

# AIR/X — a Rule-Based Multistage Indexing System for Large Subject Fields

Norbert Fuhr, Stephan Hartmann, Gerhard Lustig,  
Michael Schwantner, Konstadinos Tzeras  
Technische Hochschule Darmstadt, Fachbereich Informatik

Gerhard Knorz  
Fachhochschule Darmstadt, Fachbereich Information und Dokumentation  
W-6100 Darmstadt, Germany

## Abstract

AIR/X is a rule-based system for indexing with terms (descriptors) from a prescribed vocabulary. For this task, an indexing dictionary with rules for mapping terms from the text onto descriptors is required, which can be derived automatically from a set of manually indexed documents. Based on the Darmstadt Indexing Approach, the indexing task is divided into a description step and a decision step. First, terms (single words or phrases) are identified in the document text. With term-descriptor rules from the dictionary, descriptor indications are formed. The set of all indications from a document leading to the same descriptor is called a relevance description. A probabilistic classification procedure computes indexing weights for each relevance description. Since the whole system is rule-based, it can be adapted to different subject fields by appropriate modifications of the rule bases. A major application of AIR/X is the AIR/PHYS system developed for a large physics database. This application is described in more detail along with experimental results.

## 1 Introduction

The AIR/X system described in this paper performs an automatic indexing with index terms (called descriptors here) from a prescribed vocabulary. The texts to be indexed are abstracts written in English.

The indexing process consists of several stages, with specific rule bases involved in each stage. In order to cope with large subject fields, appropriate rule bases have to be developed. The major part of these application-specific rule bases can be derived automatically from the thesaurus and from manually indexed documents of the subject field.

An important application of the AIR/X system is the AIR/PHYS system developed for the physics database PHYS of the Fachinformationszentrum Karlsruhe/Germany. In 1983, an experimental prototype was evaluated in the AIR retrieval test by comparing the results for manual and automatic indexing (see [Fuhr & Knorz 84]). Based on the results of this test, the Fachinformationszentrum Karlsruhe decided to apply automatic indexing methods. For this application, the special system AIR/PHYS was developed, and the application started in 1985 (see [Biebricher et al. 88]). The Fachinformationszentrum Karlsruhe is one of Europe's biggest information services and runs a large number of data bases from the technical-scientific area as part of the world-wide Scientific and Technical Information Network (STN). The data base PHYS comprises more than one million documents, and the annual increment amounts to about 125 000 documents.

In this paper, we describe the current state of the AIR/X system. This system is more flexible (with respect to modifications of the rule bases) than the AIR/PHYS system, and indexing decisions made by the system are more transparent. Special emphasis is given to the description of the multistage indexing process and the role of the different rule bases involved. In Section 2, we first give a brief introduction into the major concepts of the Darmstadt indexing approach (DIA) underlying the AIR/X system. Then we describe the multistage indexing process and the rule bases in section 3, followed by an overview over the AIR/X system in section 4. An indexing example is given in section 5. In section 6, our system is compared with other work in this area. Finally, we give an outlook on further development of the AIR/X system.

## 2 Basic concepts of the Darmstadt Indexing Approach

The Darmstadt Indexing Approach (DIA) is based on the following preconditions:

2. The system of descriptors is prescribed<sup>1</sup>.

This approach has been tested successfully in two different subject fields, namely physics and food science and technology [Lustig 82]. In this section, we describe the major concepts of the DIA which form the basis of the AIR/X system. For more details, see [Lustig 86] and [Fuhr 89a].

## 2.1 The indexing dictionary

For the task of mapping text content onto the set of descriptors given by the controlled vocabulary, the automatic indexing system needs a special dictionary containing term-descriptor rules for as many *terms* (i.e. single words and phrases) of the application field as possible. The lack of such dictionaries seems to have been the main obstacle for putting previous results of indexing experiments into practice. In order to overcome this difficulty, we have investigated different techniques of automatic dictionary construction. Association factors derived from texts alone did not produce satisfactory results. However, the association factor  $z(t, s)$  introduced into automatic indexing with encouraging results twenty years ago [Fangmeyer & Lustig 69, Fangmeyer & Lustig 70], has proved to be extremely useful. Its generation requires a large set  $M$  of documents indexed manually with controlled language terms. The association factor  $z(t, s)$  of a term-descriptor pair  $(t, s)$  is defined by

$$z(t, s) = \frac{h(t, s)}{f(t)}$$

where

$f(t)$  = number of documents of  $M$  containing the term  $t$  in the abstract text and

$h(t, s)$  = number of those among the  $f(t)$  documents to which the descriptor  $s$  is manually assigned.

The association factor  $z(t, s)$  is an estimate of the probability for the descriptor  $s$  to be assigned to a document if the abstract of this document contains the term  $t$ . Obviously,  $z(t, s)$  is not a purely semantic measure but it depends also on the significance of the occurrence of the text term  $t$  with respect to the descriptor  $s$ . It is this property which makes the association factor  $z(t, s)$  an important tool in automatic indexing.

In the indexing dictionary, the association factors of all term-descriptor pairs  $(t, s)$  for which  $z(t, s)$  and  $h(t, s)$  exceed certain cutoff values  $c_1$  and  $c_2$ , respectively, are stored as elements of the relation  $Z$  with the attributes  $t, s, z(t, s)$  and  $h(t, s)$ .

For the computation of association factors  $z(t, s)$  for phrases as terms, a phrase dictionary must be given. A part of this dictionary can be taken from thesauri or general machine-readable dictionaries. The major part of the phrases is extracted from texts by means of the so-called "Begrenzerverfahren" (delimiter method) ([Jaene & Seelbach 75], [Kienitz-Vollmer & Reichardt 86]). This method uses a set of delimiters consisting of about 300 stopwords and punctuation symbols. In the texts, sequences of two to four adjacent words enclosed by delimiters are selected. For example, in the sentence

"After discussing problems  
of natural language processing  
in general, 3 systems are described"

(delimiters are underlined), three word groups are selected:

- a) the non-syntactical word group "discussing problems"
- b) the correct and specific term phrase "natural language processing"
- c) the correct, but not specific phrase "3 systems"

With simple additional criteria, most of the bad cases like a) and c) can be rejected immediately. A final strong reduction happens in the different steps of the computation process for the association factors  $z(t, s)$ : many word groups turn out to be not involved in the relation  $Z$ .

## 2.2 The indexing approach

The most important concept of the DIA is the logical subdivision of the indexing process into a description step and a decision step. In the description step information about the relationship between a descriptor  $s$  and the document  $d$  is collected. This data forms the decision base for the second step, the estimation of the probability that the assignment of  $s$  to  $d$  would be correct. The concept of subdivision also includes the iteration of these steps, thus leading to a multistage indexing method (see next section).

The description step starts with the identification of terms (single words or noun phrases). As this task cannot be done perfectly, each term is identified in a certain form of occurrence (FOC)  $v$ , where different FOCs correspond to different levels of confidence.

---

<sup>1</sup> In [Fuhr & Buckley 90] it is shown how the concepts described in the following also can be applied for indexing with terms from an unrestricted vocabulary.

concept of FOC comprises two aspects:

- 1.) the certainty with which a term is identified (e.g. measured by the distance of the components of a noun phrase),
- 2.) the significance of a term with respect to the document (characterized e.g. by the within document-frequency of a term or by the part(s) of the document in which the term occurs).

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

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

The collection of all descriptor indications from a document  $d$  leading to the same descriptor  $s$  is called the *relevance description* (RD)  $x(s, d)$  of  $s$  with respect to  $d$ . The decision step uses the RD  $x = x(s, d)$  to estimate the probability  $P(C|x)$  that, given the RD  $x$ , the correspondending descriptor assignment would be correct. This estimation is done by the *indexing function*  $a(x)$ .

For the development of indexing functions as well as for the construction of the indexing dictionary, learning samples with correct descriptor assignments must be given. Within the DIA, no assumptions are made about the origin of the correctness decisions. For practical reasons, manually indexed documents are mostly used for this purpose. Some experiments described in [Fuhr 89a] with indexing functions derived directly from retrieval judgements showed no improvements over indexing functions based on manual indexing.

For the development of indexing functions, several probabilistic classification algorithms have been investigated. With the exception of the so-called Boolean approach, all these algorithms require the transformation of the RD  $x$  into a relevance description vector (RDV)  $\vec{x}$ . This transformation has to be defined heuristically.

- 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.
- 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  $a(\vec{x}) = \frac{\exp(\vec{b}^T \cdot \vec{x})}{1 + \exp(\vec{b}^T \cdot \vec{x})}$ , where  $\vec{b}$  is a coefficient vector that is estimated based on the maximum likelihood method [Pfeifer 90].
- Least square polynomials [Knorz 83] [Fuhr 89b] yield indexing functions of the form  $a(\vec{x}) = \vec{b}^T \cdot \vec{x}$  (in the linear case), where  $\vec{b}$  is a coefficient vector that minimizes the expectation of the squared error  $(P(C|\vec{x}) - \vec{b}^T \cdot \vec{x})^2$ . This type of indexing function is actually used in the AIR/PHYS system.
- In this paper, we will discuss probabilistic classification trees as indexing functions, which are based on the probabilistic version of the learning algorithm ID3 [Hart 85], [Quinlan 84]. So far, the experiments performed with this method [Faißt 90] showed an indexing quality that is slightly inferior to that of least square polynomials, but classification trees offer the advantage of more transparency of the indexing function.

