Probabilistic Models in Information Retrieval

NORBERT FUHR
University of Dortmund, Informatik VI, P.O. Box 500500, W-4600 Dortmund, Germany

In this paper, an introduction and survey over probabilistic information retrieval (IR) is given. First, the basic concepts of this approach are described: the probability-ranking principle shows that optimum retrieval quality can be achieved under certain assumptions; a conceptual model for IR along with the corresponding event space clarify the interpretation of the probabilistic parameters involved. For the estimation of these parameters, three different learning strategies are distinguished, namely query-related, document-related and description-related learning. As a representative for each of these strategies, a specific model is described. A new approach regards IR as uncertain inference; here, imaging is used as a new technique for estimating the probabilistic parameters, and probabilistic inference networks support more complex forms of inference. Finally, the more general problems of parameter estimations, query expansion and the development of models for advanced document representations are discussed.

Received December 1991

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

1. INTRODUCTION

A major difference between information retrieval (IR) systems and other kinds of information system is the intrinsic uncertainty of IR. Whereas for database systems an information need can always (at least for standard applications) be mapped precisely on to a query formulation, and there is a precise definition of which elements of the database constitute the answer, the situation is much more difficult in IR; here neither a query formulation can be assumed to represent uniquely an information need, nor is there a clear procedure that decides whether a DB object is an answer or not. (Boolean IR systems are not an exception to this statement; they only shift all problems associated with uncertainty to the user.) As the most successful approach for coping with uncertainty in IR, probabilistic models have been developed.

According to the definition of the desired set of answers to a query in an IR system, two major approaches in probabilistic IR can be distinguished. The classical approach is based on the concept of relevance, that is, a user assigns relevance judgements to documents w.r.t. his query, and the task of the IR system is to yield an approximation of the set of relevant documents. The new approach formulated by van Rijsbergern overcomes this subjective definition of an answer in an IR system by generalising the proof-theoretic model of database systems towards uncertain inference.

In this paper, an introduction into past and current research in probabilistic IR is given. The major goal here is to present important concepts of this field of research, while no attempt is made to give a complete survey over work in this area. In the following, the classical approach in probabilistic IR is presented in Sections 2 and 3, while Section 4 describes the new direction. In Section 5, some general problems of both approaches are discussed. An outlook to future research areas finishes the paper.

2. BASIC CONCEPTS OF RELEVANCE MODELS

2.1 The binary independence retrieval model

In order to introduce some basic concepts of the classical approach to probabilistic IR, we first present a fairly simple model, the so-called binary independence retrieval (BIR) model. This model will be introduced more informally, whereas the precise assumptions underlying this model will be developed throughout the following sections.

In the BIR model, as in most other probabilistic IR models, we seek to estimate the probability that a specific document d_m will be judged relevant w.r.t. a specific query q_k. In order to estimate this probability (denoted as $P(R|q_k, d_m)$) in the following, we regard the distribution of terms within the documents of the collection. (In general, a term is any non-trivial word reduced to its word stem, but see also Section 5.3 for other kinds of term.) The basic assumption is that terms are distributed differently within relevant and non-relevant documents. This assumption, known as the 'cluster hypothesis', has been verified experimentally already in Ref. 1. Let $T = \{t_1, \ldots, t_n\}$ denote the set of terms in the collection. Then we can represent the set of terms d_m^T occurring in document d_m as a binary vector $\mathbf{x} = (x_1, \ldots, x_n)$ with $x_i = 1$, if $t_i \in d_m^T$ and $x_i = 0$ otherwise.

Now we distinguish only between documents containing different sets of terms, so instead of estimating $P(R|q_k, d_m)$ for a specific document d_m, we actually estimate the probability $P(R|q_k, \mathbf{x})$, where different documents containing the same set of terms will yield the same estimate of probability of relevance. In addition, the BIR model assumes a query q_k to be just a set of terms $q_k^T \subset T$. In Section 4.2 we shall also discuss the case of other forms of query.

In order to derive a formula for this probability, we shall apply two kinds of transformation that are frequently used for the derivation of probabilistic IR models:

(1) application of Bayes' theorem (in the form $P(a|b) = P(b|a)\,P(a)/P(b)$);

(2) usage of odds instead of probabilities, where $O(y) = P(y)/P(\bar{y}) = P(y)/[1-P(y)]$.

This way, we can compute the odds of a document represented by a binary vector $\mathbf{x}$ being relevant to a query q_k as

$$O(R|q_k, \mathbf{x}) = \frac{P(R|q_k, \mathbf{x})}{P(\bar{R}|q_k \mathbf{x})} = \frac{P(R|q_k)}{P(\bar{R}|q_k)} \cdot \frac{P(\mathbf{x}|R, q_k)}{P(\mathbf{x}|\bar{R}, q_k)}. \quad (1)$$

Now additional independence assumptions are needed in order to arrive at a formula that is applicable for retrieval of documents. As has been pointed out in a recent paper by Cooper,[2] the assumption underlying the BIR is in fact not a set of independence assumptions (from which the name of the model is derived), but rather the assumption of linked dependence of the form

$$\frac{P(x \mid R, q_k)}{P(x \mid \bar{R}, q_k)} = \prod_{i=1}^{n} \frac{P(x_i \mid R, q_k)}{P(x_i \mid \bar{R}, q_k)}. \tag{2}$$

This assumption implies that the ratio between the probabilities of x occurring in the relevant and the non-relevant documents is equal to the product of the corresponding ratios of the single terms. Of course, the linked dependence assumption does not hold in reality. However, it should be regarded as a first-order approximation. In Section 2.6, we shall discuss better approximations.

With assumption (2) we can transform (1) into

$$O(R \mid q_k, x) = O(R \mid q_k) \prod_{i=1}^{n} \frac{P(x_i \mid R, q_k)}{P(x_i \mid \bar{R}, q_k)}.$$

The product of this equation can be split according to the occurrence of terms in the current document:

$$O(R \mid q_k, x) = O(R \mid q_k) \prod_{x_i=1} \frac{P(x_i = 1 \mid R, q_k)}{P(x_i = 1 \mid \bar{R}, q_k)} \cdot \prod_{x_i=0} \frac{P(x_i = 0 \mid R, q_k)}{P(x_i = 0 \mid \bar{R}, q_k)}.$$

Now let $p_{ik} = P(x_i = 1 \mid R, q_k)$ and $q_{ik} = P(x_i = 1 \mid \bar{R}, q_k)$. In addition, we assume that $p_{ik} = q_{ik}$ for all terms not occurring in the set q_k^T of query terms. This assumption is also subject to change, as discussed in Section 5.2. With these notations and simplifications, we arrive at the formula

$$O(R \mid q_k, x) = O(R \mid q_k) \prod_{t_i \in d_m^T \cap q_k^T} \frac{p_{ik}}{q_{ik}} \prod_{t_i \in q_k^T \setminus d_m^T} \frac{1 - p_{ik}}{1 - q_{ik}} \tag{3}$$

$$= O(R \mid q_k) \prod_{t_i \in d_m^T \cap q_k^T} \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})} \prod_{t_i \in q_k^T} \frac{1 - p_{ik}}{1 - q_{ik}}. \tag{4}$$

In the application of this formula, one is mostly interested only in a ranking of the documents with respect to a query, and not in the actual value of the probability (or odds) of relevance. From this point of view, since the second product of Equation (4) as well as the value of $O(R \mid q_k)$ are constant for a specific query, we only have to consider the value of the first product for a ranking of the documents. If we take the logarithm of this product, the retrieval status value (RSV) of document d_m for query q_k is computed by the sum

$$\sum_{t_i \in d_m^T \cap q_k^T} c_{ik} \quad \text{with} \quad c_{ik} = \log \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})}.$$

The documents are ranked according to descending RSVs.

In order to apply the BIR model, we have to estimate the parameters p_{ik} and q_{ik} for the terms $t_i \in q_k^T$. This can be done by means of relevance feedback. For that, let us assume that the IR system has already retrieved some documents for query q_k (in Section 5.1, we show how the parameters of the BIR model can be estimated without relevance information). Now the user is asked to give relevance judgements for these documents. From this relevance feedback data, we can estimate the parameters of the BIR model as follows. Let f denote the number of documents presented to the user, of which r have been judged relevant. For a term t_i, f_i is the number among the f documents in which t_i occurs, and r_i is the number of relevant documents containing t_i. Then we can use the estimates $p_{ik} \approx r_i/r$ and $q_{ik} \approx (f_i - r_i)/(f - r)$. (Better estimation methods are discussed in Section 5.1.)

