# Digital Libraries

**Reginald Ferber**

**Ulrich Thiel**

**Karl Aberer**

**Peter Fankhauser**

**Kostas Tzeras**

**GMD — IPSI**
**Darmstadt**

*German National Research Center for Information Technology*

*Integrated Publication and Information Systems Institute*

**http://www.darmstadt.gmd.de/IPSI**

# Part 1: What's all about?

## 1.1: Digital Libraries

> *Any attempt to define what a digital library is or is not will either be too vague or too restrictive*
>
> Editorial Int J. Digital Libraries Vol 1/1 1997

> *It is rather an environment to bring together collections, services, and people in support of the full life cycle of creation, dissemination, use, and preservation of data, information, and knowledge.*
>
> Santa Fe Workshop

- defined by function not by technology
- involve many domains of research, development, and service
- are characterized by integration of heterogeneous sources, services, traditions, and technologies, many of them involving humans and their way to deal with information

This course will cover some of these aspects

### *1.1.a Content / Objects of a DL*

Examples

- *Document Like Object (DLO)* (Dublin Core)
- *data, information, and knowledge* (Santa Fe Workshop)
- *article, book, web page, data item, video clip, map, drawing, photo, advertisement, animation*, ... (Winograd)
- *text, images, sound, movies, video, software, animations*
- Specific collections: chemistry, biology, geography, software ...

Possible Definition:

Self contained information objects that are collected for a relevant group of users over a period of time and can be anticipated by these users in the form in which they are presented by the DL system.

### *1.1.b Services*

Integrated management for

- **Content based access**: search / filter / browse / explore
- **Presentation of samples of objects**: show / filter / cluster / evaluate
- **Presentation of single objects**: display / deliver / abstracting / extracting
- **Authentication**: control originality / control references / copy detection / find citations
- **Access control**: copyright / billing / restrict access to specific groups (decency)
- **Communication**: comment / annotate / evaluate / discuss / contact author

adaptive to: language / problem considered / knowledge & education / technical & physical restrictions / time & money

*1.1.c Aspects of Content*

- **Diversity**: topic / format / quality / access methods
- **Quality / Originality**: integrity / reviewing / control of authorship
- **Commercial Aspects**: Copyright / billing / responsibilities / warranties
- **Internal Representation**: formats / metadata / ways of storage
- **Distributed storage**: dynamics of collections / merging / mediation / description of collections
- **Archiving**: security / media life-span / hardware availability / format conversion / definition of archival object

## 1.2: Content of the Course

### Part 1: Information Retrieval (Reginald Ferber)

- What's all about
- History
- Different kinds of information systems
- Boolean retrieval
- Stemming
- Classifications
- Thesauri
- Vector space model
- Evaluation of IR systems
- TREC
- Recent developments

### Part 2: Inelligent Access to Digital Libraries (Ulrich Thiel)

### Part 3: Document Models and Applications / XML and related standards (Karl Aberer)

- Basic XML architecture
- History of XML
- Application Examples
- XML Syntax and Document Type Definitions
- XSL - Layout for XML Documents
- XLink and Xpointer - Links in XML
- RDF - Metadata for XML
- Programming in XML
- XML Tools

## Part 4: Data management for documents (Peter Fankhauser)

- Database technology and the Web
- Database architectures for the Web
- Access mechanisms to databases in the Web
- Storage of structured documents
- Indexing for documents
- Document query languages
- Document query processing

## Part 5: Applications (Kostas Tzeras)

# Part 2: Information Retrieval

## 2.1: Introduction

- The amount of digital data is increasing world wide
- More people get access to networks like the Internet

⟶ These data are only useful, if people know
  - how to find them
  - what they mean
  - how to use them

### *2.1.a Examples of Information Needs*

Find a phone number
Users of the information service in general know

- what kind of information they look for
- where they find the information
- what a telephone is good for
- how to use it
- which person they want to call

Users assume that the number given is correct.
Service is organized by a central data base

Browse the WWW

- No specific purpose given
- defining aspects: protocols HTTP & HTML
- no central organization and responsibility
- few mandatory procedures
- developing style of interaction
- divers motivations of users:
  - looking for scientific information
  - participating in specific interest groups
  - banking
  - planning holidays
  - entertainment
  - shopping
  - no clear goal

⟶ many possibilities to start with
⟶ difficult to search for specific content
⟶ users have to understand / learn services
⟶ hard to judge if the information found is complete
⟶ hard to judge if the information found is correct

## 2.2: A Bit of History

The task to identify and organize documents or knowledge is neither new nor restricted to digital material.

There are long traditions in science and libraries to

- collect documents
- assess and evaluate their content
- organize and preserve them
- give people access to them

Within the last 40 years many computer based systems have been developed to organize the access to scientific documents.

