supposed to be better than e_L, at least in combination with ϱ_{BII}. With feedback information from the top 15 documents only, we get mixed results for the learning sample. By comparing the results of the BII model with those of ϱ_{bin} from the previous tables, it can be seen that ϱ_{BII} performs clearly worse than ϱ_{bin}. Some further experiments showed that in all cases, better retrieval results can be achieved with the ϱ_{BII} function when indexing weights $u_{mi} \leq \beta$ are ignored by the retrieval function: in contrast to the other retrieval functions where every indexing weight $u_{mi} > 0$ for a term $t_i \in q_k^T \cap d_m^T$ increases the relevance value, with ϱ_{BII} weights with $0 < u_{mi} < \beta$ decrease the relevance value. With this modification of ϱ_{BII}, we got results similar to those of ϱ_{bin}. However, we do not have a theoretical justification for this modification.

Indexing Functions

In Tables XI and XII we compare the retrieval results of the probabilistic indexing functions with those of the $tf \times idf$ formula. At first glance, these results seem to be inconsistent. With the learning samples, there is a different probabilistic indexing function for each collection which yields the best retrieval results. We would expect that e_Q always performs better than e_L here, but in contrast to the experiments with the BII model, e_Q yields worse results than e_L for three of the five collections. This shows that the results depend on the choice of the retrieval function (see also the discussions

Table XI. Probabilistic Indexing Functions vs. $tf \times idf$ Formula (learning sample, top, E_x, ϱ_{bin})

collection	$tf \times idf$	e_L	e_Q	e_{tfidf}
CACM	0.2604	0.3066 + 17.7%	0.3210 + 23.3%	0.3228 + 24.0%
CISI	0.1188	0.1160 - 2.4%	0.1236 + 4.0%	0.1193 + 0.4%
CRAN	0.3567	0.3776 + 5.9%	0.3389 - 5.0%	0.3405 - 4.5%
INSPEC	0.1706	0.2094 + 22.7%	0.1875 + 9.9%	0.2139 + 25.4%
NPL	0.1580	0.1729 + 9.4%	0.1295 - 18.0%	0.1243 - 21.3%

Table XII. Probabilistic Indexing Functions vs. $tf \times idf$ Formula (test sample, top, E_x, ϱ_{bin})

collection	$tf \times idf$	e_L	e_Q	e_{tfidf}
CACM	0.2674	0.3046 + 13.9%	0.3566 + 33.4%	0.3324 + 24.3%
CISI	0.1407	0.1731 + 23.0%	0.1949 + 38.5%	0.1745 + 24.0%
CRAN	0.3841	0.4265 + 11.0%	0.3925 + 2.2%	0.3764 - 2.0%
INSPEC	0.1848	0.2307 + 24.8%	0.2141 + 15.9%	0.1929 + 4.4%
NPL	0.2141	0.2834 + 32.4%	0.1749 - 18.3%	0.1943 - 9.2%

below). However, because of its poor performance ϱ_{BII} was not considered in further experiments.

Looking at the test samples, we get more uniform results. For three of the five collections e_L yields the best retrieval results of all indexing functions considered. The CISI and CACM collections behave differently, and for both collections the similarity between learning and test sample may be the reason. With the CISI collection, the results for the probabilistic indexing functions in comparison to the $tf \times idf$ formula are better for the test sample than for for the learning sample. In the case of the CACM collection, the better performance of e_Q (in comparison to e_L) can be explained by the small difference between learning and test sample. Here we get good estimates for the large number of parameters of e_Q. With the other collections, the (relatively) small learning samples yield only good estimates for the indexing function with the lowest number of parameters and a linear structure, namely e_L. In the case of e_{tfidf}, we have the same number of parameters as for e_L, but the elements of the polynomial structure $\vec{v}_{tfidf}$ are strongly dependent on each other, which makes this function rather sensitive to

Table XIII Comparison of Different Retrieval Functions (test sample, top, E_x)

collection	$tf \times idf$	e_L			e_{ta}	
	$\varrho_{tf \cdot idf}$	ϱ_{bin}	ϱ_{tf}	$\varrho_{tf \cdot idf}$	ϱ_{bin}	ϱ_{tf}
CACM	0.2963	0.3046 + 2.8%	0.3371 + 13.8%	0.3283 + 10.8%	0.3148 + 6.2%	0.3464 + 16.9%
CISI	0.2099	0 1731 - 17.5%	0.2288 + 9.0%	0.2052 - 2.2%	0.1751 - 16.6%	0.2311 + 10.1%
CRAN	0.3816	0.4265 + 11.8%	0.4293 + 12.5%	0.3922 + 2.8%	0.4554 + 19.3%	0.4556 + 19.4%
INSPEC	0.2489	0.2307 - 7.3%	0.2708 + 8.8%	0.2502 + 0.5%	0.2326 - 6.5%	0.2694 + 8.2%
NPL	0.2138	0.2834 + 32.6%	0.2834 + 32.6%	0.2382 + 11.4%	- -	- -

differences between learning and test sample. So, with the size of the collections available, only e_L seems to be appropriate.

