Probabilistic Document Indexing from Relevance Feedback Data

Norbert Fuhr
TH Darmstadt
Darmstadt
West Germany

Chris Buckley *
Cornell University
Ithaca, NY
USA

Abstract

Based on the binary independence indexing model, we apply three new concepts for probabilistic document indexing from relevance feedback data:

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 more complex document structures or different types of terms (e.g. single words and noun phrases).
3. Probabilistic learning or classification methods for the estimation of the indexing weights making better use of the available relevance information.

We give experimental results for five test collections which show improvements over other indexing methods.

1 Introduction

Document indexing is the task of assigning terms to documents for retrieval purposes. In an early paper on probabilistic retrieval [Maron & Kuhns 60], an indexing model was developed based on the assumption that a document should be assigned 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 w.r.t. the document, was given a theoretical justification in terms of probabilities. In [Fuhr 89a], this approach is generalized 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 w.r.t. 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 [Kwok 89] [Gordon 88], but are of little value with regard to real applications. The model described in [Kwok 86] overcomes this problem by regarding document components as units to which the index term weights relate to; however, experimental evaluations showed that this model is inferior to non-probabilistic indexing approaches [Kwok & Kuan 88]. A different model for using probabilistic indexing weights in retrieval is described in [Robertson et al. 81] as the "2-Poisson-Independence" model, but also

*This study was supported in part by the National Science Foundation under grant IRI 87-02735

had little success (mainly because of parameter estimation problems). In contrast to these results, the approaches developed in [Croft 81] [Croft 83] [Wong & Yao 89] 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 w.r.t. 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 give a brief introduction to the binary independence indexing model, which forms the theoretical justification for our probabilistic indexing weights. Then we describe the basic concepts and procedures of our indexing approach. Section 4 outlines the test setting and the parameters investigated in our experiments, followed by the presentation of the experimental results in section 5.

2 The binary independence indexing model

The binary independence indexing model described in the following is based on the indexing model from [Maron & Kuhns 60] (see also [Fuhr 89a]).

Let $\underline{Q} = \{\underline{q}_1, \underline{q}_2, \underline{q}_3, \ldots\}$ denote the set of queries, where a query $\underline{q}_k$ is regarded as being unique, that is, two requests submitted to an IR system at different times are always treated as different queries. The same approach is taken in the derivation of the "unified model" [Robertson et al. 82], where a single query $\underline{q}_k$ is termed an "individual use". With $\underline{D} = \{\underline{d}_1, \underline{d}_2, \underline{d}_3, \ldots\}$ denoting the set of documents in a collection, the event space of the BII model is $\underline{Q} \times \underline{D}$. As in any probabilistic model the probabilities relate to representations of documents and queries instead to the objects itself (see [Fuhr 89b]), let $Q = \{q_1, q_2, q_3, \ldots\}$ and $D = \{d_1, d_2, d_3, \ldots\}$ denote the corresponding sets of representations of queries and documents. (In the unified model, the set of queries having the same query representation q_k is called the "class of similar uses"). In the case of the BII model, 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 of q_k of a query $\underline{q}_k$ is a subset $q_k^T \subset T$. Below, we will also use a binary vector $\vec{x}_k = (x_{k_1}, \ldots, x_{k_n})$ instead of q_k^T, where $x_{k_i} = 1$, if $t_i \in q_k^T$, and $x_{k_i} = 0$ 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 w.r.t. 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 e.g. [Fuhr 89a]). Let us further assume that we have a binary relevance scale $\Re = \{R, \overline{R}\}$ denoting relevant/non-relevant 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 $r_{km} = r(\underline{q}_k, \underline{d}_m) \in \Re$.