But these systems have been build for well defined and homogeneous

- collections
- domains and
- users

These conditions change. Now systems have to adapt to

- divers types of documents and services
- heterogeneous content
- heterogeneous user groups
- distributed servers
- heterogeneous social and technical conditions

## 2.3: Different Kinds of Information Systems

To give a feeling for the topic

- diverse types of information
- diverse methods to handle it
- diverse motivations of users

will be discussed in the following examples

### 2.3.1: Searching for Scientific Documents

#### *2.3.1.a Methods*

- ask an expert
- look for a book on the topic
- follow citations and references
- use a bibliography, an abstracting service, or a bibliographic database specific to the domain
- look in the web
  - by browsing
  - using search engines

#### *2.3.1.b Characteristics and Problems*

- *ask an expert*
  - problem to find someone
  - gives the state of knowledge and the view of the respective person
  - chance to discuss and elaborate the information need

ferber@darmstadt.gmd.de

- *look for a book on the topic*
  - textbooks are slow: new fields are covered only after a while
  - conference proceedings do not offer a systematic introduction to a domain
  - selection criteria for proceedings are not only governed by content
- *follow citations and references*
  - citations my be limited to a specific view
  - in general only the title of a reference is know; its often unclear, what the content is about.
  - citations are only “backward” in time
  - sometimes it is difficult and time consuming to obtain the referenced material
- *use a bibliography, an abstracting service, or a bibliographic database specific to the domain*
  - bibliographies use complex structures for content organization like classifications. Users have to be familiar with these systems
  - bibliographic databases are expensive and sometimes complex to use. Access is restricted to customers
  - systems offer in general only references to documents
- *look in the web*
  - link pages in the web are often provided by single persons resulting in similar problems like asking an expert. But: only limited interaction and personal communication
  - no established standards of citation
  - little control of quality and correctness
  - many pages are no longer maintained
  - search engines are limited by many factors
    - no uniform structure of documents
    - formats other than HTML
    - “for sale” material
    - material provided from databases

### 2.3.2: Example of a Search in a Scientific Database

Information need: **Retrieval systems for multimedia objects especially for images**

Database INSPEC contains documents that describe articles and books:

- bibliographic data
- abstract
- classification codes
- key terms

(see fig. 1)

Generate a query: `RETRIEVAL SYSTEMS and MULTIMEDIA and IMAGES`

The (Boolean) retrieval system selects all documents that contain all the three terms. There are three hits for January to June 1995: “Image Engine: an object-oriented multimedia database for storing, retrieving and sharing medical images and text”, “Multimedia information retrieval using knowledge in encyclopedia texts”, “Images database management system: a ’server-client producer’ system on a local network and on the Internet”

Different queries produce different result sets: `RETRIEVAL and MULTIMEDIA and IMAGE$` results in 35 hits including: “PhotoFile: a digital library for image retrieval”, “Spatial knowledge representation and retrieval in 3–D image databases”, “Multimedia retrieval technology”,“A WWW interface to the OMNIS/Myriad literature retrieval engine”, “Problems of content-based retrieval in image databases”, It is not obvious that these documents are less adequate to the problem than those of the first query. More data are given in fig. 2.

**Figure 1: A document from INSPEC**

```
 INSPEC XXXXXXXXX
 Doc Type:     Journal Paper
 Title:        Images database management system: a 'server-client
               producer' system on a local network and on the Internet
 Authors:      Ageron, P.; Besson, F.; Desfarges, P.
 Affiliation:  Univ. Lumiere, Lyon, France
 Journal:      Computer Networks and ISDN Systems
               Vol: 26  Iss: suppl., no.2-3  p. S101-6
               Date: 1994
               Country of Publication: Netherlands
               ISSN: 0169-7552  CODEN: CNISE9
               CCC: 0169-7552/94/$07.00
 Language:     English
 Treatment:    Practical
 Abstract:     Lumiere University offers a full range of arts and social
   sciences. The research resources are very large and are specialized
   in politics, economics, finance, arts and archeology. As soon as the
   first network services were available, researchers tried to find
   information on the network about digitised photo management and
   everything about multimedia communication. The 'Images Database
   Management System' program tries to manage all information about
   digitised images: to give somebody access to, to modify and work on
   images and to export images into other information retrieval systems.
   This application program is based on a special definition of an
   'entity'. This is a physical entity (Photo CD) or a folder or
   directory. It may also be a book, if images are reproduced from a
   document. These entities are named CDphoto, folder or book while
...
   Network like WAIS, WWW, or GOPHER and hope to place this tool at the
   Internet's users' disposal. (0 Refs.)
 Classification: C6160S (Spatial and pictorial databases); C6130M
   (Multimedia); C5620L (Local area networks); C6150N (Distributed
   systems software); C5620W (Other computer networks)
 Thesaurus: Client-server systems; File servers; Internet; Local area
   networks; Multimedia communication; Multimedia computing; Visual
   databases
 Free Terms: Images database management system; Server-client producer
   system; Local network; Internet; Digitised photo management;
   Multimedia communication; Information retrieval systems; Photo CD;
   Text information; SUN servers; OS Macintosh; HyperCard; JPEG Format;
   Communication protocol; Unix directories; GIF format
```

### 2.3.3: Data Retrieval

Data retrieval is characterized by strongly structured data.