Comparing the results of the probabilistic indexing functions with those of the $tf \times idf$ function, one can see that the probabilistic functions outperform the SMART function in most cases.

Retrieval Functions

If one is interested in good retrieval results, the comparison of indexing functions by using a simple retrieval function like ϱ_{bin} may not be appropriate. Table XIII shows the results for the indexing function e_L in combination with the three retrieval functions ϱ_{bin}, ϱ_{tf} and $\varrho_{tf \cdot idf}$. It can be seen that ϱ_{tf} yields the best results among the retrieval functions. As ϱ_{tf} performs better than ϱ_{bin}, the information about the within-query frequency of the search terms seems to be useful in consideration with probabilistic document indexing. This result confirms the utility-theoretic justification of linear retrieval functions. On the other hand, there is no improvement by using $\varrho_{tf \cdot idf}$ instead of ϱ_{tf} for the probabilistic indexing weights. This is plausible, since the information about the inverse document frequency of the terms has been considered already in the document indexing process. The significance tests for the comparison of ϱ_{bin} and ϱ_{tf} showed significant differences ($P > 95$ percent) for the CISI collection at all three recall levels and for the CACM and the INSPEC collection at the 0.75 recall level.

The retrieval results for the probabilistic indexing functions are compared with those of the $tf \times idf$ indexing weights and the $\varrho_{tf \cdot idf}$ retrieval function. Salton and Buckley [27] prove this combination to be, more or less, the best SMART indexing and retrieval method. The comparison of this method with e_L in combination with ϱ_{tf} shows that the probabilistic indexing function yields better retrieval results for all collections. (The Wilcoxon test indicated significant differences ($P > 99.5$ percent) for the CRAN and the NPL collection at the recall levels of 0.25 and 0.5.) This finding is not surprising. The

SMART approach offers a general indexing function which is applicable to a broad range of collections, whereas our approach can be adapted to each specific collection. On the other hand, the development of probabilistic indexing function requires learning data which has to be collected from the running retrieval system, but the SMART indexing functions can be applied without having any relevance information at all. For this reason, with regard to applications, the two approaches are complementing each other. When a new collection is set up, first the SMART approach should be applied and relevance information should be collected. After a while, when there is enough learning data available, the probabilistic approach can be applied. As more and more relevance information is collected, the probabilistic indexing can be further improved by choosing more detailed relevance descriptions and more complex indexing functions (polynomial structures).

Document Representation

In order to test the effect of an improved document representation, we performed a few experiments with the indexing function e_{ta}. With this function, we consider whether an index term occurs in the title of a document or only in the abstract. The experimental results of e_{ta} for the test samples are shown in Table XIII (No results are given for the NPL collection here, since the NPL document texts were not available for our experiments.) In comparison to e_L, we get significant improvements for the CRAN collection ($P > 99$ percent at all three recall levels). For the other collections, only minor improvements were achieved (and a slight degradation with ϱ_{tf} for the INSPEC collection). The Wilcoxon test showed significant differences only for the CACM collection at 0.25 recall for ϱ_{tf} and at 0.5 recall for ϱ_{bin} ($P > 99$ percent). The best probabilistic combination (e_{ta}, ϱ_{tf}) is significantly better than the SMART approach for the CRAN collection at all three recall levels ($P > 99$ percent), for the INSPEC collection at the recall points of 0.25 and 0.75 ($P > 95$ percent) and for the CACM collection at 0.75 recall ($P > 95$ percent).

On the learning sample (results not given here), we observed the same behavior of e_L for the different collections. So one can conclude that the distinction between title and abstract should be considered when developing indexing functions, but the effect is collection-dependent. (In the case of the CRAN collection, the titles are one-sentence descriptions of the experiments described in the articles, so the terms occurring in the title are most significant with relation to the document.) Again, these results show a major advantage of our indexing functions: they are able to adapt to specific characteristics of a collection. On the other hand, if a new feature included in the relevance description is insignificant, there is no loss in retrieval performance (provided that the learning sample is not too small).