The ID3 algorithm requires a description vector  $\vec{x}$  with discrete-valued elements. So elements with continuous values have to be discretized first (see e.g. [Wong & Chiu 87]). Now ID3 constructs a probabilistic classification tree using a top down approach: select an attribute (= a component of  $\vec{x}$ ), divide the training set into subsets characterized by the possible values of the attribute, and repeat the same produce recursively with each subset until no more attribute can be selected. Each leaf  $l$  of the final tree represents a class of RDVs  $\vec{x}$ , where each possible vector  $\vec{x}$  belongs to exactly one class  $l$ . Let  $l_j$  denote the class into which RDV  $\vec{x}$  falls, then the indexing function is defined as  $a(\vec{x}) = P(C|\vec{x} \in l_j)$ . This probability can be estimated from the training sample.

The crucial point of the ID3 algorithm is the selection criterion. Assume that a component  $x$  of  $\vec{x}$  has possible values  $v_0, \dots, v_\alpha$ . Now we regard the distribution of these values within RDVs leading to correct ( $C$ ) and incorrect ( $\bar{C}$ ) descriptor assignments. For the current subset of our training sample, let  $O_{ij}$  with  $i = 0, \dots, \alpha$  and  $j = C, \bar{C}$  denote the observed number of RDVs with attribute value  $v_i$  and correctness decision denoted by  $j$ . Correspondingly, let  $E_{ij}$  denote the expected number of these RDVs by assuming a random distribution of the attribute values within correct and incorrect RDVs. Following the proposal in [Hart 85], we regard as a measure for a good discriminator between correct and incorrect RDVs the  $\chi^2$ -value

$$\sum_{i,j} \frac{(E_{ij} - O_{ij})^2}{E_{ij}},$$

where the number of degrees of freedom is  $\alpha$ . So, among all components of  $\vec{x}$  that have not been selected yet, the component with the highest significant  $\chi^2$ -value is selected, provided that the corresponding probability is

only components with all values  $E_{ij} \geq 10$  are considered. Due to these additional criteria, in most cases only a subset of the components of  $\vec{x}$  is actually present in the final classification tree.

### 3 Rule-based multistage indexing

As mentioned in the previous section, the general concepts of the DIA also include the iteration of the sequence of description step and decision step, where we call each sequence an *indexing phase*. By using more than one indexing phase, where each phase consists of two or more indexing stages, the structure of the indexing task is refined. This way, the complexity of each indexing stage is reduced, which is important for system development and maintenance. On the other hand, since each indexing phase can specialize on different aspects of the relationship between documents and descriptors, the overall indexing quality can be improved (in comparison to that of a single indexing phase). In the development of the AIR/X system, three different approaches to a 2-phases indexing have been investigated:

- For the AIR retrieval test, an automatic classification of documents was performed first, followed by class-specific description and decision steps ([Fuhr & Knorz 84]).
- In [Knorz 83], specialized indexing functions for the task of binary indexing were investigated, where the binary indexing was derived from weighted indexing by application of a cutoff value. In the first indexing phase, a general indexing function for all RDs was used. For the second phase, only RDs with weights close to the cutoff value were considered, and a special indexing function for these RDs was developed. This calculation of indexing functions gave a better indexing quality than the first indexing function alone.
- The current version of the AIR/X system includes a second indexing phase that copes with interdependencies between descriptors. The major reason for this approach comes from application fields with descriptor-descriptor relations of a thesaurus and indexing rules (for human indexers) that refer to these thesaurus relations (see below).

In the following, we describe the different stages of the indexing process of the AIR/X system. We first give a brief survey over the whole indexing process, followed by a detailed description of the different stages.

Figure 1 shows the 5 indexing stages together with the rule bases involved in different stages. In the *term identification* stage, the text to be indexed is mapped onto a set of (*term*, *FOC*) pairs. For these terms, term-descriptor rules are retrieved from the indexing dictionary in the *RD1 construction* stage, and descriptor indications and RDs are formed. Based on these RDs, indexing weights are computed in the *1st decision* stage. The sequence of these three indexing stages forms the first indexing phase. Interdependencies between descriptors are considered in the second indexing phase with another RD construction stage and a decision stage. For all descriptors with an indexing weight exceeding a predefined cutoff value, new RDs are formed in the RD2 construction phase based on descriptor-descriptor rules and descriptor classification rules from the indexing dictionary. In the *2nd decision* stage, the final indexing weights are computed.

In each of the indexing stages, certain rule bases are involved. According to the origin of the rules, three different types of rules can be distinguished (these types are indicated in figure 1):

- P: Probabilistic rules are associated with probabilistic weights, which are derived from statistical analysis of learning samples with correct descriptor assignments. Examples for probabilistic rules are the term-descriptor rules of the relation *Z* and the indexing functions.
- H: Heuristic rules represent some kind of approximation to perfect rules which are not possible in an automatic indexing system (as well as in the whole field of information retrieval, see [Fuhr 89b]). Examples for heuristic rules are the stemming rules, the definition of the FOCs and the transformations of RDs into RDVs.
- T: Thesaurus rules are mainly derived from the thesaurus of the subject field. Examples for thesaurus rules are USE- or SEE-relations as term-descriptor rules and hierarchical relations as descriptor-descriptor rules. In addition, there may be term-descriptor or descriptor-descriptor rules developed by subject specialists especially for automatic indexing (e.g., the AIR/PHYS system uses a contrast relation with descriptor pairs that should not be assigned to the same document).

The separation between programs and rule bases in the AIR/X system offers two major advantages:

- For a single application, system maintenance is restricted to the update of the rule bases. With respect to this task, the rule-based approach makes the decision of the indexing system transparent and eases the identification of erroneous or missing rules. The major problem in system maintenance is the incompleteness of the indexing dictionary. Furthermore changes in the subject field that occur over time also require rule base updates. For these reasons, the development of an interactive system for rule base maintenance is a major goal of our current research (see last section).

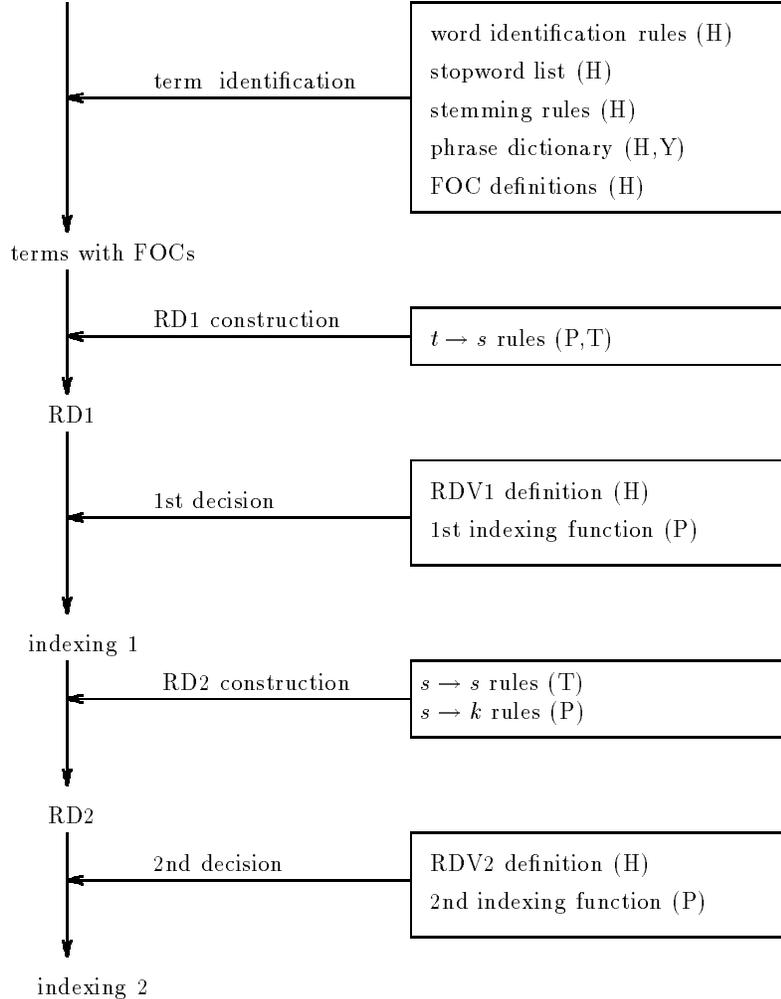


Figure 1: Rule bases in the multistage indexing process

- For new subject fields, only the rule bases have to be exchanged, whereas the programs remain the same. Of course, some applications require specific extensions of the system. For example, the AIR/PHYS system has a specific component in the term identification stage that analyzes physical and chemical formulas and maps them onto so-called formula identifiers, which are treated like terms in the following stages.