We illustrate this model by giving an example. Assume a query q containing two terms, that is $q^T = \{t_1, t_2\}$. Table 1 gives the relevance judgements from 20 documents together with the distribution of the terms within these documents.

For the parameters of the BIR model, we get $p_1 = 8/12 = 2/3$, $q_1 = 3/8$, $p_2 = 7/12$ and $q_2 = 4/8$. So we get the query term weights $c_1 = \log 10/3 \approx 1.20$ and $c_2 = \log 7/5 \approx 0.33$. Based on these weights, documents are ranked according to their corresponding binary vector x in the order $(1,1)-(1,0)-(0,1)-(0,0)$. Obviously this ranking is correct for our example.

Table 2. Estimates for the probability of relevance for our example.

	$P(R \mid q, x)$	
x	BIR	actual
(1, 1)	0.76	0.8
(1, 0)	0.69	0.67
(0, 1)	0.48	0.5
(0, 0)	0.4	0.33

In addition, we can also compute the estimates for the probability of relevance according to Equation (3). With $O(R \mid q) = 12/8$, we get the estimates shown in Table 2 for the different x vectors, where they are compared with the actual values. Here the estimates computed by the two methods are different. This difference is due to the linked dependence assumption underlying the BIR model.

We have described this model in detail because it illustrates a number of concepts and problems in probabilistic IR. In the following, we shall first describe the major concepts.

Table 1. Example for the BIR model.

d_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x_1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
x_2	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
$r(q, d_i)$	R	R	R	R	$\bar{R}$	R	R	R	R	$\bar{R}$	$\bar{R}$	R	R	R	$\bar{R}$	$\bar{R}$	$\bar{R}$	R	$\bar{R}$	$\bar{R}$

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

2.2 A conceptual model for IR

In our example, it may be argued that the approach chosen in the BIR model for representing documents may be rather crude, and that a more detailed representation of documents may be desirable, especially since documents in an IR system may be rather complex. The relationship between documents and their representations (and similarly for queries) can be illustrated best by regarding the conceptual IR model depicted in figure 1.

$$\begin{tikzcd}
 & \underline{Q} \arrow[r, "\alpha_Q"] \arrow[dl, "\text{Rel.}"'] & Q \arrow[r, "\beta_Q"] & Q^D \arrow[dr, "\rho"] & \\
\mathcal{R} & & & & \text{IR} \\
 & \underline{D} \arrow[r, "\alpha_D"] \arrow[ul, "\text{Judg.}"] & D \arrow[r, "\beta_D"] & D^D \arrow[ur] &
\end{tikzcd}$$

Figure 1. Conceptual model.

Here $\underline{d_m}$ and $\underline{q_k}$ denote the original document and query, respectively. In our terminology, a query is unique (i.e. a specific information need of a specific user), so two queries from different users (or issued from the same user at different times) can never be identical. This concept of a query has been introduced first in Ref. 3, where it was termed a 'use'. Between a document and a query, there exists a relevance relationship as specified by the user. Let $\mathcal{R} = \{R, \bar{R}\}$ denote the set of possible relevance judgements, then the relevance relationship can be regarded as a mapping $r: \underline{Q} \times \underline{D} \rightarrow \mathcal{R}$. Since an IR system can only have a limited understanding of documents and queries, it is based on representations of these objects, here denoted as d_m and q_k. These representations are derived from the original documents and queries by application of the mappings α_D and α_Q, respectively. In the case of the BIR model the representation of a document d_m is a set of terms, namely the set d_m^T. For queries, the representation contains, in addition to the set of query terms q_k^T, a set of relevance judgements $q_k^J = \{(d_m, r(\underline{d_m}, \underline{q_k}))\}$. So in our conceptual model the representation of an object comprises the data relating to this object that is actually used in the model. Different IR models may be based on quite different representations. For example, in Boolean systems with free text search, the document representation is a list of strings (words), and the query representation is a Boolean expression, where the operands may be either single words or adjacency patterns (comprised of words and special adjacency operators).

For the models regarded here, there is an additional level of representation, which we call description. As can be seen from the BIR model, the retrieval function does not relate explicitly to the query representation, it uses the query-term weights derived from the relevance judgements instead. We call the arguments of the retrieval function the description of documents and queries. In the BIR model the document representation and the description are identical. The query description, however, is a set of query terms with the associated query-term weights, that is, $q_k^D = \{(t_i, c_{ik})\}$.

Representations are mapped on to descriptions by means of the functions β_D and β_Q, respectively. Based on the descriptions, the retrieval function $\rho(q_k^D, d_m^D)$ computes the retrieval status value, which is a real number in general. This conceptual model can be applied to probabilistic IR models as well as to other models. Especially when comparing the quality of different models, it is important to consider the representations used within these models. With respect to representations, two directions in the development of probabilistic IR models can be distinguished.

(1) Optimisation of retrieval quality for a fixed representation. For example, there have been a numer of attempts to overcome the limitations of the BIR model by revising the linked dependence assumption and considering certain other forms of term dependencies (see Section 2.6). In these approaches, documents are still represented as sets of terms.

(2) Development of models for more detailed representations of queries and documents. Since the document representation used within the BIR model is rather poor, it is desirable to derive models that can consider more detailed information about a term in a document, e.g. its within-document frequency, or the output of advanced text analysis methods (e.g. for phrases in addition to words). We shall discuss this issue in Section 5.3.

2.3 Parameter learning in IR

We can make another observation with the BIR model: this model makes very poor use of the relevance feedback information given by the user, since this information is only considered in the ranking process for the current query. For a new query, none of this data can be used at all. If we regard probabilistic IR models as (parameter) learning methods, then three different approaches as shown in Fig. 2 can be distinguished. 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 $\underline{Q}_L \times \underline{D}_L \times T_L$ of $\underline{Q} \times \underline{D} \times T$ (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 query-related 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 $\underline{D}_L$, 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 $\underline{D}$.

Document-related learning is orthogonal to the query-related strategy: probabilistic indexing models (see Section 3.1) collect relevance feedback data for a specific document d_m from a set of queries $\underline{Q}_L$, 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

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

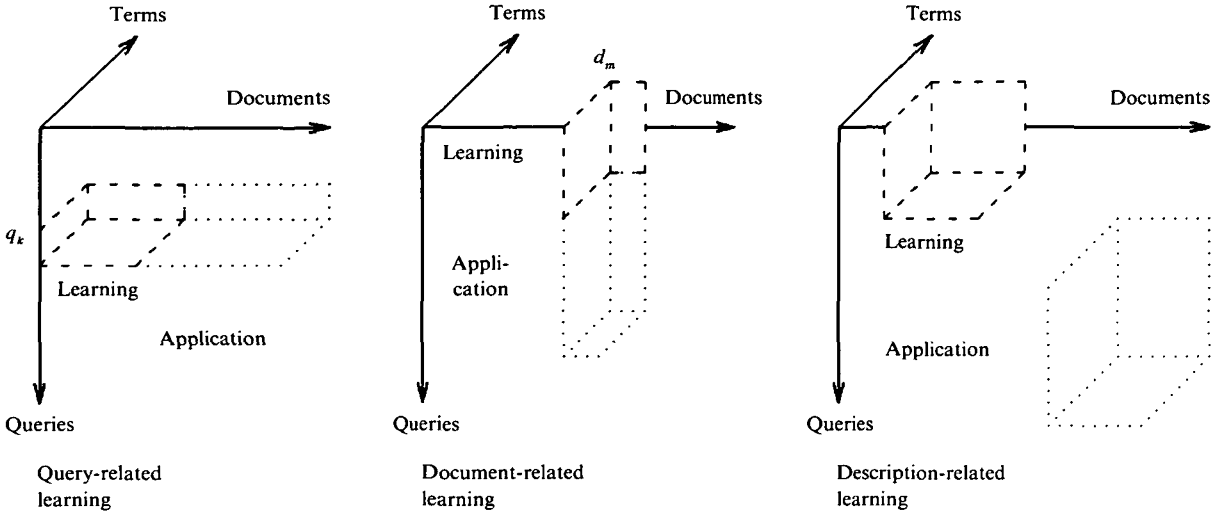

Figure 2. Learning approaches in IR.

document in real databases, so it is almost impossible to estimate the parameters of this approach.

The major drawback of these two approaches is their limited application range, since the application phase is either restricted to a single query or to a single document (as in the case of the BII model). In order to overcome these deficiencies, we must introduce abstractions from specific documents, queries and terms. This description-related strategy has been implemented first within the Darmstadt Indexing Approach[4] by introducing the concept of relevance descriptions. Like 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 $\underline{Q}_L \times \underline{D}_L \times T_L$. For the application phase, there are no restrictions concerning the subset $\underline{Q}_A \times \underline{D}_A \times T_A$ 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. Since this approach is based on descriptions of IR objects instead of the objects itself, we call it description-oriented, in contrast to the model-oriented approaches described before (see also Section 2.6).