Objects are described by tuples of attribute values, that have a well defined (and simple) type like

- boolean (false / true)
- limited sets of strings
- numbers

These types allow easy comparisons like

- same / different
- =, <, >, ...
- Hamming distance (number of different characters in two strings)

**Figure 2: Number of documents found in INSPEC with different queries for the time interval January – June 1995**

| Query | Number of documents found |
|---|---|
| RETRIEVAL SYSTEMS and MULTIMEDIA and IMAGES | 3 |
| RETRIEVAL SYSTEMS and MULTIMEDIA and IMAGE$ | 5 |
| RETRIEVAL and MULTIMEDIA and IMAGE$ | 35 |
| RETRIEVAL and MULTIMEDIA | 148 |
| RETRIEVAL or MULTIMEDIA | 2559 |
| RETRIEVAL or MULTIMEDIA or IMAGE$ | 9364 |

*This query contains terms connected by the Boolean operators "and" and "or". If two terms are connected by "and" those documents are selected that contain both terms; if they are connected by "or" documents that contain either the first one or the second one or both are selected. This means in particular, that result sets from term connected by "or" include the result sets of a query in which the same terms are connected by "and".*

**Figure 3: Database entries**

| | $m^2$ | Kaltmiete | Zimmer | Balkon | Ort | Stockwerk | Heizung |
|---|---|---|---|---|---|---|---|
| A | 64 | 820 | 3 ZKB | n | Kranichstein | 13 | zentral |
| B | 78 | 1200 | 4 ZKB | j | Bessungen | 2 | Gasetage |
| C | 86 | 1475 | 3 ZKB | j | Martinsviertel | 4 | zentral Fussboden |
| D | 102 | 580 | 3 | n | Wiebelsbach | EG | Ofen |
| E | 36 | 680 | 2 ZKB | j | DA-Ost | 3 | Nacht-speicher |
| F | 34 | 640 | 3 ZKB | n | Arheilgen | EG | Oel |
| G | 38 | 590 | 1,5 ZB | j | Griesheim | 2 | zentral |
| H | 87 | 890 | 4 ZKB | n | Heimstätten-siedlung | 3 | zentral |

*Values in different columns have different types. Column 2 and 3 contain real numbers that can be compared in the usual way. However a evaluation of these numbers will show, that for column 2 higher values are "better" while for column 3 lower values are "better". Evaluation the other columns with respect to what is "better" depends on the circumstances.*

*Another observation is that some entries become more meaningful if they are combined with values of other attributes.*

Tuples are selected according to these comparison operations.

However, to find data sets for a specific information need, it is necessary to know the semantics of the entries.

To select a “good” data set additional domain knowledge and elaborated optimization procedures are necessary.

## 2.3.4: Hypertext Systems

The WWW offers hypertext functionality on the Internet

Organizations like

- universities,
- corporations
- cities
- societies

as well as single persons offer Information on “web sites” beginning with a “homepage”

⟶ information has to be structured “around” the homepage
⟶ often hierarchial structures are used
⟶ it has to be possible to find any information using a “path” that starts at the homepage
⟶ users must be enabled to decide on every step in that path which way to go

### *2.3.4.a Example*

TheHomepage of Darmstadt (http://stadt.darmstadt.gmd.de/) contains for example six buttons:

- Städtische Einrichtungen
- Kunst & Kultur
- Zu Gast in Darmstadt
- Darmstädter Leben
- Wirtschaft
- Darmstadt aktuell

To find informations on hotels, one can proceed in the following steps “Zu Gast in Darmstadt”, leading to “Hotels und Restaurants”, and further to “Hotels in der Innenstadt”, “Hotels in den Vororten” etc. “Hotels in der Innenstadt” finally offers a list of hotels, some of them with their own homepages.

For information on parks it is not clear which link to follow.

- “Städtische Einrichtungen” offers a long list of links to offices including “Gartenamt”, but there only the office hours can be found.
- “Darmstädter Leben” offers a link to “Sport und Freizeit” but only a very general notice about the “nice parks”
- ”Zu Gast in Darmstadt” offers a link to “Virtueller Stadtrundgang”. This virtual walk through the city includes pages on the various parks in Darmstadt.

General observations: Hypertext systems have to keep the balance between clear structure according to one specific view and massive linking bearing the danger that people feel lost in hyperspace.

© Ferber GMD-IPSI
ferber@darmstadt.gmd.de

### 2.3.5: Expert Systems

Expert systems give answers to specific questions

Example: travel planning

- clearly structured questions
- many conditions
- many divers answers
- in many divers forms

To find a train connection from A to B it is necessary to

- find possible lines
- evaluate connections according to specific conditions
- select the best one according to
  - distance
  - time needed
  - price
  - convenience (number of change)
- (select and) generate presentation format

$\longrightarrow$ Information provided is no longer a fixed object but generated from a suitable knowledge base.