### 3.1 Term identification

In this stage, a document to be indexed is mapped onto a set of (*term*, *FOC*) pairs. First, the text is broken down into sentences, and the sentences are decomposed into words. This process is based on *word identification rules* (e.g. for the treatment of punctuation symbols, hyphens and the recognition of abbreviations). A *stopword list* specifies which words are ignored subsequently. By application of *stemming rules*, a dictionary independent algorithm [Kuhlen 77] reduces each word to its standard form (e.g. a verb form to the infinitive and a noun to its singular form). After application of the appropriate *FOC definitions*, the term identification for single words is finished. Phrases are identified by accessing the *phrase dictionary* with the words identified, and then the text is checked for the occurrence of the other components of a phrase from the dictionary. If all components are present, the *FOC definitions* for phrases are applied.

### 3.2 RD1 construction

For the terms identified in the document, *term-descriptor rules*  $t \rightarrow s$  together with additional data about  $t$  and  $s$  are retrieved from the indexing dictionary. For each term and each  $t \rightarrow s$  rule, a descriptor indication is generated. The collection of all descriptor indications from one document leading to the same descriptor forms a RD.

The recursive structure of the RD first has to be mapped onto the fixed-size structure of a RDV. This mapping is specified by the *RDV1 definition* (e.g. by applying aggregate functions to elements of indications with certain features). For indexing functions requiring discrete-valued attributes, the discretization parameters for attributes with continuous values are specified here, too. Then the *1st indexing function*  $a_1(\vec{x})$  can be applied to the RDV  $\vec{x}$ , thus yielding the indexing weights  $a_1(\vec{x}(s, d))$ . In the following, let  $S_1(d)$  denote the set of all descriptors assigned to document  $d$  with an indexing weight exceeding a predefined cutoff value.

### 3.4 RD2 construction

In the second indexing phase, interdependencies between descriptors are considered. Unlike other approaches that cope with index term dependencies on a statistical basis (see e.g. [Rijsbergen 77, Yu et al. 83, Wong et al. 87]), we regard two different kinds of descriptor dependencies here:

- For many document databases, there are indexing rules that restrict the possible combinations of descriptor assignments to a single document. In the case of the PHYS database, there is a rule that states that always the most specific descriptor (from the thesaurus hierarchy) should be assigned. So, if two descriptors are assigned in the first indexing phase which occur as a pair in the Broader-Term relation, then the more general descriptor should not be assigned (Similar rules are discussed in [Humphrey 87], see also section 6).
- By regarding the set of descriptor assignments from the first indexing phase as a whole, some descriptors that are inconsistent with other assignments can be detected, thus improving the indexing quality.

In order to consider these dependencies, there are two kinds of rules in the indexing dictionary:

- *Descriptor-descriptor rules*  $s_1 \rightarrow s_2$  specify either preferred or forbidden combinations of assignments. For constructing the relevance descriptions RD2 for a document  $d$ , all rules  $s_1 \rightarrow s_2$  with  $s_1, s_2 \in S_1(d)$  are retrieved from the dictionary. The corresponding elements of RD2 also refer to the weights  $a_1(s_1, d)$  and  $a_1(s_2, d)$ .
- If there is a document classification  $K = \{k_1, \dots, k_n\}$  for the subject field, then the distribution of the descriptors over these classes can be considered. The underlying assumption is that there is a correlation between the probability of correctness of the assignment of a descriptor and its distribution over the classes in comparison to that of the other descriptors  $s_i \in S_1(d)$ . For this purpose, probabilistic *descriptor classification rules*  $s \rightarrow k$  can be derived from a sample of manually indexed and classified documents (similar to the relation  $Z$  for term-descriptor pairs). By retrieving all rules  $s_i \rightarrow k_j$  with  $s_i \in S_1(d)$  from the dictionary, the distribution of a single descriptor  $s_i$  over  $K$  can be compared with that of all descriptors  $s_j \in S_1(d)$ , thus forming additional elements of RD2. Besides elements derived from these rules, RD2 may also contain some parameters describing the document and the set  $S_1(d)$  with the associated indexing weights.

### 3.5 2nd decision

The 2nd decision stage is analogous to the 1st decision stage: First the *RDV2 definition* specifies the mapping of RD2 onto the relevance description vector RDV2. Then the *2nd indexing function*  $a_2(\vec{x})$  can be applied, thus yielding the final indexing weights.

## 4 The AIR/X system

Figure 2 gives a survey over the whole AIR/X system with two indexing phases. The dictionary construction software is a collection of programs that derives dictionary data from the thesaurus and from manually indexed documents of the subject field. Most important among these programs, the system ZWERG computes the relation  $Z$ . All dictionary data is stored in the dictionary database. The system ARCHIBALD (which is built on top of a commercial DBMS) manages this database. Besides storage, update and retrieval of dictionary data, this system offers some functions for statistical analysis of the dictionary database. ARCHIBALD also extracts the indexing dictionaries for the two indexing phases from the database. The indexing dictionaries are implemented as index-sequential files, thus providing a more efficient access than a database system.

The five indexing stages are subdivided into two systems (mainly for historical reasons): DAISY comprises the term identification stage and the RD1 construction stage, whereas the final three stages are implemented in the UNIDARES system. The only modules not shown in this picture are programs for evaluation of indexing quality (that is, comparison of manual and automatic indexing of a set of documents) and for the development of indexing functions (based on learning samples of manually indexed documents).

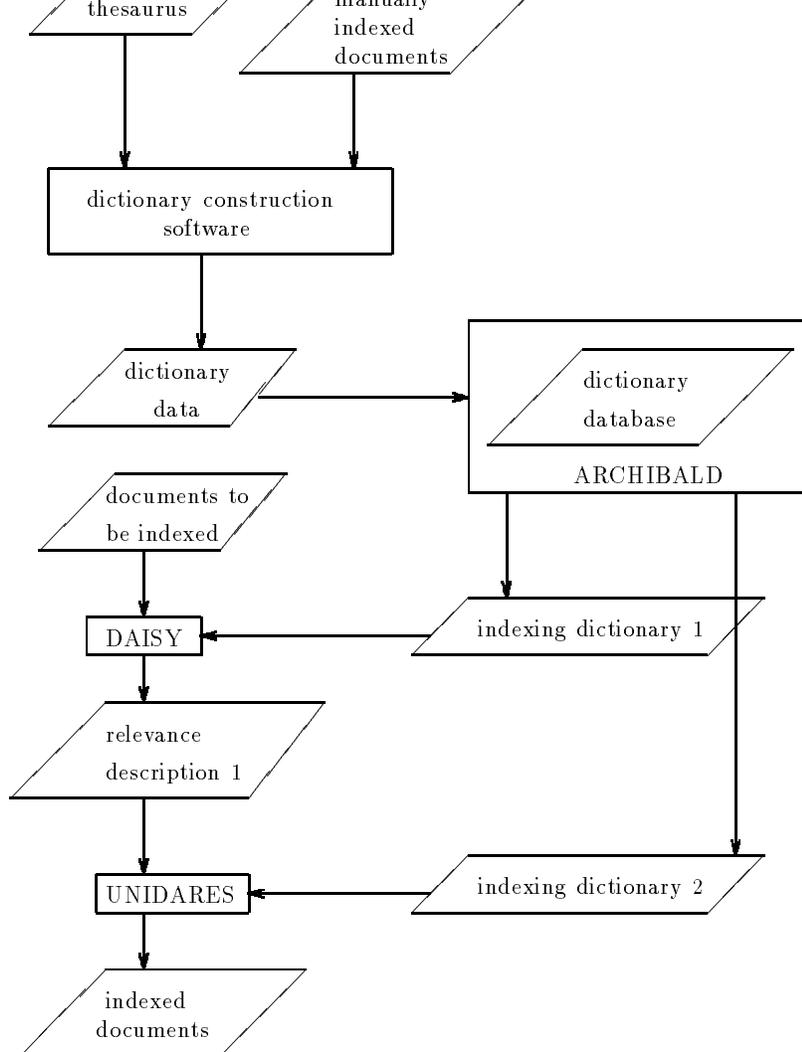


Figure 2: Survey of the AIR/X system

The whole AIR/X system is written in PL/1 and runs on Siemens mainframes under BS2000. Major parts of the dictionary construction software also run on IBM mainframes. The implementation of the DAISY and the UNIDARES system on IBM mainframes is currently under development.

## 5 Application of the AIR/X system

In this section, we show how the concepts and the system described in the previous sections are applied for the automatic indexing of the PHYS database of the Fachinformationszentrum Karlsruhe. First, we give a survey over the AIR/PHYS system developed for this application, followed by an example of indexing a document, and then we describe some experiments.