2.4 Event space

In the presentation of the BIR model, we have not specified the event space to which the probabilities relate. Now we shall define this event space, which is also underlying most probabilistic models.

The event space is $\underline{Q} \times \underline{D}$. A single element of this event space is a query-document pair $(\underline{d}_m, \underline{q}_k)$, where we assume that all these elements are equiprobable. Associated with each element is a relevance judgement $r(\underline{d}_m, \underline{q}_k) \in \mathcal{R}$. We assume that the relevance judgements for different documents w.r.t. the same query are independent of each other. This is a rather strong assumption. Variations of this assumption are discussed in Refs 5 and 6, but most of these variations lead to models that can hardly be applied in practice. The event space can be illustrated as a matrix as shown in Fig. 3, where a query corresponds to a single row and a document to a column. The relevance judgements can be assumed as the values of the elements of this matrix. Since a retrieval system deals with representations of documents and queries, it treats different queries or documents having identical representations the same. This fact is illustrated here by mapping adjacent rows to a single query representation q_k and adjacent columns to a single document representation d_m. With this model, the interpretation of the probability of relevance $P(R \mid q_k, d_m)$ is obvious: the pair (q_k, d_m) corresponds to the set of elements having the same representation (shown as a submatrix here). So $P(R \mid q_k, d_m)$ is the proportion of elements in this set that have been judged relevant. One might argue that this explanation is oversimplified, since in real collections there are hardly ever two objects that share the same representation. But we regard collections as samples of

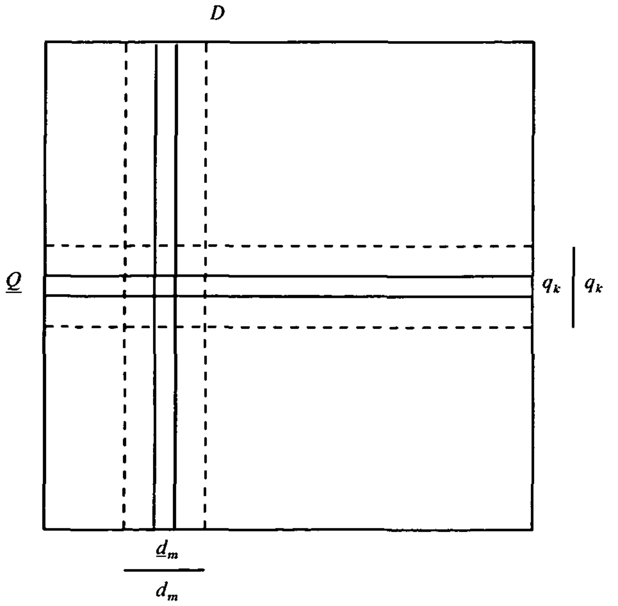

Figure 3. Event space of relevance models.

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

possibly infinite sets of documents and queries, where there might be several (up to infinity) objects with the same representation. Especially with regard to the poor representation of retrieval objects that is actually in use (in comparison to the human understanding of documents and queries), it is obvious that a single representation may stand for a number of different objects.

2.5 The Probability Ranking Principle

The Probability Ranking Principle (PRP) represents the theoretical justification of probabilistic IR models. It shows how optimum retrieval quality can be achieved. Optimum retrieval is defined w.r.t. representations. In contrast, perfect retrieval relates to the objects itself, saying that all relevant documents should be ranked ahead of any nonrelevant one. But as an IR system is based on representations, perfect retrieval is not a suitable goal. Optimum retrieval has only been defined precisely for probabilistic IR, where the optimality can be proved theoretically. The 'Probability Ranking Principle' described in Ref. 5 says that optimum retrieval is achieved when documents are ranked according to decreasing values of the probability of relevance (with respect to the current query). The decision-theoretic justification of the PRP is as follows (in Ref. 5, justifications w.r.t. retrieval measures are also given). Let $\bar{C}$ denote the costs for the retrieval of a nonrelevant document, and C the costs for the retrieval of a relevant document. Now the decision-theoretic rule says that document d_m should be retrieved next from the collection if

$$CP(R \mid q_k, d_m) + \bar{C}(1 - P(R \mid q_k, {}_m)) \leqslant CP(R \mid q_k, d_j) + \bar{C}(1 - P(R \mid q_k, d_j))$$

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

$$P(R \mid q_k, d_m) \geqslant P(R \mid q_k, d_j).$$

So we have the rule that documents should be ranked according to their decreasing probability of being relevant. The PRP can be extended to cope with multivalued (ordinal) relevance scales instead of binary ones, as shown in Ref. 7. Assume that for n relevance values with $R_1 < R_2 < \ldots < R_n$ the correponding costs for the retrieval of a document with that retrieval judgement are $C_1, C_2, \ldots, C_n$. Then documents should be ranked according to their expected costs

$$EC(q_k, d_m) = \sum_{l=1}^{n} C_l P(R_l \mid q_k, d_m).$$

In contrast to the binary case where only the probability $P(R \mid q_k, d_m)$ has to be estimated for a query document pair, here $n-1$ estimates $P(R_l \mid q_k, d_m)$ are needed in order to rank the documents w.r.t. a query. Furthermore, the actual values of the cost factors C_l are required in order to produce a ranking, since they cannot be eliminated as in the binary case. Using multivalued relevance scales instead of binary ones seems to be more appropriate; however, the only experimental results comparing binary vs. multivalued relevance scales published so far did not show any differences in terms of retrieval quality.[8] So it might be feasible to offer a multivalued relevance scale for the users of a probabilistic IR system, but this scale can be mapped on to a binary one for the calculations performed by the system.

With multivalued relevance scales, we can also draw a connection to fuzzy retrieval approaches (see Ref. 9 for a survey on this subject), where the relevance scale is assumed to be continuous, that is, a relevance judgement now is a real number $r \in [0, 1]$. In this case, the probability distribution $P(R_l \mid q_k, d_m)$ from above is replaced by a density function $p(r \mid q_k, d_m)$ as well as the cost factors C_l by a cost function $c(r)$. This way, fuzzy and probabilistic retrieval can be combined. In contrast, pure fuzzy retrieval approaches seem to be inappropriate from the point of view of probabilistic IR, since the intrinsic aspect of uncertainty in IR is ignored in these approaches.

2.6 Model-oriented vs. description-oriented approaches

The formulation of the PRP acts as a goal for any probabilistic IR model. In general, the optimum retrieval quality as specified by the PRP cannot be achieved by a real system. For example, in the BIR model, we would have to know the exact probabilities $P(R \mid q_k, \mathbf{x})$ for all binary vectors $\mathbf{x}$ occurring in the document collection. Except for rather trivial cases, these probabilities can hardly be estimated directly, because the number of different representations is too large in comparison to the amount of feedback data available (so there would not even be any observation for most representations). In order to overcome this difficulty, additional simplifying assumptions are needed. With regard to the nature of these assumptions, two kinds of approaches have been developed, as follows.

Model-oriented approaches (like the BIR model, for example) are based on certain probabilistic independence assumptions concerning the elements of the representations (e.g. single terms or pairs, triplets of terms). In these approaches, first probability estimates relating to the representation elements are computed. Then, by applying the independence assumptions, the estimates for the different representations can be derived.

Description-oriented approaches are similar to feature-based pattern-recognition methods. Given the representations of queries and documents, first a set of features for query-document pairs is defined, and each pair is mapped on to a feature vector $\mathbf{x}(q_k, d_m)$. (A specific feature vector could be for example the binary vector $\mathbf{x} = \mathbf{x}(d_m)$ as defined in the BIR model; however, since this definition does not consider the query, the resulting retrieval function would be query-specific.) With the help of a learning sample containing query-document pairs with their corresponding relevance judgements, a probabilistic classification function $e(\mathbf{x})$ that yields estimates of the probability $P(R \mid \mathbf{x}(q_k, d_m))$ is developed (see Section 3.4).

Because of the problem of specifying the feature vector, description-oriented approaches are more heuristical in comparison to model-oriented ones. On the other hand, the assumptions underlying the description-oriented approach do not have to be made as explicit as in the model-oriented case. The most important advantage of description-oriented approaches is their adaptability to rather complex representations, where it

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

is hard to find appropriate independence assumptions. Especially with regard to advanced text analysis methods, this feature seems to be rather important.