$\longrightarrow$ knowledge has to be provided in suitable form to be handled by the system

### 2.3.6: Management Information Systems

– Management information systems decision support systems data warehouseintranet are defined by the functionality they should provide:

- uniform and controlled access to documents and information in an institution
- selection of and access to vital information for the management
- descriptions of various alternative action plans
- predictive descriptions for future developments

### 2.3.7: Specific Systems for Specific Domains

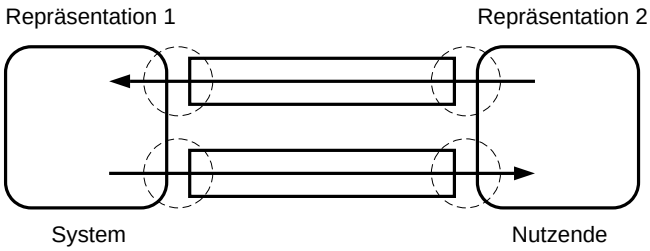
Specific domains have specific

- problems and needs
- ways to structure information
- specific data formats to search in

Examples:

- Geographic Information systems
- Chemistry
- Biology
- Mathematics / Physics

**Figure 5: Simple Dialog Scheme**



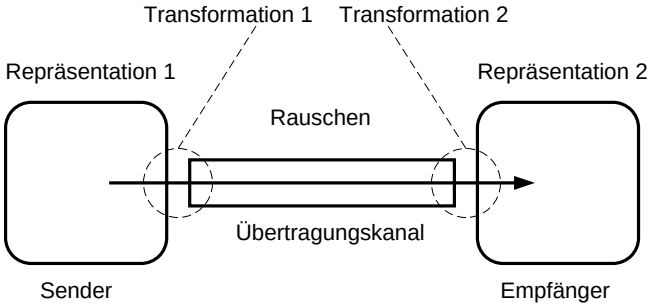

*The easiest way to model a dialogue: Add one channel*

# Part 3: Knowledge Representations and Retrieval Models

## 3.1: Models of Communication and Interaction

### 3.1.1: Transfer of Information

**Figure 4: Basic Scheme of Information Transfer**




*Information is available at the sender in a specific format. To transport it through the channel it has to be transformed into an appropriate format. At the receiver side it has to be transformed into an adequate format.*

- Errors occurring in the channel can be detected if not every pattern that the channel supplies is “legal”
- If the distance between patterns is large enough corrupted messages can be recovered by inferring what pattern was sent based on the pattern received. Example: typing errors are detected because:
  - the resulting string is no word
  - the resulting chain of words is no sentence
  - the resulting sentence makes no sense

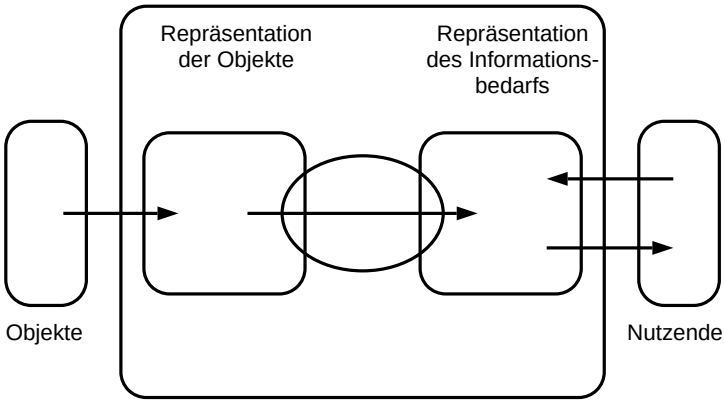
  Critical:
  - names: little knowledge about allowed patterns
  - numbers: Every sequence of digits is a number. Solution: writing numbers as words

### 3.1.2: Dialogues

Simple solution: Add one channel

But it is useful to include representations of both sides into the model. A representation of the objects and a representation of the users information need. This allows to better support and control dialogues and searches, for example by additional knowledge bases, changing existing queries, use of representations of objects to optimize queries.

**Figure 6: Basic Scheme of an Information system**




*Objects and information need of users are represented within the model of IR. The information channels are replaced by a more complex interaction and comparison mechanism*

## 3.2: Boolean Retrieval

Boolean retrieval is still the most popular retrieval model
Advantages:

- easy to understand why a document was found
- easy to implement

Problems:

- difficult to formulate advanced queries
- difficult to understand which documents were **not** found
- no ranking of results
- based on strings of characters
- adequate for language?

### 3.2.1: The Boolean Model

Idea:

- describe documents by attributes
- use set operations for retrieval

#### *3.2.1.a Attributes*

Let $D$ be a set of documents, $T$ a set of values. A function $t : D \rightarrow T$, $t(d) = t_i$ is called an **attribute**.

$$D_{t,t_i} = t^{-1}(\{t_i\}) = \{d \in D \mid t(d) = t_i\}$$

is the set of documents in which the attribute $t$ takes the value $t_i \in T$.

### *3.2.1.b Queries*

Queries are constructed using **attribute – value pairs** like $q = (t, t_1)$. For this **simple query** the result set is $D_q := D_{t,t_i}$

Complex queries are constructed by combining queries with the operators **AND** and **OR** as well as the use of the operator **NOT**:

$(t, t_1)$ AND $(s, s_1)$ denotes the intersection $D_{t,t_1} \cap D_{s,s_1}$ of result sets, $(t, t_1)$ OR $(s, s_1)$ denotes the disjunction $D_{t,t_1} \cup D_{s,s_1}$ of result sets and the unary Operator NOT $(t, t_1)$ denotes the complement $D \setminus D_{t,t_1}$ of a result set. In general NOT is used only together with AND: it excludes documents with a specific attribute value.