### 5.1 The AIR/PHYS system

In 1985, the PHYS database had a thesaurus with 22 683 descriptors. For this application, an indexing dictionary named PHYS/PILOT was developed. Since the documents of the PHYS database contain a significant number of physical or chemical formulas as well as numerical data, a special formula processing component (as part of the term identification stage) had to be developed for this application. Based on specific rules for the identification and transformation of formulas and numerical data, these elements are mapped onto so-called formula identifiers. Besides single words and phrases, these identifiers were used as terms for the computation of the relation  $Z$ , which was based on 392 000 manually indexed documents. With the conditions  $z(t, s) \geq 0.3$  and  $h(t, s) \geq 3$ ,

in PHYS/PILOT (unless mentioned otherwise, the figures given in the following are based on this dictionary.). They have been chosen according to their probable importance estimated by means of the following heuristic criteria: We only regarded the 620 617 pairs where the term is a phrase. From this set, first all pairs where  $10 \cdot z(t, s) + h(t, s) \leq 14$  were removed. Then we computed the set of descriptors for which no elements of the relation  $Z$  were remaining after this reduction. For any of these descriptors, all elements of  $Z$  previously excluded were added to the dictionary. This way, finally 170 697 pairs of  $Z$  where  $t$  is a phrase were included in PHYS/PILOT.

The other term-descriptor relations of PHYS/PILOT are:

- the USE relation of the PHYS thesaurus,
- the IDENTITY relation connecting each of the 22 683 descriptors with itself,
- the FORMULA IDENTITY relation mapping identifiers derived from formulas onto the corresponding descriptors.

For the second indexing phase, the dictionary contains the following descriptor-descriptor relations:

- the BROADER-TERM relation of the thesaurus,
- a FORMAL INCLUSION restricted to descriptor-descriptor pairs  $(s_1, s_2)$ . This relation holds if and only if each word occurring in  $s_1$  occurs also in  $s_2$ , e.g. (collision, atom collision), (beam injection, ion beam injection),
- a CONTRAST relation between opposite descriptors, e.g. (elastic scattering, inelastic scattering).

The descriptor classification rules of the PHYS/PILOT dictionary are based on a subject classification of the documents of the PHYS database. The underlying classification scheme is a 3 level decimal classification. For the descriptor classification, only the 10 classes from the first level are considered, and for each descriptor its probability of occurrence in a document belonging to class  $k_i, i = 1, \dots, 10$  was estimated based on the 392 000 document sample. The size of the PHYS/PILOT dictionary is illustrated in table 1.

Descriptors (with classifications)	22 683
Other terms	179 675
Single words	85 017
Phrases	94 658
Pairs $(t, s)$ in the relations:	
Relation $Z$	355 933
$t =$ single word	159 930
$t =$ phrase	170 697
$t =$ formula identifier	25 306
Relation USE	50 138
IDENTITY relation $t = s$	22 683
FORMULA IDENTITY relation	15 214
BROADER-TERM relation	165 020
FORMAL INCLUSION	27 787
CONTRAST relation	100
Total of related pairs	636 875

Table 1: Survey of the PHYS/PILOT dictionary

The AIR/PHYS system uses linear indexing functions in both indexing phases, where RDV1 consists of 34 elements and RDV2 of 70 elements (see [Knorz 86]). Instead of describing the indexing procedure of AIR/PHYS in detail, we present a more illustrative indexing example based on probabilistic classification trees below.

At the end of 1990, the data base PHYS contained about 1 400 000 documents. In 1990, 130 000 new documents were incorporated into the system. All incoming documents have to be included into the abstract journal PHYSICS BRIEFS. For this reason, they have to be classified into the classification system with about 600 classes at the lowest level. Further, another indexing is needed in order to represent the documents in the subject index of the abstract journal.

After the Fachinformationszentrum Karlsruhe had decided to apply automatic indexing to the input of the data base PHYS, we studied these further tasks of content analysis in order to accomplish them automatically, too. Experimental research has been done using modified versions of our approaches to dictionary construction and indexing. However, the results obtained until now are far below the quality level needed for a printed journal.

analysis of the same documents. Simultaneously with the classification and the register indexing, the human indexers check and correct the automatic indexing output. This can be done on-line or off-line.

By applying an appropriate cutoff value the weighted indexing produced by the AIR/PHYS system is transformed into a binary indexing with an average indexing depth of 12 descriptors/document. From this kind of a proposal the indexers cancel an average of 4 descriptors/document, and substitutes them by 4 other descriptors. In comparison with the previous average of 8.5 descriptors/document in the manual indexing for the data base PHYS, we have now an average of 12 manually checked resp. corrected descriptors/document. As a consequence higher recall ratios of retrieval results without substantial precision losses can be expected.

For each of the first 20000 documents to which the AIR/PHYS system has been applied the indexers had to judge globally if this contribution to the indexing work was really useful [Lück 88]. The automatic indexing of 19 per cent of the documents was considered to be good. Furthermore, the indexers accepted the automatic indexing of 63 per cent of the documents as being useful in spite of some corrections needed. However, in 18 per cent of the documents there were too many faults so that the automatic indexing was of no help for the indexers' work. After this evaluation the Fachinformationszentrum Karlsruhe decided to continue the application of the AIR/PHYS system and even to extend it to other databases.

The above-mentioned figures vary significantly between the different subfields of physics. Special difficulties arise when many formulas of high complexity must be treated, e.g. in nuclear physics and in the subfield of elementary particles.

The main source of indexing faults is the incompleteness of the already very large dictionary. Though most of the lacking data are needed only rarely their absence cumulates to a considerable error ratio. For example, the most effective relation  $Z$  covers only 10 000 (44 per cent) of the descriptors. The other descriptors had not been assigned often enough in the manual indexing used for the generation of the relation  $Z$  (cf. section 2). In consequence, the continuous maintenance of the dictionary is not limited to the insertion of new descriptors and their relations, but also more relations covering the already known descriptors have to be generated [Schwantner 88]. For this purpose, the manual corrections of the automatic indexing are evaluated.

## 5.2 Indexing example

We give an example of the application of the current version of the AIR/X system with the ID3 algorithm as indexing function in both phases. Figure 3 shows the example document from the PHYS database. As indexing dictionary, the PHYS/PILOT dictionary is used.

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

In metal/oxide/SnTe tunnel junctions (where the oxide is  $\text{Al}_2\text{O}_3$  or  $\text{SiO}_2$  and the metal is lead or aluminium) on  $\text{BaF}_2$  or  $\text{NaCl}$  substrates the tunnel current  $I(U)$  and its derivatives  $I'(U)$  and  $I''(U)$  were measured at 4.2 K. Additionally the Hall coefficient and electrical conductivity of the monocrystalline SnTe films were determined at the same temperature. The pronounced oscillations in  $I''$  suggest the existence of a quantum size effect in the very thin SnTe films in several cases, although this is complicated by various other processes. The most important features of the different types are discussed briefly.

Figure 3: Example document from the PHYS database

The first column of table 2 lists certain terms (see below) in their standard form identified in this document. Terms written in italics are formula identifiers; for example, *TIN@TELLURIDE* is derived from "SnTe", and the temperature value "4.2 K" is mapped onto the formula identifier *ULTRALOW@TEMPERATURE*.

For the definition of the FOCs, we consider for all kind of terms (single words, formula identifiers and phrases) the location of the term in the document, namely title vs. abstract. For noun phrases, we regard in addition the number of stopwords between the first and the last component of the phrase in the text: P+ with no stopwords, and P- with 1...3 stopwords in between. Examples for P+ are "TUNNEL JUNCTION", "TUNNEL CURRENT" and "ELECTRICAL CONDUCTIVITY", and for P- "ALUMINIUM SUBSTRATES", "MEASURE ELECTRICAL" and "FILM COEFFICIENT". Syntactically, the former three phrases are identified correctly, while all the phrases identified as P- are wrong. However, the phrase "MEASURE ELECTRICAL" can be regarded as being semantically correct, since the first sentence is about the measurement of electrical currents.

Based on these FOC definitions and the relations of the indexing dictionary, RDV1 as shown in table 3 was defined. Here we choose to define all elements of RDV1 as binary-valued, so no additional discretization was necessary.

For our example document, AIR/X creates RDs for 51 descriptors (That is, there is at least one descriptor indication from a term in the text for each of these descriptors). 23 of these descriptors do not occur in the