As a general property of both kinds of approaches, we can see that the additional assumptions are only approximations to reality. For example, we can hardly expect that terms are distributed independently in documents (as suggested by the BIR model). A similar statement holds for the description-oriented approaches. This fact makes the main difference between optimum retrieval quality and the actual performance of a model. The other reason is the problem of parameter estimation. Without going into the details of parameter estimation here (but see Section 5.1), we can describe the general problem by using the example of the BIR model. The direct estimation of the probabilities $P(R \mid q_k, x)$ vs. the computation of this parameter by means of the BIR model are two extreme possibilities where either the probabilities cannot be estimated in a real application or the independence assumptions seem to be too strong. It is possible to develop variants of the BIR model where only pairs or triplets of terms are assumed to be independent of each other (see e.g. Refs 10 and 11 for such models and 12, chapter 8 for a general survey on probabilistic dependence models). With these models, however, more parameters have to be estimated from less observations for each parameter. For example, in the tree dependence model developed by van Rijsbergen which considers pairwise dependencies,[13] the parameters to be estimated for a dependent pair (t_i, t_j) are $P(x_i = 1, x_j = 1 \mid R)$, $P(x_i = 1, x_j = 0 \mid R)$ $P(x_i = 0, x_j = 1 \mid R)$ and $P(x_i = 0, x_j = 0 \mid R)$ (plus the corresponding estimates for nonrelevant documents). In contrast, the BIR model only requires the parameters $P(x_i = 1 \mid R)$ and $P(x_i = 0 \mid R)$ for the relevant documents, so the tree dependence model splits the learning data required for estimating these parameters according to the value of x_j. As a consequence, experimental evaluations showed that the gain from improved independence assumptions does not outweigh the loss from increased estimation errors.

3. SURVEY OVER RELEVANCE MODELS

In this section we shall first present two probabilistic models that are representative of second and third different learning strategies as described above. Then we shall discuss models that aim to overcome the simple representation of the BIR model.

3.1 The binary independence indexing model

The binary independence indexing (BII) model[14] is a variant of the very first probabilistic IR model, namely the indexing model of Maron and Kuhns.[15] Whereas the BIR model regards a single query w.r.t. a number of documents, the BII model observes one document in relation to a number of queries submitted to the system. In this model, the representation q_k of a query q_k is a set of terms $q_k^T \subset T$. As a consequence, the BII model will yield the same ranking for two different queries formulated with the same set of terms. In the following we shall also use a binary vector $z_k = (z_{k_1}, \ldots, z_{k_n})$ instead of q_k^T, where $z_{k_i} = 1$, if $t_i \in q_k^T$, and $z_{k_i} = 0$ otherwise. The document representation is not further specified in the BII model, and below we shall show that this is a major advantage of this model. In the following, we shall 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 shall call d_m^T 'the set of terms occurring in the document' in the following, although the model can also be applied in situations where the elements of d_m^T are derived from the document text with the help of a thesaurus (see e.g. Ref. 16). 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

$$P(R \mid z_k, d_m) = P(R \mid d_m) \frac{P(z_k \mid R, d_m)}{P(z_k \mid d_m)}. \tag{5}$$

Here $P(R \mid d_m)$ is the probability that document d_m will be judged relevant to an arbitrary request. $P(z_k \mid R, d_m)$ is the probability that d_m will be relevant to a query with representation z_k, and $P(z_k \mid d_m)$ is the probability that such a query will be submitte to the system.

Assuming that the distribution of terms in all queries to which a document with representation d_m is independent:

$$P(z_k \mid R, d_m) = \prod_{i=1}^{n} P(z_{k_i} \mid R, d_m)$$

and the additional simplifying assumption that 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, we get the ranking formula

$$P(R \mid z_k, d_m) = \frac{\prod_{i=1}^{n} P(z_{k_i})}{P(z_k)} P(R \mid d_m) \prod_{z_{k_i}=1} \frac{P(R \mid z_{k_i} = 1, d_m)}{P(R \mid d_m)} \prod_{z_{k_i}=0} \frac{P(R \mid z_{k_i} = 0, d_m)}{P(R \mid d_m)}. \tag{6}$$

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 w.r.t. q_k.

$P(R \mid z_{k_i} = 1, d_m) = P(R \mid t_i, d_m)$ 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 arbitary 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 $P(R \mid t_i, d_m) \neq P(R \mid d_m)$. Assuming that $P(R \mid t_i, d_m) = P(R \mid d_m)$ for all $t_i \notin d_m^T$, the final BII formula yields

$$P(R \mid q_k, d_m) = c_k / P(R \mid d_m) \prod_{t_i \in q_k^T \cap d_m^T} \frac{P(R \mid t_i, d_m)}{P(R \mid d_m)}. \tag{7}$$

However, in this form the BII model can hardly be applied, because in general there will not be enough relevance information available for estimating the probabilities $P(R \mid t_i, d_m)$ for specific term-document pairs. In order to overcome this difficulty, one can assume a document to consist of independent components (e.g. sentences or words) to which the indexing weights relate, but experimental evaluations

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

showed only moderate retrieval results for this approach.[17]

3.2 A description-oriented indexing approach

As a more successful method, the application of the third learning strategy as outlined above has been devised. This learning strategy leads to a description-oriented approach, where features of terms in documents are regarded instead of the document–term pairs themselves. The basic ideas of this approach have been developed within the framework of the Darmstadt Indexing Approach (DIA).[16, 18] Within the DIA, the indexing task is subdivided into a description step and a decision step. In the description step, relevance descriptions for term–document pairs (t_i, d_m) are formed, where a relevance description $x(t_i, d_m)$ contains values of attributes of the term t_i, the document d_m and their relationship. Since this approach makes no additional assumptions about the choice of the attributes and the structure of x, the actual definition of relevance descriptions can be adapted to the specific application context, namely the representation of documents and the amount of learning data available. For example, in the work described in Ref. 14, the following elements were defined: $x_1 = tf_{mi}$, the within-document frequency (wdf) of t_i *in* d_m; $x_2 =$ the inverse of the maximum wdf of a term in d_m; $x_3 =$ inverse document frequency of t_i in the collection; $x_4 = \log|d_m^T|$ (numbers of terms in d_m); $x_5 = 1$, if t_i occurs in the title of d_m, and 0 otherwise.

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. The probabilistic index term weights $P(R \mid x(t_i, d_m))$ are derived from a learning example $L \subset \underline{Q} \times \underline{D} \times \mathscr{R}$ 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 multi-set (bag) of relevance descriptions with relevance judgements $L^x = [(x(t_i, d_m), r_{km}) \mid t_i \in q_k^T \cap d_m^T \wedge (\underline{q}_k, \underline{d}_m, r_{km}) \in L]$. From this set with multiple occurrences of elements, the parameters $P(R \mid x(t_i, d_m))$ could be estimated directly by computing the corresponding relative frequencies. However, better estimates can be achieved by applying probabilistic classification procedures as developed in pattern recognition or machine learning. Within the DIA, this classification procedure yielding approximations of $P(R \mid x(t_i, d_m))$ is termed an indexing function $e(x(t_i, d_m))$. Several probabilistic classification algorithms have been used for this purpose (see e.g. Ref. 14). Here we want to describe briefly the application of least-square polynomials (LSP)[19, 16] as indexing functions, where we furthermore restrict to the case of linear functions. So our indexing function yields $e(x) = \boldsymbol{a}^T x$, where $\boldsymbol{a}$ 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 $y_{km} = 1$ if $r_{km} = R$ and $y_{km} = 0$ otherwise. Then the coefficient vector $\boldsymbol{a}$ is estimated such that it minimises the squared error $E((y - \boldsymbol{a}^T x)^2)$, where $E(\cdot)$ denotes the expectation. The coefficient vector $\boldsymbol{a}$ can be computed by solving the linear equation system (see Ref. 8)

$$E(xx^T)\,\boldsymbol{a} = E(xy). \tag{8}$$

As an approximation for the expectations, the corresponding arithmetic means from the learning sample are taken. The major advantage of this indexing approach is its flexibility w.r.t. the representation of documents, which becomes important when advanced text analysis methods are used (e.g. noun phrases in addition to words, see for example Ref. 8).