These operations can be applied as well to result sets of complex queries, leading to rather complex descriptions of sets of documents.

## 3.2.2: Boolean Text Retrieval

Attributes: occurrence of a term in the text of a document or in a specific part of a document

A term is defined as a sequence of characters with well defined boundaries.

Example: The reference Database document (fig. 1) has various fields:

- Title
- Authors
- Journal
- Abstract
- Thesaurus
- Free Terms

Boolean queries are formulated using field names and terms. Examples: author=smith, author: smith, smith in author ...

Attribute $TI_x : D \rightarrow \{0, 1\}$: Term $x$ appears in Title

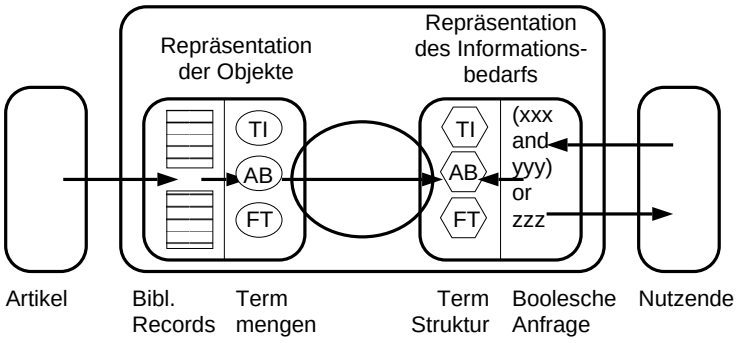
Attribute $AU_x : D \rightarrow \{0, 1\}$: Term $x$ appears in Author

Attribute-value-pair for boolean search: $(TI_x, true)$

Result set:

$$D_{TI_x,true} = TI_x^{-1}(\{true\}) = \{d \in D \mid Title\ of\ d\ contains\ x\}$$

**Figure 7: Model of a Text-based Boolean Information Retrieval Systems**




*Objects are described in intellectually generated documents. These documents are again represented as sets of terms in different fields.*

ferber@darmstadt.gmd.de

## 3.2.3: Implementation

Boolean retrieval systems are in general implemented using inverted files: for each term occurring in a collection the documents it appears in are listed. This allows fast access but causes storage overhead.

### *3.2.3.a Controlled Vocabulary*

List of all terms that can be used in a search

Construction:

- define rules for the decomposition of text into words: define word boundaries, treatment of hyphens, dashes, numbers, ...
- define a list of **stop words** to be excluded from the vocabulary: Very frequent words like articles, prepositions, “and”, “of”, ... that are not useful to characterize content, but would enlarge the inverted list.
- define rules to exclude further words like roman numbers, single characters, ...
- all remaining words are used as terms of the controlled vocabulary

### *3.2.3.b Construction of an inverted list*

In principle an inverted list can be constructed as follows:

- decompose each document of the collection into terms according to the rules for the controlled vocabulary
- add the document number an the location within the document to these terms (-> “terms within location list”)
- sort these pairs alphabetically according to the terms
- for pairs with the same term construct a list of document numbers
- concatenate the single lists of documents to a file and construct a list of the terms containing pointers to the starting points of the respective lists in that file

In practice more sophisticated methods for construction and more elaborated access structures can be used (See for example Frakes Baeza-Yates 1992, Harman, Baeza-Yates, Fox and Lee 1992, Fox and Lee (1991 described in Frakes Baeza-Yates 1992).

### *3.2.3.c Processing of a query*

- decompose the query into terms
- for each term use the inverted list to get the list of documents
- construct new lists of documents according to the operators connecting the terms until one single list remains
- show the number of documents in this final list
- get the documents of the list if the user requests them.

### *3.2.3.d Further Features*

Most Boolean retrieval systems offer further features:

- comparison operators (< >) for numbers (years)
- search for composed “terms” (groups of several words)
- distance functions (allow at most / exactly n terms between term a and term b)
- expansion of truncation –> set of terms matching a given pattern.

## 3.3: Strings, Terms, Concepts

Terms have been defined as strings that obey a number of rules based on character patterns

But: the words of a language are used to describe contents.

Their “meaning” defines important relations, that cannot always been mapped using patterns of characters

**Figure 8: Truncations that do not only exclude animals (from Ferber, Wettler, Rapp 1995)**

```
3  273 FIND CT D Vertebrates
4   24 FIND CT D Invertebrates
5  346 FIND ALL Animal$
6 2981 FIND ALL [Cat$;Dog$;Pidgeon$;Monkey$;
           Rat$;Dog$;Mouse;Mice;Rabbit$]
7 3264 FIND 3 TO 6
8   15 FIND 2 NOT 7
```

*The first column gives a reference number for each single query; the second gives the number of hits, the query starts with FIND. Numbers after FIND refer to the result sets previous queries.*

*Line 6 contains a query including truncated names of animals often used as subjects in psychological research. The documents found with this query are excluded in line 8. But line 6 does not only specify animals but also terms like Category, Dogmatism, or Rating. As a consequence documents containing these terms are also excluded from the final result set. Probably this was not intended by the searcher.*

There are several approaches to use the meaning of words in IR. They can be characterized by two different methods:

- to represent and process language in such a way that similarities are used
- to restrict the means of description in such a way, that the built-in structures of the descriptions represent the similarities.