term	rel.	$z(t, s)$	descriptor	$a_1(\vec{x})$	$a_2(\vec{x})$	ass.
ALUMINIUM	Z	.44	ALUMINIUM	.60	.21	
ALUMINIUM	FID					
ALUMINIUM	ID					
ALUMINIUM METAL	Z	.36				
ALUMINIUM SUBSTRATE	Z	.56				
AL <sub>2</sub> O <sub>3</sub>	Z	.64	ALUMINIUM OXIDES	.52	.61	*
ALUMINIUM@OXIDE	FID					
ALUMINIUM@OXIDE	Z	.54				
BAF <sub>2</sub>	Z	.69	BARIUM FLUORIDES	.52	.61	A
BAF <sub>2</sub> SUBSTRATE	Z	.42				
BARIUM@FLUORIDE	FID					
BARIUM@FLUORIDE	Z	.65				
HALL	Z	.34	ELECTRIC CONDUCTIVITY	.84	.91	*
CONDUCTIVITY	Z	.50				
ELECTRICAL	Z	.34				
ELECTRICAL CONDUCTIVITY	USE					
ELECTRICAL COEFFICIENT	Z	.75				
DETERMINE ELECTRICAL	Z	.46				
HALL COEFFICIENT	Z	.51				
MEASURE ELECTRICAL	Z	.59				
FILMS	Z	.38	FILMS	.64	.63	A
FILMS	ID					
NACL SUBSTRATE	Z	.39				
FILM COEFFICIENT	Z	.39				
MONOCRYSTALLINE FILM	Z	.40				
QUANTUM FILM	Z	.32				
			FERMI LEVEL			M
HALL	Z	.75	HALL EFFECT	.67	.72	*
ELECTRICAL COEFFICIENT	Z	.30				
HALL COEFFICIENT	Z	.89				
CURRENT-VOLTAGE	Z	.57	IV CHARACTERISTIC	.76	.75	*
I(U)	Z	1.00				
JUNCTION EXPERIMENT	Z	.40	JOSEPHSON JUNCTIONS	.44	.49	A
JUNCTION OXIDE	Z	.30				
TUNNEL JUNCTION	Z	.41				
TUNNEL OXIDE	Z	.30				
			LEAD			M
TIN@TELLURIDE	Z	.46	LEAD TELLURIDES	.31	.36	A
			MIS JUNCTIONS			M
MONOCRYSTALLINE	Z	.57	MONOCRYSTALS	.60	.61	A
MONOCRYSTALLINE FILM	Z	.40				
OSCILLATIONS	Z	.36	OSCILLATIONS	.47	.46	*
OSCILLATIONS	ID					
SIO <sub>2</sub>	Z	.30	SILICON	.24	.28	
STISHOVITE	Z	.33				
SIO <sub>2</sub>	Z	.59	SILICON OXIDES	.52	.61	A
STISHOVITE	Z	.57				
SIZE EFFECT	ID		SIZE EFFECT	.86	.86	*
QUANTUM SIZE EFFECT	Z	.45				
QUANTUM SIZE	Z	.30				
NACL	Z	.57	SODIUM CHLORIDES	.52	.61	A
NACL SUBSTRATE	Z	.42				
SODIUM@CHLORIDE	FID					
SODIUM@CHLORIDE	Z	.52				
SPECTRA	ID		SPECTRA	.18	.10	
SUBSTRATES	ID		SUBSTRATES	.44	.49	A
NACL SUBSTRATE	Z	.42				
TEMPERATURE	Z	.50	TEMPERATURE DEPENDENCE	.71	.75	*
HALL	Z	.30				
ULTRALOW@TEMPERATURE	Z	.39				
ELECTRICAL COEFFICIENT	Z	.38				
HALL COEFFICIENT	Z	.47				
MEASURE ELECTRICAL	Z	.35				
THIN FILMS	Z	.53	THIN FILMS	.86	.91	*
THIN FILMS	ID					
NACL SUBSTRATE	Z	.32				
ALUMINIUM SUBSTRATE	Z	.30				
FILM COEFFICIENT	Z	.33				
MONOCRYSTALLINE FILM	Z	.30				
QUANTUM SIZE EFFECT	Z	.38				
QUANTUM FILM	Z	.41				
SNTE	Z	.72	TIN TELLURIDES	.92	.91	*
TIN@TELLURIDE	FID					
TIN@TELLURIDE	Z	.62				
TUNNEL DIODES	Z	.59	TUNNEL DIODES	.69	.72	*
TUNNEL DIODES	ID					
TUNNELS	Z	.34	TUNNEL EFFECT	.84	.91	A
JUNCTION EXPERIMENT	Z	.35				
JUNCTION OXIDE	Z	.33				
TUNNEL CURRENT	Z	.58				
TUNNEL EXPERIMENT	Z	.30				
TUNNEL JUNCTION	Z	.56				
TUNNEL OXIDE	Z	.50				
ULTRALOW@TEMPERATURE	FID		ULTRALOW TEMPERATURE	.48	.58	*
ULTRALOW@TEMPERATURE	Z	.50				

Table 2: Relevance descriptions, manual and automatic indexing of the example document

element	description
IOU	i.w. ID- or USE-relation?
USE	i.w. USE-relation?
FO-ID	i.w. FID-relation?
F-Z	i.w. Z-relation from formula identifier?
T+	indication from single word or "good" noun phrase (P+)?
TI	indication from term in the title?
IOU-TI	i.w. ID- or USE-relation from term in the title?
ID-TI	i.w. ID-relation from term in the title?
ID-P	i.w. ID-relation where descriptor is a phrase?
Z $\geq$ 0.8	i.w. Z-relation where $z(t, s) \geq 0.8$ ?
Z $\geq$ 0.7	i.w. Z-relation where $z(t, s) \geq 0.7$ ?
Z $\geq$ 0.6	i.w. Z-relation where $z(t, s) \geq 0.6$ ?
Z $\geq$ 0.5	i.w. Z-relation where $z(t, s) \geq 0.5$ ?
Z $\geq$ 0.45	i.w. Z-relation where $z(t, s) \geq 0.45$ ?
Z $\geq$ 0.40	i.w. Z-relation where $z(t, s) \geq 0.4$ ?
Z $\geq$ 0.35	i.w. Z-relation where $z(t, s) \geq 0.35$ ?
H $\geq$ 7	i.w. Z-relation where $h(t, s) \geq 7$ ?
H $\geq$ 10	i.w. Z-relation where $h(t, s) \geq 10$ ?
H $\geq$ 20	i.w. Z-relation where $h(t, s) \geq 20$ ?
Z5 $\geq$ 0.5	i.w. Z-relation where $z(t, s) \geq 0.5$ and $h(t, s) \geq 5$ ?
Z5 $\geq$ 0.45	i.w. Z-relation where $z(t, s) \geq 0.45$ and $h(t, s) \geq 5$ ?
Z5 $\geq$ 0.40	i.w. Z-relation where $z(t, s) \geq 0.40$ and $h(t, s) \geq 5$ ?
Z5 $\geq$ 0.35	i.w. Z-relation where $z(t, s) \geq 0.35$ and $h(t, s) \geq 5$ ?
Z5 $\geq$ 0.30	i.w. Z-relation where $z(t, s) \geq 0.30$ and $h(t, s) \geq 5$ ?
FO-ID+Z	i.w. FID- and Z-relation?
FO-ID+Z+	i.w. FID- and Z-relation, where the term occurs at least twice?
IOU-ZW	i.w. Z- and ID- or USE-relation from the same term?
IOU-ZW+	i.w. Z- and ID- or USE-relation, which occurs at least twice?
#I>1	at least 2 indications in the RD?
#I>2	at least 3 indications in the RD?
#I>3	at least 4 indications in the RD?
#MO>0	at least 1 term in the RD with multiple occurrence?
#MO>1	at least 2 term in the RD with multiple occurrence?
#MO>2	at least 3 term in the RD with multiple occurrence?
IOU>2	at least 3 i.w. ID- or USE-relation
IOU>3	at least 4 i.w. ID- or USE-relation
#DT>1	indications from at least 2 different terms?
#DT>2	indications from at least 3 different terms?
#DT>3	indications from at least 4 different terms?
#DZ>0	at least 1 term with Z-relation?
#DZ>1	at least 2 terms with Z-relation?
#DZ>2	at least 3 terms with Z-relation?
#DZ>3	at least 4 terms with Z-relation?
#DZ>4	at least 5 terms with Z-relation?
#Z-SW>0	at least 1 term with Z-relation from single word?
#Z-SW>1	at least 2 terms with Z-relation from single word?
#Z $\geq$ 4>2	at least 3 terms with $z(t, s) \geq 0.4$ ?
#Z $\geq$ 5>1	at least 2 terms with $z(t, s) \geq 0.5$ ?
#Z $\geq$ 5>2	at least 3 terms with $z(t, s) \geq 0.5$ ?

Table 3: Elements of RDV1 (i.w. = indication(s) with)

text. By application of the 1st indexing function, 26 out of the 51 RDs get an indexing weight  $a_1(\vec{x}) \leq 0.1$  (among these is one descriptor (LEAD) which was assigned by manual indexing). The major reason for these low weights is the fact that none of these RDs contains an indication with an element of the  $Z$  relation. The other 25 RDs are shown in abbreviated form (without FOCs) in table 2. In the first column, the terms forming indications to the descriptors shown in the fourth column are listed. The second column gives the name of the relation linking the term with the descriptor (Here ID denotes the IDENTITY relation and FID the FORMULA IDENTITY relation), and  $z(t, s)$  gives the corresponding weight of the  $Z$  relation, if applicable. The indexing weight from the first indexing phase is shown in the column headed  $a_1(\vec{x})$ . The probabilistic classification tree which yields these values has 125 leaves (classes  $l_j$ ) and a height of 14. Table 4 shows a part of the classification tree together with the RDs of the example document that fall into this subtree.