3.3 The 2-Poisson model

On the other hand, one might prefer to have a more explicit model relating to the elements of the representation. One such approach is the 2-Poisson model. This model has been proposed first by Bookstein and Swanson.[20] Similar to the indexing model described above, the Bookstein/Swanson model seeks for the decision whether an index term should be assigned to a document or not. So there are two classes of documents with respect to a specific term. Now the number of occurrences tf_{im} of the term t_i within the document d_m is regarded, and it is assumed that the distribution of this feature is different in the two document classes. As a simple probabilistic model, Bookstein and Swanson assumed a Poisson distribution in each of these classes. For a specific document class K_{ij}, let λ_{ij} denote the expectation of the wdf of t_i. Then the probability that a document contains l occurrences of t_i, given that it belongs to class K_{ij} is

$$P(tf_{im} = l \mid d_m \in K_{ij}) = \frac{\lambda_{il}^l}{l!} e^{-\lambda_{ij}}.$$

For a document chosen randomly from the collection, we assume that π_{ij} is the probability that it belongs to class K_{ij}. Then the probability of observing l occurrences within such a document is

$$P(tf_{im} = l \mid d_m \in K_{ij}) = \frac{\lambda_{ij}^l}{l!} e^{-\lambda_{ij}}.$$

In the 2-Poisson model, there are two document classes K_{i1} and K_{i2} for each term, so $\pi_{i1} + \pi_{i2} = 1$. From these equations, the probabilistic index term weighs $P(d_m \in K_{ij} \mid tf_{im} = l)$ can be derived. The parameters π_{ij} and λ_{ij} can be estimated without feedback information from the document collection.

Experimental evaluations of this model were only partially successful. In Refs 21 and 22 the χ^2-test rejected the hypothesis of a 2-Poisson distribution for 62% of the terms tested. Experiments with a higher number of classes (termed n-Poisson model) as described in Ref. 23, also did not give clear improvements. In the study,[24] an improved parameter estimation method is applied in combination with longer documents than in previous evaluations, thus leading to the result that the assumption of an n-Poisson distribution holds for about 70% of all terms.

3.4 Retrieval models for improved document representations

As a consequence of the poor performance of the 2-Poisson model, a so-called non-binary Retrieval model

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

has been proposed as a variant of the BIR model in Refs 25. Instead of indicating only the presence or absence of a term t_i, the elements x_i of the vector representing a document now give the wdf of t_i. As a consequence, parameters $P(x_i = 1 | R)$ and $P(x_i = l | \bar{R})$ for $l = 0, 1, 2, \ldots$ have to be estimated in this model. The results given in Ref. 26 for predictive retrieval did not show any improvements over the BIR model, obviously due to parameter estimation problems. This problem seems to be intrinsic to all approaches that aim to improve the BIR model by using a more detailed document representation. Although the document representation of the BIR model is rather poor, the amount of feedback data available in predictive retrieval prohibits any refinement of the document representation.

A different approach has been taken in the formulation of the retrieval-with-probabilistic-indexing (RPI) model presented in Ref. 16. This model assumes that a more detailed document representation than in the case of the BIR model has been used for estimating probabilistic index term weights of the form $P(C | t_i, d_m)$, where C denotes the event of correctness. The decision whether the assignment of t_i to d_m is correct or not can be specified in various ways, e.g. by comparison with manual indexing or by regarding retrieval results as in the case of the BII model. Like the non-binary model mentioned before, the RPI model also is a generalisation of the BIR model. However, since weighted document indexing is regarded as document description in the RPI model, the number of parameters remains the same as in the BIR model, only the definition of these parameters is changed appropriately. For this reason, there are no additional parameter estimation problems in comparison to the BIR model, but a more detailed document representation can be considered. This goal is achieved by shifting the task of mapping document representations on to indexing weights over to an appropriate indexing model.

A similar model that integrates probabilistic indexing with the BIR model has been proposed as the 'unified model' in Ref. 3; however, this model suffered from incompatible independence assumptions. In Ref. 27, a generalisation of this model with modified independence assumptions is presented.

As mentioned before, description-oriented approaches also can be applied for developing retrieval functions that are able to consider more detailed document representations. Here query–document pairs are mapped on to a feature vector $\boldsymbol{x}(q_k, d_m)$. In principle, there is no restriction on the structure of the representations of queries and documents, only the feature vector has to be defined appropriately. In order to develop the retrieval function $\rho(\boldsymbol{x})$ that yields estimates of the probability $P(R | \boldsymbol{x}(q_k, d_m))$, a learning sample of query–document pairs (according to the third learning strategy) is used (see Ref. 8 for a detailed description of this approach). For most applications, it may be more appropriate to consider improved document representations already in the indexing process, so the RPI model can be used for retrieval instead of a description-oriented function. However, when more complex query structures are to be used, retrieval functions derived with the description-oriented approach may be feasible. In addition, a major advantage of retrieval functions of this kind is that they yield estimates of the probability of relevances, whereas the estimation of this probability is rather difficult with most other models.

4. IR AS UNCERTAIN INFERENCE

Although the relevance models described in the previous sections have been rather successful in the past, there are three major shortcomings of this approach, as follows.

- The concept of relevance can be interpreted in different ways. One can either regard relevance of a document w.r.t. a query or information need, in which cases the user who submitted the query gives the relevance judgement; this approach has been taken so far in this paper. Alternatively, relevance can be defined w.r.t. the query formulation, assuming that an objective judgement (e.g. given by specialists of the subject field) can be made. Of course, the latter approach would be more desirable in order to collect 'objective' knowledge within an IR system.
- The relevance models are strongly collection-dependent, that is, all the parameters of a model are only valid for the current collection. When a new collection is set up, the 'knowledge' from other collections cannot be transferred.
- Relevance models are restricted to rather simple forms of inference. In the models presented here, only the relationships between terms and queries are considered. It would be desirable to include information from other knowledge sources (e.g. from a thesaurus) in an IR model. With description-oriented approaches, this problem can be partly solved (see e.g. Ref. 8), but there is a need for a general model dealing with this issue.

4.1 Rijsbergen's model

In Ref. 28 a new paradigm for probabilistic IR is introduced: IR is interpreted as uncertain inference. This approach can be regarded as a generalisation of deductive databases, where queries and database contents are treated as logical formulas. Then, for answering a query, the query has to be proved from the formulas stored in the database.[29] For document retrieval, this means that a document d_m is an answer to a query q_k if the query can be proved from the document, that is, if the logical formula $q_k \leftarrow d_m$ can be shown to be true. In order to prove this formula, additional knowledge not explicitly contained in the document can be used. For example, if d_1 is about 'squares', and q_1 asks for documents about 'rectangles', the inference process can use the formula 'rectangle' ← 'squares' in order to prove $q_1 \leftarrow d_1$. For IR, however, the approach from deductive databases is not sufficient, for two reasons: (1) whereas in databases all statements are assumed to be true at the same time, a document collection may contain documents that contradict each other; (2) in order to cope with the intrinsic uncertainty of IR, first-order predicate logic must be replaced by a logic that incorporates uncertainty. For the first problem, Rijsbergen identifies each document with a possible world W, that is, a set of propositions with associated truth values. Let τ denote a truth function, then $\tau(W, x)$ denotes the truth of the proposition x in the world W, where $\tau(W, x) = 1$ if x is true at W and $\tau(W, x) = 0$ if x is false at W.

In order to cope with uncertainty, a logic for

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

probabilistic inference is introduced. In this logic, conditionals of the form $y \rightarrow x$ can be uncertain. For quantifying the uncertainty, the probability $P(y \rightarrow x)$ has to be estimated in some way. As described in Ref. 30, this probability can be computed via imaging. Let $\sigma(W, y)$ denote the world most similar to W where y is true. Then $y \rightarrow x$ is true at W if and only if x is true at $\sigma(W, y)$.

For estimating $P(y \rightarrow x)$ (independent of a specific world), all possible worlds must be regarded. Each world W has a probability $P(W)$, so that they sum to unity over all possible worlds. Then $P(y \rightarrow x)$ can be computed in the following way:

$$\begin{aligned} P(y \rightarrow x) &= \sum_W P(W)\,\tau(W, y \rightarrow x) \\ &= \sum_W P(W)\,\tau(\sigma(W,y), y \rightarrow x) \\ &= \sum_W P(W)\,\tau(\sigma(W,y), x). \end{aligned} \tag{9}$$

So we have to sum over all possible worlds, look for the closest world where y is true, and add the truth of x for this world. This formula can be illustrated by an example shown in Table 3. If we assume that $P(W_i) = 0.1$ for $i = 1, \ldots, 10$, then we get $P(y \rightarrow x) = 0.6$.

Table 3. Imaging example.