The BII model now seeks for an estimate of the probability $P(R|q_k, d_m) = P(R|\vec{x}_k, d_m)$ 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

$$P(R|\vec{x}_k, d_m) = P(R|d_m) \cdot \frac{P(\vec{x}_k|R, d_m)}{P(\vec{x}_k|d_m)} \tag{1}$$

Here $P(R|d_m)$ is the probability that document d_m will be judged relevant to an arbitrary request. $P(\vec{x}_k|R, d_m)$ is the probability that d_m will be relevant to a query with representation $\vec{x}_k$, and $P(\vec{x}_k|d_m)$ 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, first two independence assumptions are made:

- The distribution of terms in all queries is independent:

$$P(\vec{x}_k|d_m) = \prod_{i=1}^{n} P(x_{k_i}|d_m)$$

- The distribution of terms in all queries to which a document with representation d_m is relevant is independent:

$$P(\vec{x}_k|R, d_m) = \prod_{i=1}^{n} P(x_{k_i}|R, d_m)$$

With these assumptions, (1) can be transformed into

$$\begin{aligned} P(R|\vec{x}_k, d_m) &= P(R|d_m) \cdot \prod_{i=1}^{n} \frac{P(x_{k_i}|R, d_m)}{P(x_{k_i}|d_m)} \\ &= P(R|d_m) \cdot \prod_{x_{k_i}=1} \frac{P(x_{k_i}=1|R, d_m)}{P(x_{k_i}=1|d_m)} \prod_{x_{k_i}=0} \frac{P(x_{k_i}=0|R, d_m)}{P(x_{k_i}=0|d_m)} \end{aligned} \quad (2)$$

Now we make an additional simplifying assumption that is also used in [Maron & Kuhns 60]:

- 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. We can transform the elements of the first product by using the relationship

$$\frac{P(x_{k_i}=1|R, d_m)}{P(x_{k_i}=1|d_m)} = \frac{P(R|x_{k_i}=1, d_m)}{P(R|d_m)} = \frac{P(R|t_i, d_m)}{P(R|d_m)}$$

Here $P(R|t_i, d_m)$ is the probabilistic index term weight of t_i w.r.t. d_m, the probability that document d_m will be judged relevant to an arbitrary query, given that it contains t_i. From our model, it follows that d_m^T should contain at least those terms from T for which $P(R|t_i, d_m) \neq P(R|d_m)$. Assuming that $P(R|t_i, d_m) = P(R|d_m)$ for all $t_i \notin d_m^T$, we get the final BII formula

$$P(R|q_k, d_m) = P(R|d_m) \prod_{t_i \in q_k^T \cap d_m^T} \frac{P(R|t_i, d_m)}{P(R|d_m)}. \quad (3)$$

In this form it is nearly impossible to apply the BII model, because there hardly will be enough relevance information available to estimate the probabilities $P(R|t_i, d_m)$ for specific term-document pairs. All attempts in this direction are doomed to fail ([Maron 83] [Kwok 89]).

3 New indexing concepts

The basic ideas for our new approach stem from the Darmstadt Indexing Approach (DIA) [Fuhr 89a] [Biebricher et al. 88]. 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. First, attribute values of the term t_i, the document d_m and their relationship are collected in the *relevance description* $x(t_i, d_m)$. Our approach makes no additional assumptions about the choice of the attributes and the structure of x. So 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 89a]), 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 $P(R|t_i, d_m)$ the probability $P(R|x(t_i, d_m))$. In the former case, we would have to regard a single document d_m with respect to all queries which contain t_i in order to estimate $P(R|t_i, d_m)$. Now we regard the set of all query-document pairs in which the same relevance description x occurs. Here the probability $P(R|x(t_i, d_m))$ is the probability that a document will be judged relevant to an arbitrary 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:

- 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 $P(R|x(t_i, d_m))$. 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 $t_i \notin q_k^T$ for the estimation of $P(R|x(t_i, d_m))$, too. This is a major improvement over other probabilistic IR models, which yield either document- or query-specific estimates. These models can only use the relevance information that is available for the specific document (query), and no information about other documents (queries) is considered by these models. In our approach, 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 an application, the amount of relevance data from which the indexing weights are computed will always increase and therefore improve the probability estimates.
- Relevance descriptions can be defined for different forms of representation. Most other probabilistic IR models are based on a specific form of representation of documents or queries, and for every new form of representation, a different model has to be developed. The independence of our approach 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 in [Fuhr 89a]). 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 [Salton et al. 75], [Croft 86], [Smeaton 86], [Fagan 87], [Fagan 89]. 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 $P(R|t_i, d_m)$ are computed. These estimates are derived from a learning example $L \subset \underline{Q} \times \underline{D} \times \Re$ 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 multi-set of relevance descriptions with relevance judgements $L^x = [(x(t_i, d_m), r_{km}) | t_i \in q_k^T \cap d_m^T \wedge (\underline{q}_k, \underline{d}_m, r_{km}) \in L]$. 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 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 non-equiprobability 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), therefore we will 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 $P(R|x(t_i, d_m))$ as the relative frequency from those elements of L^x that have the same relevance description (in the case of E_x). (Attributes with continous values would have to be discretized for this purpose, see e.g. [Wong & Chiu 87]). As a simple example, assume that the relevance description consists of two elements defined as

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

$$x_2 = \text{number of occurrences of } t_i \text{ in } d_m.$$

Furthermore, assume that we have relevance information about two query-document pairs as shown in table 1. From this table, we can estimate $P(R|(0,1)) = \frac{2}{3}$ by using E_x and $P(R|(0,1)) = \frac{1}{2}$ based on E_{BII}

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 $P(R|x(t_i, d_m))$. 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.

query	document	r_{km}	term	x
q_1	d_1	R	t_1	(0,1)
			t_2	(0,1)
			t_3	(1,2)
q_2	d_2	$\overline{R}$	t_1	(0,2)
			t_4	(0,1)

Table 1: Example for simple estimation of indexing weights

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 [Beinke-Geiser et al. 86] exploits prior knowledge about the relationship between single elements of the relevance description x and the corresponding probability $P(R|x)$ for the development of a discrete indexing function.
- The probabilistic learning algorithm ID3 developed by Quinlan [Quinlan 86] seeks for significant components of $\vec{x}$ that form a probabilistic classification tree [Faißt 90].
- By assuming only pair-wise dependencies among the components of $\vec{x}$, one can apply the tree dependence model [Chow & Liu 68] [Rijsbergen 77] as indexing function [Tietze 89].
- Using logistic regression [Freeman 87] 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 [Pfeifer 90].
- In this paper, we will use least square polynomials (LSP) [Knorz 83] [Fuhr 89a] 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 $y(\underline{q}_k, \underline{d}_m) = y_{km}$ denote a class variable for each element of L with

$$y_{km} = \begin{cases} 1 & \text{if } r_{km} = R \\ 0 & \text{else} \end{cases}$$

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. $\vec{a}$ can be computed by solving the linear equation system [Fuhr 89a]

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

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

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

  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 [Knorz 83] [Fuhr 88] have shown that we need about 50–100 elements per parameter in order to achieve reliable estimates.

4 Test setting

Some experiments with a preliminary version of our approach in combination with controlled vocabulary indexing have been described in [Fuhr 88, 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 in [Salton & Buckley 88]. We use the same representation of queries and documents as the SMART approach here. 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 2. 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 2 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:

- 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 $d_m^T \cap q_k^T \neq \emptyset$.

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

Table 2: Collections used for experiments

- Top 15 documents: Only the top 15 documents for each query (by applying the retrieval function ϱ_{tfidf} 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 $\vec{x}$. Here we consider only information that is also used in the standard SMART indexing procedures [Salton & Buckley 88], which are based on the following parameters:

tf_{mi}: within-document frequency (wdf) of t_i in d_m.
$max\, tf_m$: maximum wdf tf_{mi} of all terms $t_i \in d_m^T$.
n_i: number of documents in which t_i occurs.
$|\underline{D}|$: number of documents in the collection.
$|d_m^T|$: number of different terms in d_m.