### 3.3.1: Stemming

Assumption: morphological variants of a word share the basic meaning.

Various grades of reduction:

- reduce to basic form of the same type (noun, verb, ...)
- reduce to stems conflating various types

Effects for IR:

- smaller inverted lists (if done at indexing time)
- generalization of meaning

#### *3.3.1.a Replacement Rules*

Approach: scan word endings for specific patterns and replace these patterns.

⟶ fast algorithm

⟶ few resources needed

⟶ no need to “know” words or stems

Evaluation (according to Kuhlen 1977):

- For a text with 72 000 different words (“types”) it can be expected with a probability of 0.95 that the error rate will be less than 0.005
- For the same corpus the reduction rates can be expected between 13 % (lexikographische Grundform) and 27.3 % (stemming)

© Ferber GMD-IPSI
ferber@darmstadt.gmd.de

**Figure 9: Various depths of reduction (from Kuhlen 1977, p. 58)**

| *Formale Grundform* | *Textwörter* | *Lexikalische Grundform* | *Stamm-form* |
|---|---|---|---|
| ABSORB | ABSORB<br>ABSORBED<br>ABSORBING<br>ABSORBS | ABSORB | ABSORB |
| | ABSORBER<br>ABSORBERS | ABSORBER | |
| ABSORBAB | ABSORBABLE<br>ABSORBABLY | ABSORBABLE | |
| ABSORBANC | ABSORBANCE<br>ABSORBANCES | ABSORBANCE | |
| | ABSORBANCY<br>ABSORBANCIES | ABSORBANCY | |
| ABSORBENT | ABSORBENT<br>ABSORBENTS<br>ABSORBENTLY | ABSORBENT | |
| ABSORPTION | ABSORPTION<br>ABSORPTIONS | ABSORPTION | |
| ABSORPTIV | ABSORBTIVELY<br>ABSORBTIVE | ABSORPTIVE | |

### *3.3.1.b Lexicon Based Approaches*

Problems of replacement rules

- provide no morphological information
- do not resolve ambiguities
- problematic to use in languages with strong changes especially
  - with prefixes (“**ge**laufen”)
  - if the stem is changed (“Fluß” “Flüsse”)
  - with many irregular forms
  - complicated (or irregular) rules for the separation of prefixes

Examples:

“er brachte den Brief mit” not “er mitbrachte den Brief” ”er überbrachte den Brief” not: “er brachte den Brief über”. Separated prefixes may change the meaning of a word: “Professorin Mayer schlug ihren Assistenten(für die Stelle vor)”

Some of these problems can be approached with lexicon based systems.

**Figure 10: Some of the Rules of the Kuhlen Algorithm**

| No | **Suffix** | **Replace** | **Condition** |
|---|---|---|---|
| 1 | IES | Y | |
| 2 | *xy*ES | *xy* | *xy* = *k*O, CH, SH, SS, ZZ or *x*X |
| 3 | *xy*S | *xy* | *xy* = *xk*, xE, *v*Y, *v*O, OA or EA |
| 4 | IES' | Y | |
| 5 | *x*ES' | *x* | |
| 6 | *x*S' | *x* | |
| 7 | *x*'S | *x* | |
| 8 | *x*' | *x* | |
| 9 | *xy*ING | *xy* | *xy* = *kk*, *xv*, *x*X |
| 10 | *xy*ING | *xy*E | *xy* = *vk* |
| 11 | IED | Y | |
| 12 | *xy*ED | *xy* | *xy* = *kk*, *xv*, *x*X |
| 13 | *xy*ED | *xy*E | *xy* = *vk* |

*The first column gives the pattern at the end of the word, the second the pattern that replaces it, if the condition in column three is satisfied. Capital letters stand for letters as they are found in the string. Lowercase x and y denote arbitrary but fixed letters, v denotes an arbitrary vowel, k an arbitrary consonant. For each rule there is a list of exceptions. The rules are applied top down until one matches*

**Figure 11: Application of the Kuhlen Rules**

| wordform: | Forms | generated | using | the | specified | rules |
|---|---|---|---|---|---|---|
| rule no.: | 3 | 13 | 10 | - | 11 | 3 |
| basic form: | Form | generate | use | the | specify | rule |

Idea:

- for each stem store the information necessary to construct every morphological form.
- for a given word try to find all stems that allow to construct the word as a morphological form
- offer all information available for that word (type of word, morphological form, meaning?)
- offer all possible stems of a word

Example: “Morphy” (Lezius 1995) uses the following steps to analyze a given string. It stops if a step is successful:

- check a small list of known high frequency forms for the string
- check the lexicon of stems for the string, removing recursively the last character. While checking change also the last “Umlaut” to its basic vowel and replace “ß” by “ss”. Check all stems to find ambiguous forms.

ferber@darmstadt.gmd.de

- Check for composites: Beginning at the end of the word identify recursively the longest part that can be constructed based on the lexicon. If the string can be decomposed into known parts in this way, assume that it is a composite.