W_i	$\tau(y)$	$\tau(x)$	$\sigma(W_i, y)$	$\tau(\sigma(W_i, y)\,x)$
1	0	1	2	0
2	1	0	2	0
3	0	0	6	1
4	0	1	6	1
5	0	0	6	1
6	1	1	6	1
7	0	0	8	1
8	1	1	8	1
9	0	0	10	0
10	1	0	10	0

In this framework, the concept of relevance does not feature. The most obvious way for mapping the outcome of the uncertain inference process on to the probability of relevance is via another conditional probability:

$$\begin{aligned} P(R) = P(R \mid q_k \leftarrow d_m)\,P(q_k \leftarrow d_m) + P(R \mid \neg(q_k \leftarrow d_m)) \\ P(\neg(q_k \leftarrow d_m)). \end{aligned} \tag{10}$$

This leaves us with the problem of estimating the probabilities $P(R \mid q_k \leftarrow d_m)$ and $P(R \mid \neg(q_k \leftarrow d_m))$. So far, there is no obvious method for deriving the values of these parameters. On the other hand, according to formula (10), $P(R)$ is a monotonous function of $P(q_k \leftarrow d_m)$, thus only the value of the latter probability is required for a ranking of documents w.r.t. a query.

4.2 Inference networks

When IR is regarded as uncertain inference as described above, the structure of inference from documents to queries become more complex, as in the case of the relevance models. In general, one gets an inference network. As a probabilistic formalism for inference networks with uncertainty, Bayesian inference networks have been described in Ref. 12. Turtle and Croft[31] applied this formalism to document retrieval. An example inference network is shown in Fig. 4. Here each node representing either a document, a query or a concept can take on the value true or false. In contrast to the models discussed so far in this paper, we assume that there are two kinds of concepts, namely document concepts t_i and query concepts r_i. The directed arcs of this network indicate probabilistic dependence of nodes. The probability of a node being true depends only on the values of its parents. This relationship must be specified as a function within the node. In order to estimate the probability of $P(d \rightarrow q)$, the document node is set to 'true' and then the probabilities of the depending nodes are computed until the value of $P(q = \text{true})$ is derived.

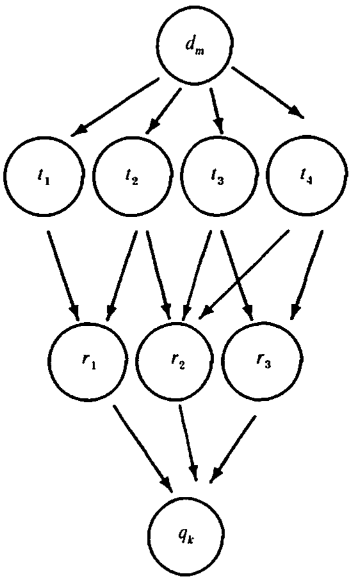

Figure 4. Example inference network.

Depending on the combining function of a node, an inference network may contain different types of nodes, e.g. for Boolean connectors as well as for probabilistic correlation as in the relevance models described before. As a simple example, assume that the representation concept r_1 is 'IR' which is defined as an OR-combination of the two document concepts (terms) t_1 = 'information retrieval' an t_2 = 'document retrieval'. Then the probability of r_1 being true can be computed by the function

$$P(r_1 = \text{true}) = 1 - (1 - P(t_1 = \text{true}))(1 - P(t_2 = \text{true})).$$

This approach has several advantages in comparison to the relevance models, as follows.

- As most probabilistic IR models can be mapped on to an inference network, this formalism gives a unifying representation for the different models (see e.g. Ref 31). In contrast to these models, the network approach does not require the derivation of a closed probabilistic formula, so more complex interdependences can be incorporated.
- The network approach allows for the combination of multiple sources of evidence. For example, information about the similarity of documents can be considered, as well as knowledge from external sources like a thesaurus.
- Different query formulations and types of query formulation can be combined in order to answer a single query. For example, a Boolean and a probabilistic formulation can be used in parallel,

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

where the network combines the results of both formulations.

5. GENERAL PROBLEMS

In this section we discuss some problems that are general to any probabilistic IR model, namely parameter estimation, query expansion and the representation of documents and queries.

5.1 Parameter estimation

Any probabilistic IR model requires the estimation of certain parameters before it can be applied. A survey over estimation schemes in IR is given in Ref. 32. Here we want to describe briefly the two major solutions to this problem.

The general situation is as follows: in a collection of documents, each document may have several features e_i. For a fixed set of n feature pairs, we are seeking for estimates of $P(e_i | e_j)$, the probability that a random document has feature e_i, given that it has feature e_j. In a random sample of g objects, we observe f objects with feature e_j of which h objects also have the feature e_i. In the case of the BIR model, the features e_j are either relevance or non-relevance w.r.t. the current query, and the features e_i denote the presence of the terms t_i.

Now the problem is to derive an estimate $p(e_i | e_j, (h, f, g))$ for $P(e_i | e_j)$, given the parameter triple (h, f, g). The most simple estimation method uses the maximum likelihood estimate, which yields $p(e_i | e_j, (h, f, g)) = h/f$. Besides the problem with the quotient 0/0, this estimate also bears a bias (see the experimental results in Ref. 32).

Bayesian estimates in combination with a beta prior have been used by most researchers in probabilistic IR. For the parameters a and b of the beta distribution and a quadratic loss function, one gets the estimate

$$p_{\text{beta}} = \frac{h+a}{f+a+b}.$$

For the choice of the parameters a and b, $a = b = 0.5$ has been used in most experiments (see also Ref. 33 for experiments with different values for these parameters). This kind of estimate also can be applied when no relevance feedback data is available, that is, $f = h = 0$. In the case of the BIR model, experiments without feedback data as described in Ref. 34 gave good results.

However, experimental results described in Ref. 32 have shown that the assumption of a beta prior may be theoretically inadequate. Instead, an optimum estimate based on empirical distributions is derived. Assume that $E(h, f, g)$ denotes the expectation of the number of those of the n feature pairs for which the parameters (h, f, g) were observed. These expectations can be taken from the frequency statistics of a large number of feature pairs (e_i, e_j). Then the optimum estimate is computed according to the formula

$$p_{\text{opt}} \approx \frac{(h+1)\,E(h+1, f+1, g)}{(h+1)\,E(h+1, f+1, g) + (\mathrm{f}+1-h)\,E(h, f+1, g)}.$$

Experimental comparisons of this optimum estimate with Bayesian estimates showed almost no difference in terms of retrieval quality, whereas maximum likelihood estimates gave significantly worse results.

5.2 Query expansion

Closely related to the problem of parameter estimation is the question: which terms should be included in the query formulation? In derivation of the BIR model, we have assumed that $p_{ik} = q_{ik}$ for all terms not occurring in the query. Of course, there will be a number of additional terms for which this assumption does not hold, so they should be included in the query, too. If the query formulation is to be expanded by additional terms, there are two problems that are to be solved, namely (1) how are these terms selected, and (2) how are the parameters c_{ik} estimated for these terms?

For the selection task three different strategies have been proposed, as follows.

- Dependent terms: here terms that are dependent on the query terms are selected. For this purpose, the similarity between all terms of the document collection has to be computed first.[35]
- Feedback terms: from the documents that have been judged by the user, the most significant terms (according to a measure that considers the distribution of a term within relevant and nonrelevant documents) are added to the query formulation.[36]
- Interactive selection: by means of one of the methods mentioned before, a list of candidate terms is computed and presented to the user, who makes the final decision on which terms are to be included in the query.[37]

With respect to the parameter estimation task, experimental results have indicated that the probabilistic parameters for the additional query terms should be estimated in a slightly different way than for the initial query terms, e.g. by choosing different parameters a and b in the beta estimate (that is, a different prior distribution is assumed for the new terms).[36,38]

The experimental results available so far indicate that the dependent-terms method does not lead to better retrieval results,[35] whereas clear improvements are reported in Refs 36 and 39 for the feedback-terms method.

5.3 Representation of documents and queries

So far in this paper we have assumed that a query is a set of terms, which in turn are words (with the exception of the Bayesian network approach, where a query also may contain Boolean connectors). But as more advanced text analysis methods are available, there is a growing need for models that can be combined with refined representation formalisms. Several authors have investigated the additional use of phrases as query terms.[40–42] The results from this experimental work do not give a clear indication whether retrieval quality can be improved with phrases as query terms. However, three problems should be considered when interpreting these results.

(1) Phrases are a different kind of terms in comparison to words. For this reason, the application of the standard weighting schemes developed for words may not be appropriate.

(2) When phrases as well as their components are used as query terms, these terms are highly dependent, so a

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

dependence model should be used, as for example in Ref. 43.

(3) The document collections used for experiments may be too small to show any benefit from phrases. Other experiments with larger collections have successfully used phrases in addition to words.[18,44]

A dependence model for phrases is not sufficient, since this approach only regards the occurrence of the phrase components in a document, without considering the syntactical structure of the phrase as it occurs in the document. So the certainty of identification also should be considered (e.g. whether the components occur adjacent or only within the same paragraph). This can be achieved via application of probabilistic indexing methods (e.g. with the BII model in combination with a description-oriented approach); furthermore, with regard to the first point above, indexing methods compute correct weights for different kinds of terms.