As an example, we describe the RD of the descriptor "HALL EFFECT" in more detail. For this descriptor, there are indications from three terms, all of which are based on the  $Z$  relation (and  $h(t, s) \geq 5$  in each case):

- the single word "HALL" identified in the abstract with  $z(t, s) = 0.75$ ,
- the noun phrase "ELECTRICAL COEFFICIENT" identified in the FOC "P-" in the abstract, with  $z(t, s) = 0.30$ ,
- the noun phrase "HALL COEFFICIENT" identified in the FOC "P+" in the abstract, with  $z(t, s) = 0.89$ .

```

if (IOU-ZW) then
  if (#MO>2) then 0.67
  else
    if (TI) then 0.95
    else 0.83
else
  if (FO-ID+Z+) then 0.92 {TIN TELLURIDES}
  else
    if (TI) then 0.76 {IV CHARACTERISTIC}
    else
      if (F-Z) then 0.52 {ALUMINIUM OXIDES, BARIUM FLUORIDES, SILICON OXIDES}
      else
        if (#DZ>4) then 0.84
        else 0.67 {HALL EFFECT}

```

Table 4: Part of the probabilistic classification tree

For the computation of the indexing weight for this RD, the probabilistic classification procedure shown in table 4 is applied, and the following elements of RDV1 lead to the final indexing weight  $a_1(\vec{x}) = 0.67$  for this RD:

- (#DZ>1): there are three indications with a  $Z$  relation,
- (Z5  $\geq$  0.5): there are two indications with  $h(t, s) \geq 5$  and  $z(t, s) \geq 0.5$ ,
- (Z  $\geq 5 > 1$ ): there are two indications with  $z(t, s) \geq 0.5$ ,
- (ID-P): there is no indication from a phrase in the title,
- (FO-ID+Z+): there is no FORMULA IDENTITY relation,
- (TI): there is no indication from a term in the title,
- (FZ): there is no  $Z$  relation from a formula identifier,
- (# DZ>4): there are less than 5 indications with  $Z$  relations.

In the second indexing phase, for all descriptors with  $a_1(\vec{x}(s, d)) > 0.1$  the relevance descriptions RD2 are formed by using the  $s \rightarrow s$  and  $s \rightarrow k$  rules from the dictionary. By application of the RDV2 definition (not shown here) and another probabilistic classification tree, we get the final indexing weights  $a_2(\vec{x})$ . The column headed “ass.” shows the assignment decisions (M — by manual indexing only, A — by automatic indexing only, \* — assigned by both). The effect of the second indexing phase can be seen most clearly with the descriptor “ALUMINIUM”: Since there is the more special descriptor “ALUMINIUM OXIDES”, the weight of the former is significantly reduced in the second phase.

### 5.3 Indexing experiments

For the development of the AIR/PHYS system, a large number of experiments was performed. These experiments are described in [Knorz 83] and [Knorz 86]. Here we want to present some results that compare the indexing procedure that is used in the Fachinformationszentrum Karlsruhe with some possible alternatives that are currently investigated by our research group.

In order to evaluate the quality of automatic indexing, we regard the coincidence with manual indexing<sup>2</sup>. For this purpose, we regard the consistency factor

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

where  $AUT$  and  $MAN$  denote for a given test set of documents the set of all automatic resp. manual descriptor assignments. Since our indexing system yields a weighted indexing, we regard the consistency factors for binary indexing derived from different cutoff values, and take the maximum  $q_{max}$  of these  $q$  values for a given sample.

In addition, we regard two measures similar to recall and precision for describing the quality of automatic indexing by taking manual indexing as a standard. For brevity, we will call these measures recall ( $r$ ) and precision in the following (although they relate to the quality of descriptor assignments and not to retrieval quality). With

<sup>2</sup>Of course, retrieval results are the final yardstick for indexing quality. This approach was taken in the AIR retrieval test [Fuhr & Knorz 84]. The results of the AIR retrieval test revealed that the comparison with manual indexing is also a good measure of indexing quality, so we prefer this method because it requires substantially less effort.

$$p = \frac{|AUT \cap MAN|}{|AUT|} \quad \text{and} \quad r = \frac{|AUT \cap MAN|}{|MAN|}$$

These measures are computed for different cutoff values, and then the corresponding  $(r, p)$  pairs are plotted in a recall-precision graph. In contrast to the consistency factor, this method is suited better to weighted indexing (which is important when these weights are used for ranking retrieval output, see e.g. [Fuhr 89a]).

As training and test sets for the development of indexing functions, different samples of 1000 documents each were considered in our evaluations. According to the origin of the manual indexing, there are two kinds of samples:

- Samples named *X* and *Z* have a purely manual indexing with an average indexing depth of 8.5 descriptors per document.
- The samples named 5 and 6 stem from the new input routine for the PHYS database, that is, first the automatic indexing procedure of the AIR/PHYS system was applied, and then these descriptor assignments are revised by the human indexers, thus leading to an indexing depth of 10.7 for this kind of manual indexing.

In the experiments described below, sample 5 is used as test set, and the other samples are training sets. All the evaluation figures given relate to the test set. First we describe experiments that compare the indexing functions of the AIR/PHYS system as it runs in the Fachinformationszentrum Karlsruhe with those based on probabilistic classification trees. The least square polynomial indexing function LSP1 of the first indexing phase of the AIR/PHYS system (with 34 elements in RDV1) was developed by using sample *X* as training sample. The corresponding function for the second indexing phase (LSP2) was derived from sample *Z*.

All the other indexing functions described in the following are based on sample 6 as training set. PCT1 is the probabilistic classification tree indexing function of the first indexing phase as described in the previous section, and PCT2 is the corresponding function of the second indexing phase. Table 5 gives the consistency factors for these indexing functions (for the test sample 5).

function	description	$q_{max}$
LSP1	first indexing phase of AIR/PHYS	0.439
PCT1	prob. class. tree, first indexing phase	0.449
LSP2	second indexing phase of AIR/PHYS	0.491
PCT2	prob. class. tree, second indexing phase	0.456
PCT-S	like PCT1, with word stems in FOCs	0.444
PCT-SE	like PCT-S, with extended dictionary	0.442

Table 5: indexing results

It can be seen that LSP1 performs slightly worse than PCT1. This result can be explained with regard to the different training sets used for the two indexing functions: LSP1 is based on the old, purely manual indexing, whereas PCT1 uses material from the new input routine for PHYS, and the test sample comes also from this new input routine (although disjoint from the training samples!). For the second indexing phase, however, the situation is different, since the LSP2 performs clearly better than PCT2 in terms of  $q_{max}$  values. Here the recall-precision graphs of these two functions shown in figure 4 should be considered. It can be seen that LSP2 performs clearly better than PCT2 for the low and high recall end of the curve. The reason for this strange result is the origin of the test sample used here: The manual indexing of this sample is in fact a revision of LSP2 done by human indexers, so this sample favours LSP2, and the comparison with PCT2 is not fair. So far no sample of revised automatic indexing based on another indexing is available for our experiments. When the results of PCT1 and PCT2 are compared, there is almost no difference in terms of  $q_{max}$  values for the two indexing phases (the recall-precision graphs not shown here also support this statement). We got similar findings for LSP1 and LSP2 when they were applied to pure manual indexing (see [Knorz 86]). At this point, a distinction between quantitative and qualitative evaluations of indexing quality should be made. When the results of the two indexing phases were shown to human indexers, they preferred the second indexing phase, mainly because there were less 'obvious' indexing errors. In terms of statistical measures, however, this judgement cannot be verified. In the following, we only regard indexing functions for the first indexing phase.

A second series of experiments was performed in order to investigate the effect of using additional FOCs or an extended indexing dictionary. For the experiments with additional FOCs (with a lower certainty of identification), we considered the word stems of single words and phrases (our stemming algorithm is described in [Kuhlen 77]).

PUTING', 'COMPUTATION', 'COMPUTE', 'COMPUTATIONAL'. Since our indexing dictionary is based on the standard form of terms, allowing word stems in FOCs will lead to new relevance descriptions with a lower indexing precision, but possibly increase indexing recall. E.g., in the example from above, a relation from the dictionary for the term 'COMPUTER' would form a descriptor indication even when only the term 'COMPUTE' is identified in the text. The actual definition of the new FOCs is as follows: corresponding to each FOC based on standard forms, an additional FOC with word stems was defined. For a phrase, no distinction was made whether only one or all of its components matched the word stem of a component of a phrase in the dictionary only (and not the standard form).

With these additional FOCs, the average number of RDs per document rises from 27.7 to 92.3. However, if all descriptors with RDs were assigned, indexing recall would only increase from 0.727 to 0.749. The average precision of RDs based exclusively on terms identified in their stem form is 0.004, in contrast to 0.281 for RDs with at least one term in standard form. So there is a very little increase in recall and a drastic loss in terms of precision. Some preliminary experiments also showed that an indexing function would yield weights of at least 0.1 for hardly any RD based on word stems only. So we can conclude that the precision of these RDs is too low for indexing purposes. For this reason, we do not consider RDs based exclusively on word stems in the following. For the other RDs, however, we also include the indications from terms in their stem form.