With these parameters, we defined the components of the relevance description:

$$\begin{aligned} x_1 &= tf_{mi} \\ x_2 &= 1/max\, tf_m \\ x_3 &= \log(n_i/|\underline{D}|) \\ x_4 &= \log|d_m^T| \end{aligned}$$

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

$$\begin{aligned} \vec{v}_L &= (1, x_1, x_2, x_3, x_4) \\ \vec{v}_Q &= (1, x_1, x_2, x_3, x_4, x_1^2, x_1x_2, x_1x_3, x_1x_4, x_2^2, x_2x_3, x_2x_4, x_3^2, x_3x_4, x_4^2) \\ \vec{v}_{tfidf} &= (1, x_1x_2x_3, x_1x_2, x_3, x_4) \end{aligned}$$

So we have the indexing functions

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

e_L is a linear function 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 $tf \times idf$ [Salton & Buckley 88].

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 & Buckley 88]). 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_{mi} for each term in a document is computed:

$$\alpha_{mi} = (0.5 + 0.5 \frac{tf_{mi}}{max tf_m}) \cdot \log \frac{n_i}{|D|}.$$

These weights are further normalized by the factor

$$w_m = \sqrt{\sum_{t_i \in d_m^T} \alpha_{mi}^2}$$

So the final indexing weight for a term t_i in a document d_m according to the $tf \times idf$ formula yields

$$u_{mi} = \frac{\alpha_{mi}}{w_m}.$$

In the retrieval process, the indexing weights u_{mi} are used by the retrieval function $\varrho(q_k, d_m)$ which computes a relevance value for each query-document pair. Then the documents are ranked by decreasing relevance values. In our experiments, we only considered the scalar product as retrieval function with

$$\varrho(q_k, d_m) = \sum_{t_i \in q_k^T \cap d_m^T} c_{ki} \cdot u_{mi}.$$

Here c_{ki} denotes the weight of the term t_i with respect to the query q_k. As mentioned in [Wong & Yao 89], this retrieval function can be given a utility theoretic interpretation in the case of probabilistic indexing weights u_{mi}: The weight c_{ki} can be regarded as the utility of the term t_i, 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_{ki}, 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 $c_{ki} = 1$ for all $t_i \in q_k^T$.
- ϱ_{tf}: The query terms weight c_{ki} is set equal to the number of occurrences tf_{ki} of t_i 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_{ki} (and $max\, tf_k$) are regarded instead of the within-document frequencies.

For evaluation, the standard SMART evaluation routines were taken, and then the average precision value at the recall points 0.25, 0.50 and 0.75 is considered as global retrieval measure.

5 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 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)?

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%

Table 3: Average precision values for learning and test samples (ϱ_{tfidf}, $tf \times idf$)

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 3 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 2); 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 in [Salton & Buckley 88], the combination of ϱ_{tfidf} and $tf \times idf$ is not appropriate for the NPL collection, since terms occur at most once in the queries of this collection, and are possibly from a controlled vocabulary. Therefore, our measure of similarity may be invalid for the NPL collection. This assumption is also supported by the results presented in the following.

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 4 and 5. 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_{tfidf}, since e_L is the only linear function. 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, 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.

collection	e_L full	e_L top	e_Q full	e_Q top	e_{tfidf} full	e_{tfidf} top
CACM	0.2889	0.3024	0.3260	0.3187	0.3010	0.3167
	+ 4.7%		- 2.2%		+ 5.2%	
CISI	0.1034	0.1159	0.1094	0.1231	0.1012	0.1180
	+ 12.1%		+ 12.5%		+ 16.6%	
CRAN	0.3741	0.3786	0.3534	0.3386	0.3493	0.3372
	+ 1.2%		- 4.2%		- 3.5%	
INSPEC	0.1960	0.2033	0.2228	0.1847	0.2093	0.2105
	+ 3.7%		- 17.1%		+ 0.6%	
NPL	0.2109	0.1705	0.1750	0.1285	0.1975	0.1237
	- 19.2%		- 26.6%		- 37.4%	