In contrast to the approaches discussed so far that are based on free text search, there is also work on automatic indexing with a controlled vocabulary (descriptors). In this case an indexing dictionary is required that contains pairs (with associated weights) of text terms and descriptors. For the problem of computing an indexing weight for a descriptor with indications from different terms within a document, either model-oriented[45] or description-oriented approaches[18,44] can be applied. Although many researchers are convinced that a controlled vocabulary offers no advantage over a free vocabulary, there is in fact little substantial experimental evidence supporting this position.[46]

As an example for an advanced text representation method, in Ref. 47 a semantic network representation is used for medical documents and queries (in conjunction with a fuzzy retrieval function). For probabilistic IR, this kind of representation is a challenge. Theoretically, the uncertain inference approach as developed by van Rijsbergen could be applied here – although the problem of parameter estimation has not been finally solved. Another possibility is the description-oriented approach described in Ref. 8, which, however, requires fairly large learning samples for application.

6. CONCLUSION AND OUTLOOK

In this paper the major concepts of probabilistic IR have been described. Following this goal, we have only occasionally referred to experimental results; of course, experiments are necessary in order to evaluate and compare different models. For future research it seems to be important to use test collections that are more representative for the intended applications, e.g. collections with 10^5–10^6 documents with regard to large online databases.

As new possible applications for IR methods arise, the scope of the field also has to be revised. In the past, collections with short documents (i.e. abstracts) have been investigated almost exclusively. Nowadays, fulltext document databases are set up for many applications; so far, no experimental results are available for the applicability of probabilistic methods (for example, the BIR model seems to be inappropriate in this case since it does not distinguish between terms that occur only once in a text and others that represent important concepts of a document). With multimedia documents there is the problem of representation for the non-textual parts: should they be represented by a set of keywords, or are structured descriptions required (as in Ref. 48)?

In the field of database research, there is also growing interest in methods for coping with imprecision in databases.[49,50] As new databases for technical, scientific and office applications are set up, this issue becomes of increasing importance. A first probabilistic model that can handle both vague queries and imprecise data has been presented in Ref. 51. Furthermore, the integration of text and fact retrieval will be a major issue.[52]

Finally, it should be mentioned that the models discussed here scarcely take into account the special requirements of interactive retrieval. Even the feedback methods are more or less related to batch retrieval, where feedback from the first retrieval run is used in order to improve the quality of the second run (an exception must be made for Bookstein's paper,[53] where iterative feedback methods are discussed). Since an interactive system allows a larger variety of interactions than just query formulation and relevance feedback (see e.g. Ref. 54), these interactions should be incorporated in a model for interactive probabilistic retrieval. Especially, the role of probabilistic parameters in the interaction process should be investigated: how should probabilistic weights (if at all) be presented to the user, and should there be a possibility for a user to specify probabilistic weights? In order to answer these questions, more experimental research is necessary. A major impediment for this kind of research is the fact that experiments with interactive systems and real users require a much bigger effort than testing ranking procedures in a batch environment.

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

REFERENCES

1. C. van Rijsbergen and K. Sparck Jones, A test for the separation of relevant and non-relevant documents in experimental retrieval colletions. *Journal of Documentation* **29**, 251–257 (1973).
2. W. Cooper, Some inconsistencies and misnomers in probabilistic IR. In *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, edited A. Bookstein, Y. Chiaramella, G. Salton, V. Raghavan, pp. 57–61. ACM, New York (1991).
3. S. Robertson, M. Maron and W. Cooper, Probability of relevance: a unification of two competing models for document retrieval. *Information Technology: Research and Development* **1**, 1–21 (1982).
4. N. Fuhr and G. Knorz, Retrieval test evaluation of a rule based automatic indexing (AIR/PHYS). In *Research and Development in Information Retrieval*, edited C. Van Rijsbergen, pp. 391–408. Cambridge University Press, Cambridge (1984).
5. S. Robertson, The probability ranking principle in IR. *Journal of Documentation* **33**, 294–304 (1977).
6. K. Stirling. The effect of document ranking on retrieval system performance: a search for an optimal ranking rule. In *Proceedings of the American Society for Information Science* **12**, 105–106 (1975).
7. A. Bookstein, Outline of a general probabilistic retrieval model. *Journal of Documentation* **39** (2), 63–72 (1983).
8. N. Fuhr, Optimum polynomial retrieval functions based

on the probability ranking principle. *ACM Transactions on Information Systems* **7** (3), 183–204 (1989).

9. A. Bookstein, Probability and fuzzy-set applications to information retrieval. *Annual Review of Information Science and Technology* **20**, 117–151 (1985).
10. C. van Rijsbergen, *Information Retrieval*, 2nd edn, chapter 6, pp. 111–143. Butterworths, London (1979).
11. C. Yu, C. Buckley, K. Lam and G. Salton, A generalized term dependence model in information retrieval. *Information Technology: Research and Development* **2**, 129–154 (1983).
12. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, San Mateo, CAL (1988).
13. C. van Rijsbergen, A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation* **33**, 106–119 (1977).
14. N. Fuhr and C. Buckley, A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems* **9** (3), 223–248 (1991).
15. M. Maron and J. Kuhns, On relevance, probabilistic indexing, and information retrieval. *Journal of the ACM* **7**, 216–244 (1960).
16. N. Fuhr, Models for retrieval with probabilistic indexing. *Information Processing and Management* **25** (1), 55–72 (1989).
17. K. Kwok, Experiments with a component theory of probabilistic information retrieval based on single terms as document components. *ACM Transactions on Information Systems* **8**, 363–386 (1990).
18. P. Biebricher, N. Fuhr, G. Knorz, G. Lustig and M. Schwantner, The automatic indexing system AIR/PHYS – from research to application. In *11th International Conference on Research and Development in Information Retrieval*, edited Y. Chiaramella, pp. 333–342. Presses Universitaires de Grenoble, Grenoble, France (1988).
19. G. Knorz, *Automatisches Indexierne als Erkennen abstrakter Objekte*. Niemeyer, Tübingen (1983).
20. A. Bookstein and D. Swanson, Probabilistic models for automatic indexing. *Journal of the American Society for Information Science* **25**, 312–318 (1974).
21. S. Harter, A probabilistic approach to automatic keyword indexing, part I. On the distribution of speciality words in a technical literature. *Journal of the American Society for Information Science* **26**, 197–206 (1975).
22. S. Harter, A probabilistic approach to automatic keyword indexing. Part II. An algorithm for probabilistic indexing. *Journal of the American Society for Information Science* **26**, 280–289 (1975).
23. P. Srinivasan, A comparison of two-poisson, inverse document frequency and discrimination value models of document representation. *Information Processing and Management* **26** (2), 269–278 (1990).
24. E. Margulis, *N-Poisson Document Modelling Revisited*. Technical Report 166, ETH Zürich, Dèpartment Informatique, Institut für Informationssysteme (1991).
25. C. Yu, W. Meng and S. Park, A framework for effective retrieval. *ACM Transactions on Database Systems* **14** (2), 147–167 (1989).
26. C. Yu and H. Mizuno, Two learning schemes in information retrieval. In *11th International Conference on Research & Development in Information Retrieval*, edited Y. Chiaramella, pp. 201–218. Presses Universitaires de Grenoble, Grenoble, France (1988).
27. S. Wong and Y. Yao, A generalized binary probabilistic independence model. *Journal of the American Society for Information Science* **41** (5), 324–329 (1990).
28. C. J. van Rijsbergen, A non-classical logic for information retrieval. *The Computer Journal* **29** (6) (1986).
29. R. Reiter, Towards a logical reconstruction of relational database theory. In *On Conceptual Modelling*, edited M. Brodie, J. Mylopoulos and J. Schmidt, pp. 191–233. Springer, New York (1984).
30. C. van Rijsbergen, Towards an information logic. In *Proceedings of the Twelfth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, edited N. Belkin and C. van Rijsbergen, pp. 77–86. ACM, New York (1989).
31. H. Turtle and W. B. Croft, Inference networks for document retrieval. In *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, edited J.-L. Vidick, pp. 1–24. ACM, New York (1990).
32. N. Fuhr, H. Hüther, Optimum probability estimation from empirical distributions. *Information Processing and Management* **25** (5), 493–507 (1989).
33. R. Losee, Parameter estimation for probabilistic document-retrieval models. *Journal of the American Society for Information Science* **39** (1), 8–16 (1988).
34. W. Croft and D. Harper, Using probabilistic models of document retrieval without relevance information. *Journal of Documentation* **35**, 285–295 (1979).
35. C. van Rijsbergen, D. Harper and M. F. Porter, The selection of good search terms. *Information Processing and Management* **17**, 77–91 (1981).
36. G. Salton and C. Buckley, Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science* **41** (4), 288–297 (1990).
37. D. Harman, Towards interactive query expansion. In *11th International Conference on Research & Development in Information Retrieval*, edited Y. Chiaramella, pp. 321–331. Presses Universitaires de Grenoble, Grenoble, France (1988).
38. S. Robertson, On relevance weight estimation and query expansion. *Journal of Documentation* **42**, 182–188 (1986).
39. K. Kwok, Query modification and expansion in a network with adaptive architecture. In *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, edited A. Bookstein, Y. Chiaramella, G. Salton and V. Raghavan, pp. 192–201. ACM, New York (1991).
40. W. B. Croft, Boolean queries and term dependencies in probabilistic retrieval models. *Journal of the American Society for Information Science* **37** (2), 71–77 (1986).
41. J. Fagan, The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *Journal of the American Society for Information Sciences* **40** (2), 115–132 (1989).
42. T. Sembok, and C. van Rijsbergen, SILOL: A simple logical-linguistic document retrieval system. *Information Processing and Management* **26** (1), 111–134 (1990).
43. W. Croft, H. Turtle, D. Lewis, The use of phrases and structured queries in information retrieval. In *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, edited A. Bookstein, Y. Chiaramella, G. Salton, A. Raghavan, pp. 32–45. ACM, New York (1991).
44. N. Fuhr, S. Hartmann, G. Knorz, G. Lustig, M. Schwantner and K. Tzeras, AIR/X – a rule-based multistage indexing system for large subject fields. *Proceedings of the RIAO'91, Barcelona, Spain, 2–5 April, 1991*, pp. 606–623 (1991).
45. S. Robertson and P. Harding, Probabilistic automatic indexing by learning from human indexers. *Journal of Documentation* **40**, 264–270 (1984).
46. G. Salton, Another look at automatic text-retrieval systems. *Communications of the ACM* **29** (7), 648–656 (1986).
47. Y. Chiaramella and J. Nie, A retrieval model based on an extended modal logic and its application to the rime experimental approach. In: *Proceedings of the 13th International Conference on Research and Development in*