**Figure 12: Morphological Analysis of "Flüssen" according to Lezius (1995)**

| Remove: | - | n | en | sen | ... |
|---|---|---|---|---|---|
| normal | Flüssen- | Flüsse-n | Flüss-en | Flüs-sen | ... |
| Umlaut | Flussen- | Flusse-n | Fluss-en | Flus-sen | ... |
| ß/ss | Flüßen- | Flüße-n | Flüß-en | Flü-ßen | ... |
| both | Flußen- | Fluße-n | **Fluß**-en | Flu-ßen | ... |

## 3.3.2: Classifications

Used to organize the objects of a domain into disjoint sets of related or similar objects.

Especially useful for physical objects:

- a book in a library has to have a unique place
- a picture in an exhibition is shown in only one place

### *Definition: Classification*

> *Let $D$ be a set of objects. A system of non empty, pair-wise disjoint subsets $K_1, K_2, ..., K_n$ with $D = K_1 \cup K_2 \cup ... \cup K_n$, $K_i \cap K_j = \emptyset \; \forall i, j \in \{1, ..., n\}, \; i \neq j$ is called a classification of $D$ into **classes** $K_1, K_2, ..., K_n$.*
>
> *A sequence of such systems is called a **strongly hierarchical classification system** if for each pair of a class from a system and its successor the class of the successor system is either a subset of its predecessor class or disjoint to it. This means that each class is divided into subclasses in the successor system.*

Hierarchical classification systems yield tree structures with varying levels of generalization

Besides strongly hierarchical classification system there are systems with weaker hierarchies, allowing an object to be in several classes or to have several generalizations.

Example: The seat of a car may be seen as a sitting device or as part of a car.

### *3.3.2.b Dewey Decimal Classification (DDC)*

Hierarchical classification system to organize all areas of knowledge.

The number of subclasses of a single class is restricted to 10 in each level of generalization.

Melvil Dewey (1851–1931): 1876 first edition.

Hierarchical classifications are build before objects (instances) are classified (they are **pre-coordinated**) This means that they offer little flexibility for new developments, little expressive power. Means to add more expressiveness: Add information that is not specific to a class using suffixes to the code of a class: 860=20: "Spanish and Portuguese literatures in English language" (860: Spanish and Portuguese literatures) 622.33(493): coal mining in Belgium (622.33: coal mining)

(Examples adapted from Manecke, 1997)

This allows to construct a suitable classification code when a object is classified (**post-coordination**).

**Figure 13: Top level classes of the DDC (according to http://www.oclc.org/oclc/fp/about/ddc21sm1.htm)**

| | |
|---|---|
| 0 | Generalities |
| 1 | Philosophy & psychology |
| 2 | Religion |
| 3 | Social sciences |
| 4 | Language |
| 5 | Natural sciences & mathematics |
| 6 | Technology (Applied sciences) |
| 7 | The arts, Fine and decorative arts |
| 8 | Literature & rethoric |
| 9 | Geography & history |

**Figure 14: Level 2 classes of the DDC (according to http://www.oclc.org/oclc/fp/about/ddc21sm2.htm)**

| | |
|---|---|
| 51 | Mathematics |
| 52 | Astronomy & allied sciences |
| 53 | Physics |
| 54 | Chemistry & allied sciences |
| 55 | Earth sciences |
| 56 | Paleontology Paleozoology |
| 57 | Life sciences biology |
| 58 | Plants |
| 59 | Animals |

### 3.3.3: Thesauri

A thesaurus describes words or terms of a specific domain / vocabulary and the relations between these words or terms.

Relations are not restricted to hierarchical relations. Examples are:

- synonym
- antonym
- related word
- more general term
- more specific term

ferber@darmstadt.gmd.de

**Figure 15: A path through the International Decimal Classification IDC (according to Manecke 1997)**

| | |
|---|---|
| 5 | Mathematik. Naturwissenschaften |
| 53 | Physik |
| 539 | Physikalischer Aufbau der Materie |
| 539.1 | Kernphysik. Atomphysik. Molekülphysik |
| 539.17 | Kernreaktionen |
| 539.172 | Individuelle Kernreaktionen |
| 539.172.1 | Kernreaktionen durch Atomkerne |
| 539.172.13 | Kernreaktionen durch Deuteronen |

**Figure 16: A path through the German version of the International Decimal Classification IDC (according to Fuhr 1995)**

| | |
|---|---|
| 3 | Sozialwissenschaften, Recht, Verwaltung |
| 33 | Volkswirtschaft |
| 336 | Finanzen. Bank- und Geldwesen |
| 336.7 | Geldwesen. Bankwesen. Börsenwesen |
| 336.76 | Börsenwesen. Geldmarkt. Kapitalmarkt |
| 336.763 | Wertpapiere. Effekten |
| 336.763.3 | Obligationen. Schuldverschreibungen |
| 336.763.31 | Allgemeines |
| 336.763.311 | Verzinsliche Schuldbriefe |
| 336.763.311.1 | Langfristige verzinsliche Schuldbriefe |

Further a thesaurus may define one or several meanings of a word.