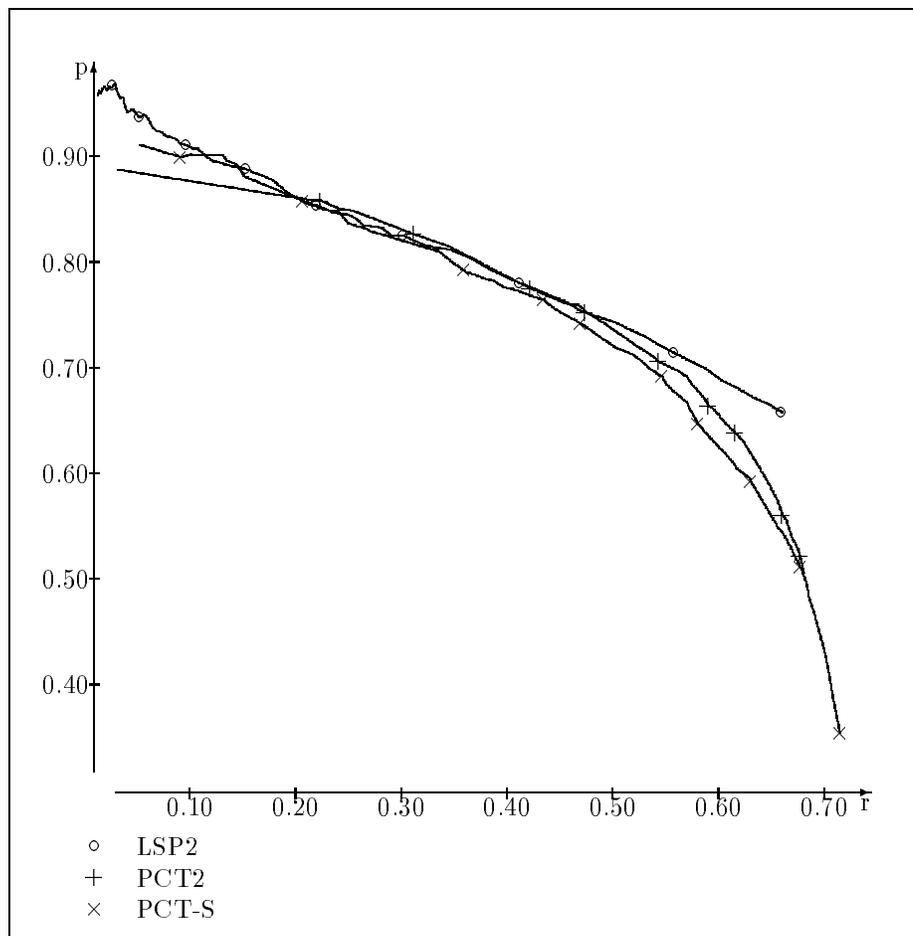


Figure 4: Recall-precision graphs of indexing functions

In order to define a RDV for these additional FOCs, we modified the definition of RDV1 shown in table 3 in the following way: The definition of the RDV elements shown here remains unchanged, that is, they refer to indications based on terms identified in their standard form only. Additionally, 20 elements were defined that relate to indications based on word stems. The results for the probabilistic classification tree indexing function PCT-S based on this RDV are shown in table 5 and figure 4. It can be seen that the consideration of additional FOCs with a lower certainty of identification leads to no improvements in terms of indexing quality.

Finally, we investigated the effect of an extended indexing dictionary on indexing quality. For this purpose, we added the 449 920 elements of the relation  $Z$  that had been computed, but not included in the PHYS/PILOT dictionary. When RDs are formed with this extended dictionary, the number of RDs per document rises from

stored in the dictionary, which is about twice as large after the extension. Since mainly term-descriptor pairs of the relation  $Z$  with low term frequencies were previously excluded from the dictionary, there is only a slight increase in the number of RDs generated.

For the development of the indexing function PCT-SE for the extended dictionary, we used the same RDV definition as for the function PCT-S from above. The  $q_{max}$  values for these two functions (see table 5) are almost identical (and there is almost no difference in the recall-precision graphs not shown here). So the effect of this extension of the indexing dictionary on indexing is negligible.

The results of the experiments described in this section justify two major decisions that were made in the development process for the AIR/PHYS system, namely the reduction of the indexing dictionary and the restriction of FOCs to the standard form of words. However, we will continue to investigate possible improvements of this system, e.g. better linguistic text analysis methods and the inclusion of descriptor-specific information in the dictionary.

## 6 Comparison with other work

In the literature, there is relatively little work on automatic indexing with a controlled vocabulary. This may be due to the fact that most IR researchers are convinced that indexing with terms from an unrestricted vocabulary yields a better retrieval quality than controlled vocabulary indexing. However, there are two major arguments justifying our approach:

- There is little evidence that automatic IR methods with free language terms are superior to manual indexing with controlled language terms. In [Salton 86] a survey about the retrieval tests concerning this point is given. Because of the diversity of factors influencing retrieval results, the evaluation of these tests does not allow us to make a general statement about the superiority of free term indexing over controlled term indexing or vice versa. Furthermore, most of the evaluations are based on relatively small collections, and it is not clear to what extent these results apply to large databases.
- In retrieval practice, manual indexing with a controlled language is in widespread use. Therefore retrieval practitioners can judge more easily about an automatic indexing method and will more likely accept it if it assigns controlled language terms as well. Above all, such a kind of automatic indexing can replace the manual indexing in an operational IR system without compatibility problems.

Following earlier work on automatic indexing at INSPEC (see e.g. [Aitchison & Harding 82]), a probabilistic indexing model for this approach is described in [Robertson & Harding 84]. Robertson and Harding propose a variant of the binary independence retrieval (BIR) model [Robertson & Sparck Jones 76] for application to the indexing task. In comparison to our approach, they consider additional probabilistic parameters in their model. In the DIA, the association factor  $z$  is an estimate of the probability  $P(C|s, t)$  that the assignment of  $s$  to a document is correct, given that  $t$  occurs in that document. Let  $\bar{t}$  denote the fact that  $t$  does not occur in the document, then the probabilities  $P(C|s, \bar{t})$  and  $P(C|s)$  (the probability that the assignment of  $s$  to a document selected randomly will be correct) are regarded in addition by Robertson and Harding. On the other hand, no FOCs are considered in this approach (only binary identification of terms). The indexing weight is computed directly from the probabilistic parameters by application of the BIR model. We have formulated extensions of the Robertson and Harding model that include also the FOC concept ([Fuhr 85],[Fuhr 90]). Experiments with these models, however, did not show significant improvements in comparison to the original AIR/X approach [Pfeifer 90].

The probabilistic indexing model described in [Fung et al. 90] can be seen as a variant of the Robertson and Harding model, with no independence assumptions. The formalism of probabilistic networks [Pearl 88] allows the consideration of dependencies between terms and between descriptors for the computation of the probabilistic indexing weights. Theoretically, this approach seems to be inferior to that underlying the AIR/X system (the FOC concept also could be included by extending the probabilistic networks appropriately). As with any probabilistic dependency model, the crucial point of the probabilistic network approach is the estimation of the probabilistic parameters (see [Fuhr & Hüther 89]). In the AIR/PHYS system, for example, most of the term-descriptor rules are based on less than 10 observations in the set of 392 000 manually indexed documents. For this reason, it seems almost impossible to derive probabilistic rules that are based on pairs or triplets of terms (in extension to phrases). A similar statement holds for dependencies between descriptors. The lack of statistical data of this kind is partially compensated by using descriptor-descriptor rules from the thesaurus in the second indexing phase of the AIR/X system. However, if dependency information about specific term or descriptor pairs is available, then this data can be included in the relevance description. The general concept of the decision step allows for the consideration of dependencies between elements of the relevance description:

independence assumptions about the elements of the relevance description.

A more application-oriented approach than the two models cited before is described in [Martinez et al. 87]. The system developed for American Petroleum Institute's Central Abstracting and Indexing Service uses three kinds of rules:

- rules for the identification of noun phrases, but without distinction of FOCs,
- term-descriptor rules from the thesaurus that are used in conjunction with the phrases identified in the document text,
- specific term-descriptor rules (developed manually) with additional conditions with respect to the document on the left-hand side of the rules.

Instead of a decision step, the system assigns all descriptors which are indicated by rules based on the current document text.

The Indexing Aid project described in [Humphrey 87] is aimed at the development of an interactive knowledge-based indexing system, so its task is not a fully automatic indexing of documents. However, the system uses knowledge about the descriptor hierarchy and about indexing rules (for human indexers) in order to support the indexer's work and to check the consistency of the set of descriptors assigned to a document. These functions are similar to the second indexing phase of the AIR/X system.

## 7 Future work

During 5 years, the application of the AIR/PHYS system at the Fachinformationszentrum Karlsruhe has shown that the concept of the AIR/X system is powerful enough to cope with real applications in large subject fields.

A major problem of such a rule-based automatic indexing system is the maintenance of the rule bases, for the following reasons :

- Erroneous rules have to be detected and corrected.
- Most important, indexing dictionaries are always incomplete. Even with the weak statistical criteria and the large number of manually indexed documents used for the development of the PHYS/PILOT dictionary, there is still a significant number of descriptors for which there are no elements in the relation  $Z$  - mainly because there are not enough observations for these descriptors. Due to the low number of assignments of these descriptors, the effect of this incompleteness is rather small in statistical terms. For terms with a low frequency  $f(t)$  (which are rather significant in many cases), the problem is similar.
- When the subject field evolves over time, descriptors are added to the thesaurus, and new terms occur in the document texts. For these new elements, additional term-descriptor, descriptor-descriptor and descriptor classification rules have to be added to the indexing dictionary. Furthermore, some older rules may become invalid, that is, either the corresponding probabilistic weight has to be corrected or the rule has to be deleted.