Table 4: 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_{tfidf} full	e_{tfidf} top
CACM	0.3078	0.3003	0.3669	0.3540	0.3352	0.3234
	- 2.4%		- 3.5%		- 3.5%	
CISI	0.1378	0.1677	0.1542	0.1918	0.1406	0.1711
	+ 21.7%		+ 24.4%		+ 21.7%	
CRAN	0.4157	0.4252	0.4062	0.3924	0.3895	0.3749
	+2.3%		- 3.4%		- 3.7%	
INSPEC	0.2316	0.2286	0.2449	0.2108	0.2031	0.1884
	- 1.3%		- 13.9%		- 7.2%	
NPL	0.2391	0.2777	0.2068	0.1745	0.2709	0.1934
	+ 16.1%		- 15.6%		- 28.6%	

Table 5: Retrieval results using either the top 15 ranked documents or full relevance information (test sample, ϱ_{bin}, E_x)

Event space

Tables 6 and 7 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.

collection	e_L		e_Q		e_{tfidf}	
	E_x	E_{BII}	E_x	E_{BII}	E_x	E_{BII}
CACM	0.3024	0.3117	0.3187	0.3001	0.3167	0.3097
	+ 3.1%		- 5.8%		- 2.2%	
CISI	0.1159	0.1142	0.1231	0.1192	0.1180	0.1166
	- 1.5%		- 3.2%		- 1.2%	
CRAN	0.3786	0.3764	0.3386	0.3279	0.3372	0.3251
	- 0.6%		- 3.2%		- 3.6%	
INSPEC	0.2033	0.2063	0.1847	0.1797	0.2105	0.2021
	+ 1.5%		- 2.7%		- 4.0%	
NPL	0.1705	0.1655	0.1285	0.1205	0.1237	0.1184
	- 2.9%		- 6.2%		- 4.3%	

Table 6: 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	0.3003	0.2980	0.3540	0.3068	0.3234	0.3252
	-0.8%		- 13.3%		+ 0.6%	
CISI	0.1677	0.1606	0.1918	0.1824	0.1711	0.1599
	- 4.2%		- 4.9%		- 6.5%	
CRAN	0.4252	0.4196	0.3924	0.3710	0.3749	0.3514
	- 1.3%		- 5.5%		- 6.3%	
INSPEC	0.2286	0.2318	0.2108	0.2049	0.1884	0.1764
	+ 1.4%		- 2.8%		- 6.4%	
NPL	0.2777	0.2743	0.1745	0.1644	0.1934	0.1843
	- 1.2%		- 5.8%		- 4.7%	

Table 7: Retrieval results using either E_x or E_{BII} (test sample, top, ϱ_{bin})

Indexing functions

In tables 8 and 9, 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 indexing function for each collection which yields the best retrieval results. We would expect that e_Q always performs better than e_L here: Since e_Q contains all the parameters of e_L plus all quadratic combinations of elements of $\vec{x}$, it can be adapted closer to the learning sample. As this assumption does not hold for three of the five collections, the deviations between the theoretical model and our experiments, namely the choice of the retrieval function, should be considered in further experiments. Furthermore, we should investigate the influence of the independence assumptions of our model on these results.

Looking at the test samples, we get more uniform results: For three of the five collections e_L yields the best retrieval results or all indexing functions considered. The CISI and the 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 the learning sample. In the

collection	$tf \times idf$	e_L	e_Q	e_{tfidf}
CACM	0.2604	0.3024 + 16.1%	0.3187 + 22.4%	0.3167 + 21.6%
CISI	0.1188	0.1159 - 2.4%	0.1231 + 3.6%	0.1180 - 0.7%
CRAN	0.3567	0.3786 + 6.1%	0.3386 - 5.1%	0.3372 - 5.5%
INSPEC	0.1706	0.2033 + 19.2%	0.1847 + 8.3%	0.2105 + 23.4%
NPL	0.1580	0.1705 + 7.9%	0.1285 - 18.7%	0.1237 - 21.7%