7. CONCLUSIONS

In this paper, we have devised a new probabilistic indexing approach which is feasible for real applications. The major concepts of our approach are the

following:

—Definition of a probabilistic indexing model in terms of the BII model. In contrast to nonprobabilistic indexing models (see Salton and Buckley [27]) or earlier probabilistic models [6], the indexing weights of the BII model have a clear notion as probabilities in a well-defined event space.

—Abstraction from specific term-document pairs by definition of relevance descriptions. Unlike many other probabilistic IR models, the probabilistic parameters do not relate to a specific document or query. This feature overcomes the restriction of limited relevance information that is inherent to other models, e.g., by regarding only relevance judgements with respect to the current request. Our approach can be regarded as a long-term learning method which complements the short-term learning method of relevance weighting of search terms. For the latter problem, the retrieval-with -probabilistic-indexing (RPI) model [13] has been developed. This model allows to distinguish between two queries q_1, q_2 with $q_1^T = q_2^T$ by regarding query-specific relevance feedback information (similar to model 3 presented by Robertson et al. [24]). Consequently, the query representation of the RPI model is a pair $q_k = (q_k^T, q_k^J)$, where q_k^J denotes a set of documents with relevance judgements with relation to q_k.

—Flexibility of the form of representation of term-document relationships in relevance descriptions. While other probabilistic models relate to specific forms of representation (which is also a reason for the large number of models published), our approach can be easily adapted to new forms of representation. This is very important for new text analysis and knowledge-based methods, which have not been considered by probabilistic models yet. Now we have devised an easy-to-apply model for the integration of these methods in IR systems.

—Probabilistic learning (or classification) methods as indexing functions instead of simple parameter estimation methods. This way, we can make better use of the available learning data, and we can choose the complexity of the indexing function according to the size of the learning sample.

The experimental results indicate that our approach can be applied in running IR systems and that it is superior to other indexing methods. Currently, the size of the available test collections puts some difficulties on the testing of the probabilistic indexing approach, as the results for the nonlinear indexing functions show. In contrast to other probabilistic models, this problem can be neglected in real applications, as the learning sample size is a function of the total number of queries with relevance judgements available. Furthermore, we have shown that the restriction of the learning sample to the top ranking documents is not a serious impediment for the applicability of our method.

With the concepts described in this paper, we have given a framework for the development of probabilistic indexing functions. Besides the investigation of different probabilistic learning and classification methods for the

development of indexing functions, the consideration of improved document representations will be a prospective field of research.

ACKNOWLEDGMENT

We thank Keith van Rijsbergen for his constructive comments on an earlier version of this paper.

REFERENCES

1. BEINKE-GEISER, U., LUSTIG, G., AND PUTZE-MEIER, G Indexieren mit dem System DAISY. In *Automatische Indexierung zwischen Forschung und Anwendung*, G. Lustig, Ed. Olms, Hildesheim, Germany, 1986, pp. 73-97.
2. 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.
3. BIEBRICHER, P., FUHR, N., KNORZ, G., LUSTIG, G., AND SCHWANTNER, M. Entwicklung und Anwendung des automatischen Indexierungssystems AIR/PHYS. *Nachrichten fuer Dokumentation 39* (1988), 135-143.
4. CHOW, C. K., AND LIU, C N. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theor. 14*, 3, (1968), 462-467.
5. CROFT, W. B. Boolean queries and term dependencies in probabilistic retrieval models. *J. Am. Soc. Inf. Sci. 37*, 2 (1986), 71-77.
6. CROFT, W. B. Document representation in probabilistic models of information retrieval. *J. Am. Soc. Inf. Sci. 32*, (1981), 451-457.
7. CROFT, W. B. Experiments with representation in a document retrieval system *Inf. Tech. Res. Dev. 2*, (1983), 1-22.
8. FAGAN, J Automatic phrase indexing for document retrieval. In *Proceedings of the Tenth Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, C. T. Yu and C. J. van Rijsbergen, Eds. ACM, New York, 1987, pp. 91-101.
9. FAGAN, J. L. The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *JASIS. 40*, 2 (1989), 115-132
10. FAISST, S. Development of indexing functions based on probabilistic decision trees (in german). Diploma thesis, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II, Darmstadt, Germany, 1990.
11. FREEMAN, D. H. *Applied Categorial Data Analysis.* Dekker, New York, 1987
12 FUHR, N. Probabilistisches Indexing und Retrieval. Dissertation, TH Darmstadt, Fachbereich Informatik, 1988. Available from Fachinformationszentrum Karlsruhe, Eggenstein-Leopoldshafen, Germany.
13. FUHR, N. Models for retrieval with probabilistic indexing. *Inf. Process. Manage. 25*, 1 (1989), 55-72.
14. FUHR, N. Optimum polynomial retrieval functions based on the probability ranking principle. *ACM Trans. Inf. Syst. 7*, 3 (1989), 183-204.
15 GORDON, M. Probabilistic and genetic algorithms for document retrieval. *Commun. ACM. 31*, 10 (1988), 1208-1218.
16. KNORZ, G. *Automatisches Indexieren als Erkennen Abstrakter Objekte.* Niemeyer, Tübingen, Germany, 1983.
17. KWOK, K. L. An interpretation of index term weighting schemes based on document components. In *Proceedings of the 1986 ACM Conference on Research and Development in Information Retrieval*, F. Rabitti, Ed. ACM, New York, 1986, pp. 275-283.
18. KWOK, K. L. A neural network for probabilistic information retrieval. In *Proceedings of the Twelfth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, N. Belkin and C. T. van Rijsbergen, Eds. ACM, New York, 1989, pp. 21-30.