So dictionary maintenance requires methods for identifying weakness of the current dictionary and for formulating new rules. Since only a part of this work can be done fully automatically, we are investigating strategies for interactive and incremental dictionary development.

Even for a static dictionary, we are interested in improving the dictionary development procedures. The general goal is to minimize development effort and dictionary size, without substantial loss of indexing quality. With respect to this problem, the criteria currently used for the computation of the relation  $Z$  (that is, the cutoff values  $c_1$  and  $c_2$ ) have to be compared with other possible strategies. Some preliminary work in this area is reported in [Hüther 89] and [Hüther 90].

In close conjunction with the task of system maintenance is the use of an interactive indexing system. The need for an interactive system comes partially from the system maintenance problem, but also from many applications, where indexing is only a part of the whole content analysis needed. In the case of the AIR/PHYS system, all incoming documents have to be included into the abstract journal PHYSICS BRIEFS. For this reason they have to be classified into a classification system of 3 hierarchical levels with about 600 classes at the lowest level. Further, another indexing is needed in order to represent the documents in the subject index of the abstract journal. We have studied these further tasks of content analysis in order to accomplish them automatically, too. Experimental research has been done using modified versions of our approaches to dictionary construction and indexing. However, the results obtained until now are below the quality level needed for a printed journal.

Thus, the application of the indexing system AIR/PHYS does not make superfluous the manual content analysis of the same documents. Simultaneously with the classification and the register indexing, the human indexers check and correct the automatic indexing output.

For this kind of applications, the transformation of the AIR/X system into an interactive indexing system seems to be worthwhile. Above all, such a system must be able to explain to the indexer, why a descriptor is

offered in variable depth, comprehensiveness, and lay-out. Obviously, the system should also answer questions concerning statistical or dictionary information. Further, the indexer should have the possibility of modifying the number of automatic descriptor assignments. Finally, more intelligent techniques could be applied in order to react immediately to the indexers' assignment corrections. For example, the conformity of the corrections with the indexing rules of the underlying database could be checked, and new dictionary information could be inferred.

## References

- Aitchison, T.; Harding, P.** (1982). Automatic Indexing and Classification for Mechanized Information Retrieval. In: *Proceedings of the EURIM 5*. Versailles.
- 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.; Liu, C.** (1968). Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Transactions on Information Theory* 14(3), pages 462–467.
- Faißt, S.** (1990). *Development of Indexing Functions Based on Probabilistic Decision Trees (in German)*. Diploma thesis, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II.
- Fangmeyer, H.; Lustig, G.** (1969). The EURATOM Automatic Indexing Project. In: International Federation for Information Processing (ed.): *IFIP Congress 68, Edinburgh*, pages 1310–1314. North Holland Publishing Company, Amsterdam.
- Fangmeyer, H.; Lustig, G.** (1970). Experiments with the CETIS Automatic Indexing System. In: International Atomic Energy Agency (ed.): *Proceedings of the Symposium on the Handling of Nuclear Information*, pages 557–567.
- Freeman, D.** (1987). *Applied Categorical Data Analysis*. Dekker, New York.
- Fuhr, N.; Buckley, C.** (1990). Probabilistic Document Indexing from Relevance Feedback Data. In: Vidick, J.-L. (ed.): *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, pages 45–61. ACM, New York.
- Fuhr, N.; Hüther, H.** (1989). Optimum Probability Estimation from Empirical Distributions. *Information Processing and Management* 25(5), pages 493–507.
- Fuhr, N.; Knorz, G.** (1984). Retrieval Test Evaluation of a Rule Based Automatic Indexing (AIR/PHYS). In: Van Rijsbergen, C. (ed.): *Research and Development in Information Retrieval*, pages 391–408. Cambridge University Press, Cambridge.
- Fuhr, N.** (1985). A Probabilistic Model of Dictionary Based Automatic Indexing. In: *Proceedings of the riao 85 (Recherche d'Informations Assistée par Ordinateur)*, pages 207–216.
- 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 Based on the Probability Ranking Principle. *ACM Transactions on Information Systems* 7(3), pages 183–204.
- Fuhr, N.** (1990). *Log-Linear Indexing Functions and the Darmstadt Indexing Approach*. Internal Report DV II 89-4, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II.
- Fung, R. M.; Crawford, S. L.; Appelbaum, L. A.; Tong, R. M.** (1990). An Architecture for Probabilistic Concept-Based Information Retrieval. In: Vidick, J.-L. (ed.): *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, pages 455–468. ACM, New York.
- Hart, A.** (1985). Experience in the Use of an Inductive System in Knowledge Engineering. In: Brauer, M. (ed.): *Research and Development in Expert Systems: Proceedings of the Fourth Technical Conference of the British Computer Society Specialist Group on Expert Systems*, pages 17–126. Cambridge University Press, Cambridge.
- Humphrey, S.** (1987). Illustrated Description of an Interactive Knowledge Based Indexing System. In: Yu, C. T.; van Rijsbergen, C. J. (eds.): *Proceedings of the Tenth Annual ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 73–90. ACM, New York.
- Hüther, H.** (1989). *Wachstumsfunktionen in der automatischen Indexierung*. Dissertation, TH Darmstadt, Fachbereich Informatik.

- Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, pages 313–326. ACM, New York.
- Jaene, H.; Seelbach, D.** (1975). *Maschinelle Extraktion von zusammengesetzten Ausdrücken aus englischen Fachtexten*. Report ZMD-A-29, Beuth, Berlin, Frankfurt.
- Kienitz-Vollmer, B.; Reichardt, J.** (1986). Bestimmung von Mehrwortgruppen mithilfe des Begrenzerverfahrens. In: Lustig, G. (ed.): *Automatische Indexierung zwischen Forschung und Anwendung*, pages 18–30. Olms, Hildesheim.
- Knorz, G.** (1983). *Automatisches Indexieren als Erkennen abstrakter Objekte*. Niemeyer, Tübingen.
- Knorz, G.** (1986). Die Anwendung von Polynomklassifikatoren für die automatische Indexierung. In: Lustig, G. (ed.): *Automatische Indexierung zwischen Forschung und Anwendung*, pages 98–126. Olms, Hildesheim.
- Kuhlen, R.** (1977). *Experimentelle Morphologie in der Informationswissenschaft*. Verlag Dokumentation, München.
- Lück, W.** (1988). Erfahrungen mit dem automatischen Indexierungssystem AIR/PHYS. In: Deutsche Gesellschaft für Dokumentation (ed.): *Deutscher Dokumentartag 1987*, pages 340–352. VCH Verlagsgesellschaft, Weinheim.
- Lustig, G.** (1982). Das Projekt WAI: Wörterbuchentwicklung für automatisches Indexing. In: *Deutscher Dokumentartag 1981*, pages 584–598. K.G. Saur, München, New York, London, Paris.
- Lustig, G. (ed.)** (1986). *Automatische Indexierung zwischen Forschung und Anwendung*. Olms, Hildesheim.
- Martinez, C.; Lucey, J.; Linder, E.** (1987). An Expert System for Machine-Aided Indexing. *Journal of Chemical Information and Computer Science* 27, pages 158–162.
- Pearl, J.** (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, San Mateo, Cal.
- Pfeifer, U.** (1990). *Development of Log-Linear and Linear-Iterative Indexing Functions (in German)*. Diploma thesis, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II.
- Quinlan, J.** (1984). Learning Efficient Classification Procedures and their Application to Chess End Games. In: Michalski, R.; Carbonell, J.; Mitchell, T. (eds.): *Machine Learning: An Artificial Intelligence Approach*, pages 463–482. Springer, Berlin, Heidelberg, New York, Tokyo.
- van Rijsbergen, C.** (1977). A Theoretical Basis for the Use of Co-Occurrence Data in Information Retrieval. *Journal of Documentation* 33, pages 106–119.
- Robertson, S.; Harding, P.** (1984). Probabilistic Automatic Indexing by Learning from Human Indexers. *Journal of Documentation* 40, pages 264–270.
- Robertson, S.; Sparck Jones, K.** (1976). Relevance Weighting of Search Terms. *Journal of the American Society for Information Science* 27, pages 129–146.
- Salton, G.** (1986). Another Look at Automatic Text-Retrieval Systems. *Communications of the ACM* 29(7), pages 648–656.
- Schwantner, M.** (1988). Entwicklung und Pflege des Indexierungswörterbuches PHYS/PILOT. In: Deutsche Gesellschaft für Dokumentation (ed.): *Deutscher Dokumentartag 1987*, pages 329–339. VCH Verlagsgesellschaft, Weinheim.
- 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.; Chiu, D.** (1987). Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9(6), pages 796–805.
- Wong, S.; Ziarko, W.; Raghavan, V.; Wong, P.** (1987). On Modeling of Information Retrieval Concepts in Vector Spaces. *ACM Transactions on Database Systems* 12(2), pages 299–321.
- Yu, C.; Buckley, C.; Lam, K.; Salton, G.** (1983). A Generalized Term Dependence Model in Information Retrieval. *Information Technology: Research and Development* 2, pages 129–154.