Table 8: 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	0.2674	0.3003 + 12.3%	0.3540 + 32.4%	0.3234 + 21.6%
CISI	0.1407	0.1677 + 19.2%	0.1918 + 36.3%	0.1711 + 21.6%
CRAN	0.3841	0.4252 + 10.7%	0.3924 + 2.2%	0.3749 - 2.4%
INSPEC	0.1848	0.2286 + 23.7%	0.2108 + 14.1%	0.1884 + 1.9%
NPL	0.2141	0.2777 + 29.7%	0.1745 - 18.5%	0.1934 - 9.7%

Table 9: Probabilistic indexing functions vs. $tf \times idf$ formula (test sample, top, E_x, ϱ_{bin})

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 larger 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 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 10 shows the results for the indexing function e_L in combination with the three retrieval functions ϱ_{bin}, ϱ_{tf} and ϱ_{tfidf}. It can be seen that ϱ_{tf} yields the best results among the retrieval functions (only for the CACM collection ϱ_{tfidf} is slightly better). 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 $\varrho_{tf\,idf}$ 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.

collection	$tf \times idf$	e_L		
	$\varrho_{tf\,idf}$	ϱ_{bin}	ϱ_{tf}	$\varrho_{tf\,idf}$
CACM	0.2963	0.3003 + 1.3%	0.3210 + 8.3%	0.3286 + 10.9%
CISI	0.2099	0.1677 - 20.1%	0.2169 + 3.3%	0.2089 - 0.5%
CRAN	0.3816	0.4252 + 11.4%	0.4280 + 12.2%	0.3929 + 3.0%
INSPEC	0.2489	0.2286 - 8.2%	0.2583 + 3.8%	0.2491 + 0.1%
NPL	0.2138	0.2777 + 29.9%	0.2777 + 29.9%	0.2354 + 10.1%

Table 10: Comparison of different retrieval functions (test sample, top, E_x)

The retrieval results for the probabilistic indexing functions are compared with those of the $tf \times idf$ indexing weights and the $\varrho_{tf\,idf}$ retrieval function. In [Salton & Buckley 88], this combination proves 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. 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).

6 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 non-probabilistic indexing models (like e.g. [Salton & Buckley 88]) or earlier probabilistic models [Croft 81], the indexing weights of the BII model have a clear notion as probabilities in a well-defined event space. For retrospective experiments, the estimation of these probabilistic indexing weights is trivial.
- 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 (similar approaches have been investigated in [Yu & Mizuno 88] and [Fuhr 89c]) which complements the short-term learning method of relevance weighting of search terms. For the latter problem, the retrieval-with-probabilistic-indexing (RPI) model [Fuhr 89a] 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 in [Robertson et al. 82]). 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 w.r.t. 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 method: 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.

Acknowledgement

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

References

Beinke-Geiser, U.; Lustig, G.; Putze-Meier, G. (1986). Indexieren mit dem System DAISY. In: Lustig, G. (ed.) : *Automatische Indexierung zwischen Forschung und Anwendung*, pages 73–97. Olms, Hildesheim.

Biebricher, P.; Fuhr, N.; Knorz, G.; Lustig, G.; Schwantner, M. (1988). The Automatic Indexing System AIR/PHYS — from Research to Application. In: Chiaramella, Y. (ed.) : *11th International Conference on Research and Development in Information Retrieval*, pages 333–342. Presses Universitaires de Grenoble, Grenoble, France.

Chow, C. K.; Liu, C. N. (1968). Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Transactions on Information Theory 14(3)*, pages 462–467.

Croft, W. B. (1981). Document Representation in Probabilistic Models of Information Retrieval. *Journal of the American Society for Information Science 32*, pages 451–457.

Croft, W. B. (1983). Experiments with Representation in a Document Retrieval System. *Information Technology: Research and Development 2*, pages 1–22.

Croft, W. B. (1986). Boolean Queries and Term Dependencies in Probabilistic Retrieval Models. *Journal of the American Society for Information Science 37(2)*, pages 71–77.

Fagan, J. (1987). Automatic Phrase Indexing for Document Retrieval. In: Yu, C. T.; van Rijsbergen, C. J. (ed.) : *Proceedings of the Tenth Annual ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 91–101.