19. KWOK, K. L. AND KUAN, W. Experiments with document components for indexing and retrieval *Inf. Process. Manage. 24*, 4 (1988), 405-417.
20. MARON, M. E. Probabilistic approaches to the document retrieval problem. In *Research and Development in Information Retrieval*, G. Salton and H. J. Schneider, Eds. Springer, Berlin et al., 1983, pp. 98-107.
21 MARON, M. E., AND KUHNS, J L. On relevance, probabilistic indexing, and information retrieval. *J. ACM. 7*, (1960), 216-244.
22. PFEIFER, U. R. Development of log-linear and linear-iterative indexing functions (in german). Diploma thesis, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II, Darmstadt, Germany, 1990.
23. QUINLAN, J. R. The effect of noise on concept learning. In *Machine Learning: An Artificial Intelligence Approach, Vol. II*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds Morgan Kaufmann, Los Altos, California, 1986, pp. 149-166.
24 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. Dev. 1*, (1982), 1-21.
25. ROBERTSON, S. E., AND SPARCK JONES, K Relevance weighting of search terms. *J. Am. Soc. Inf. Sci. 27*, (1976), 129-146.
26. ROBERTSON, S. E., VAN RIJSBERGEN, C. J., AND PORTER, M F Probabilistic models of indexing and searching. In *Information Retrieval Research*, R N. Oddy, S. E Robertson, C J. Van Rijsbergen, and P. W. Williams, Eds. Butterworths, London, England, 1981, 35-56.
27. SALTON, G. AND BUCKLEY, C Term weighting approaches in automatic text retrieval *Inf. Process. Manage. 24*, 5, (1988), 513-523.
28. SALTON, G , YANG, C S. AND YU, C. T. A theory of term importance in automatic text analysis. *J. Am. Soc. Inf. Sci. 36*, (1975), 33-44.
29. SEMBOK, T M. T., AND VAN RIJSBERGEN, C J SILOL: A simple logical-linguistic document retrieval system. *Inf. Process. Manage. 26*, 1, (1990), 111-134.
30. SMEATON, A. F. Incorporating syntactic information into a document retrieval strategy: an investigation. In *Proceedings of the 9th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York, 1986, pp. 103-113
31. TIETZE, A. Approximation of discrete probability distributions by dependence trees and their application as indexing functions (in german). Diploma thesis, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II, Darmstadt, Germany, 1989
32. TURTLE, H. R. Inference networks for document retrieval. PHD thesis, Computer and Information Science Dept., Univ. of Massachusetts, Boston, 1990.
33. VAN RIJSBERGEN, C J A theoretical basis for the use of co-occurrence data in information retrieval. *J. Doc. 33*, (1977), 106-119.
34. WONG, A. K. C., AND CHIU, D. K Y. Synthesizing statistical knowledge from incomplete mixed-mode data. *IEEE Trans. Pattern Anal. Mach. Intell. 9*, 6, (1987), 796-805.
35. WONG, S K. M., AND YAO, Y Y. A probability distribution model for information retrieval. *Inf. Process. Manage. 25*, 1, (1989), 39-53.
36. YU, C. T., AND MIZUNO, H Two learning schemes in information retrieval In *Proceedings of the 11th International Conference on Research and Development in Information Retrieval*, Y Chiaramella, Ed. Presses Universitaires de Grenoble, Grenoble, France, 1988, pp. 201-218.
37. YU, C. T , AND SALTON, G. Precision weighting. An effective automatic indexing method *J. ACM 23* (1976), 76-88.