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

Information Retrieval, edited J.-L. Vidick, pp. 25–44. ACM, New York (1990).

48. F. Rabitti and P. Savino, Image query processing based on multi-level signatures. In: *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, edited A. Bookstein, Y. Chairamella, G. Salton and V. Raghavan, pp. 305–314. ACM, New York (1991).
49. IEEE. *IEEE Data Engineering* **12** (2). Special Issue on Imprecision in Databases (1989).
50. A. Motro, Accommodating imprecision in database systems: issues and solutions. *IEEE Data Engineering Bulletin* **13** (4), 29–42 (1990).
51. N. Fuhr, A probabilistic framework for vague queries and imprecise information in databases. In *Proceedings of the 16th International Conference on Very Large Databases*, edited D. McLeod, R. Sacks-Davis and H. Schek, pp. 696–707. Morgan Kaufman, Los Altos, CA (1990).
52. F. Rabitti and P. Savino, Retrieval of multimedia documents by imprecise query specification. In *Advances in Database Technology – EDBT '90*, edited F. Bancilhon, C. Thanos and D. Tsichritzis, pp. 203–218. Springer, Berlin et al. (1990).
53. A. Bookstein, Information retrieval: a sequential learning process. *Journal of the American Society for Information Science* **34**, 331–342 (1983).
54. W. B. Croft and R. H. Thompson, I3R: A new approach to the design of document retrieval systems. *Journal of the American Society for Information Science* **38** (6), 389–404 (1987).

Downloaded from https://academic.oup.com/comjnl/article-abstract/35/3/243/525633 by guest on 13 December 2018

Book Review

Nell Dale and Chip Weems
Introduction to Pascal and Structured Design, 3rd edition
D. C. Heath and Co., Lexington, 1991
925 pp. 0-669-20238-X.
Nell Dale and Susan C. Lilly
Pascal plus Data Structures, Algorithms and Advanced Programming, 3rd edition
D. C. Heath and Co., Lexington, 1991
850 pp. 0-669-24830-4.

Any book that comes out in a third edition in the cut-throat world of first-year Pascal texts deserves close scrutiny. These two books are a sample from a suite of six books, two laboratory courses and a video course that Nell Dale and colleagues have published through D. C. Heath since 1983. They have evidently refined their winning formula to the point where it is a 'success at over 1250 schools' (quoted from the publisher's advertising). So what is it that the customers like? Why are these books so successful?

There seem to be two factors involved: pedagogy and quality. Both books cover their topics absolutely thoroughly, explaining each new feature as it arises, using interesting examples, both classic (binary search) and novel (absenteeism pattern). Some of the examples progress through the book, becoming more sophisticated as the student's expertise increases, and there are several larger case studies. The order of topics has been carefully chosen, and the authors have adapted the material through each revision so as to reflect changing trends. The teaching and learning aids are impressive: the introductory book concludes each chapter with case studies, advice on testing and debugging, and then four levels of exercises for the students, from a quick check quiz to genuine programming problems. Beginners will find these very reassuring. The advanced book ends each chapter with an extensive set of exercises, and has a separate section at the end of the book with nearly 30 programming assignments. These have obviously been class tested and the attention to detail makes it possible for the lecturer to set them for a class as-is.

Through the editions, the authors have had the time to add in additional material which may be regarded as peripheral, but which, in my opinion, greatly enhances the value of these books. The introductory text is particularly good in this regard, and includes pen sketches of famous computer scientists, snippets of related theory, advice on style and guidelines for program design. The programs are all of a consistently high standard, using plentiful comments, good type definitions, and procedures with parameters to the full. The advanced book includes a diskette containing all the programs, which is a very good sign that the programs have been well tested. Both books are produced in two colours, with cartoons and many graphic illustrations. Quality indeed!

The introductory book has 17 chapters, which take a genuine beginner through the programming process, design methodology and problem solving, to simple Pascal with control structures, procedures and parameters, on to functions, data types and recursion. Purists can note that looping is taught with **while** statements and that **repeat** and **for** only appear later on in the data types section. Subranges are emphasised from chapter 10 onwards, and I was glad to see that they come before arrays. Unlike many books that dive into array handling at the start, Nell and Weems take this conceptually difficult hurdle slowly. Chapters 11, 12 and 13 introduce array processing, patterns of array access, lists, strings tables. Chapter 15 covers files and pointers and chapter 16 provides a gentle introduction to the advanced book on data structures.

The only problem with this book is that it covers standard Pascal and only standard Pascal. Thus many common operations (reading in a string) are long-winded, some (e.g. assigning a file name) have to be side-stepped, and the important programming issues of screen and graphics handling, separate compilation and objects, are simply ignored. There is a companion volume which is a Turbo Pascal version, and one hopes that these aspects are fully addressed there.

Because the advanced book follows the same easily accessible style and high-quality layout of the introductory text (complete with jokes), I was tempted to regard it as not a serious contender in the Data Structures and Algorithms text stakes. Closer scrutiny proved me wrong. Dale and Lilly have managed to retain a high level of rigour in the presentation of abstract data types while still making each of them truly usable in practice. Stacks, queues, lists and binary trees are covered in detail, and the efficiency of each of the algorithms applied to them is formally discussed. Sorting and searching (including hashing) receive similar treatment. However, the book stops short of the next level of ADTs – bags, sets, B-Trees, directed graphs. Turbo Pascal is made use of in places where the authors were really desperate, and there is also a Turbo Pascal version of the book (but how 'Turbo' it is, I do not know).

So, how would these books fit into first courses outside the US? The introductory book is too basic and too slow for a typical first-semester course, where the majority of students will have programmed at school and may even know Pascal. The insight into programming is fine, but is dispersed throughout the text, and overall the book is just too big for the subject it covers. On the other hand, the pedagogical aids are first class, and I would certainly advise any student who is having difficulty with a Pascal course to get this book as back-up. The advanced book is also hefty, and although thorough, does not go as far into the exciting aspects of algorithms as the more formal and classic texts do. On the other hand, the examples and assignments are first rate, and I am very glad to have this book on my shelf.

These are both excellent books in all respects except coverage, and I would certainly recommend that any teachers of first or second year should inspect copies and judge whether the coverage is sufficient for the courses under consideration. If it is, then both class and lecturer will be joining the other 1250 satisfied schools, and are on to a winner.

Judy Bishop
Pretoria