Fagan, J. L. (1989). The Effectiveness of a Nonsyntactic Approach to Automatic Phrase Indexing for Document Retrieval. *Journal of the American Society for Information Science 40(2)*, pages 115–132.

Faißt, S. (1990). *Development of Indexing Functions Based on Probabilistic Decision Trees (in German).* Diploma thesis, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II.

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

Fuhr, N. (1988). *Probabilistisches Indexing und Retrieval.* Dissertation, TH Darmstadt, Fachbereich Informatik.

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

Fuhr, N. (1989b). Optimum Polynomial Retrieval Functions. In: Belkin, N.; van Rijsbergen, C. J. (ed.) : *Proceedings of the Twelfth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, pages 69–76. ACM, New York.

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

Gordon, M. (1988). Probabilistic and Genetic Algorithms for Document Retrieval. *Communications of the ACM 31(10)*, pages 1208–1218.

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

Kwok, K. L.; Kuan, W. (1988). Experiments with Document Components for Indexing and Retrieval. *Information Processing and Management 24(4)*, pages 405–417.

Kwok, K. L. (1986). An Interpretation of Index Term Weighting Schemes Based on Document Components. In: Rabitti, F. (ed.) : *Proceedings of the 1986 ACM Conference on Research and Development in Information Retrieval*, pages 275–283. ACM, New York.

Kwok, K. L. (1989). A Neural Network for Probabilistic Information Retrieval. In: Belkin, N.; van Rijsbergen, C. J. (ed.) : *Proceedings of the Twelfth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, pages 21–30. ACM, New York.

Maron, M. E.; Kuhns, J. L. (1960). On Relevance, Probabilistic Indexing, and Information Retrieval. *Journal of the ACM 7*, pages 216–244.

Maron, M. E. (1983). Probabilistic Approaches to the Document Retrieval Problem. In: Salton, G.; Schneider, H.-J. (ed.) : *Research and Development in Information Retrieval*, pages 98–107. Springer, Berlin et al.

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

Quinlan, J. R. (1986). The Effect of Noise on Concept Learning. In: Michalski, R. S.; Carbonell, J. G.; Mitchell, T. M. (ed.) : *Machine Learning: An Artificial Intelligence Approach , Vol. II*, pages 149–166. Morgan Kaufmann, Los Altos, California.

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

Robertson, S. E.; Van Rijsbergen, C. J.; Porter, M. F. (1981). Probabilistic Models of Indexing and Searching. In: Oddy, R. N.; Robertson, S. E.; Van Rijsbergen, C. J.; Williams, P. W. (ed.) : *Information Retrieval Research*, pages 35–56. Butterworths, London.

Robertson, S. E.; Maron, M. E.; Cooper, W. S. (1982). Probability of Relevance: A Unification of Two Competing Models for Document Retrieval. *Information Technology: Research and Development 1*, pages 1–21.

Salton, G.; Buckley, C. (1988). Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management 24(5)*, pages 513–523.

Salton, G.; Yang, C. S.; Yu, C. T. (1975). A Theory of Term Importance in Automatic Text Analysis. *Journal of the American Society for Information Science 36*, pages 33–44.

Smeaton, A. F. (1986). Incorporating Syntactic Information into a Document Retrieval Strategy: an Investigation. In: *9th International Conference on Research & Development in Information Retrieval*, pages 103–113. ACM, New York.

Tietze, A. (1989). *Approximation of Discrete Probability Distributions by Dependence Trees and their Application as Indexing Functions (in German)*. Diploma thesis, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II.

Wong, A. K. C.; Chiu, D. K. Y. (1987). Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence 9(6)*, pages 796–805.

Wong, S. K. M.; Yao, Y. Y. (1989). A Probability Distribution Model for Information Retrieval. *Information Processing and Management 25(1)*, pages 39–53.

Yu, C. T.; Mizuno, H. (1988). Two Learning Schemes in Information Retrieval. In: Chiaramella, Y. (ed.) : *11th International Conference on Research & Development in Information Retrieval*, pages 201–218. Presses Universitaires de Grenoble, Grenoble, France.