In an IR system a thesaurus has an additional role: It defines a **controlled vocabulary** as subset of all words in the thesaurus that is used to index documents. Formal definition of indexing with a controlled vocabulary $T$: An attribute

$$t : D \rightarrow \mathfrak{P}(T)$$

is defined, that has the (set of subsets of the) controlled vocabulary as its range (set of values). The controlled vocabulary contains exactly one **descriptor** for each set of mutually synonymous words. All other words of such a “synonym set” have a “USE” relation pointing to this descriptor. Thus for indexing and retrieval an unique term is used for each subset of synonyms.The definition of the “synonym” relation controls how many details can be represented by means of a thesaurus.

Further the controlled vocabulary is structured hierarchically by the “more general” and “more specific” relations. These relations can be used to make queries more general and more specific.

#### *3.3.3.a Thesaurus construction*

To construct a thesaurus intellectually the following steps can be taken (c.f. Burkart 1997)

- definition of the overall properties: domain, specificity, type of language, size.
- selection of sources for words: ask users and experts, books and journals from the domain, other thesauri.
- terminology control: define the "synonym sets" for the thesaurus. This step determines how detailed the thesaurus will be, what can be distinguished and what cannot. Steps are:
  - collect synonyms and similar terms that are not to be distinguished as well as variations of a term with respect to spelling, abbreviations, styles, foreign words, regional dialects
  - distinguish meanings of ambiguous words: construct separate descriptors for each meaning of a string using unique synonyms, suffixes, scope notes.
  - control of composites: those that are too specific can be broken up into their (less specific) components.
- term relations: define relations between the "synonym sets" defined in the previous step. These relations include hierarchical relations and non hierarchical relations.

## 3.4: Vector Space Model

Boolean retrieval is based on set operations with terms or attribute values represented by strings.

In the last chapter it was discussed what "terms" or attributes can look like.

This chapter introduces more flexible use of such terms by weighting their influence according to their importance.

### 3.4.1: The Model

Representation of a document or a query: vector $\in \mathbb{R}^n$

This means that methods from the vector space can be used like:

- metric (-> distance, similarity)
- calculus

***Definition: Vector Space Model of IR:***

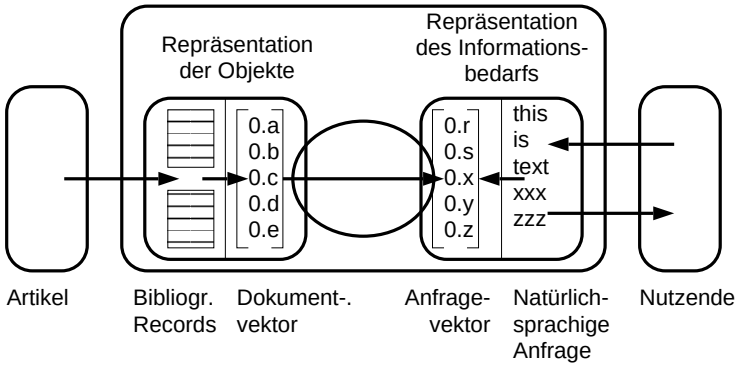
> *Let $D = \{d_1, ..., d_m\}$ be a set of documents and $A = \{A_1, ..., A_n\}$ be a set of attributes. A **document vector** $w_i = (w_{i,1}, ..., w_{i,n}) \in \mathbb{R}^n$ for a document $d_i \in D$ is defined by a set of **weights** $\{w_{i,k} \in \mathbb{R},\ k = 1, ..., n\}$. In the same way a **query vector** $q = (q_1, ..., q_n) \in \mathbb{R}^n$ is defined for a query.*
>
> *If further a similarity measure $s : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ is given that assigns a real value to every pair of vectors, the whole system is called a **vector space model of IR**.*

In text retrieval the attributes are in general defined by the occurrence of terms in the text. In this case the weight $w_{i,k} \in \mathbb{R}$ describes the importance of term $t_k \in T$ for the document $d_i \in D$. In the same way the weight $q_k \in \mathbb{R}$ describes the importance of term $t_k \in T$ for the query.

The document vector could in general as well be defined directly by real valued attributes: $A_k : D \rightarrow \mathbb{R}$ For simplicity reasons and to be consistent with most of the literature we will assume for the future $w_{i,k} = A_k(d_i)$:

The similarity measure can be used to compare document and query vectors i. e. find the most similar documents for a query.

**Figure 17: Vektor Space Model of Text Retrieval**




*The human generated document are represented by document vectors*

## 3.4.2: Relation to Boolean Retrieval

Attributes: $A_i : D \rightarrow \{0,1\}$

$$A_i(d) = \begin{cases} 1 & if\ t_i\ occurs\ in\ d \\ 0 & otherwise \end{cases}$$

Query $q = (q_1, ..., q_n)$

$$q_i = \begin{cases} 1 & if\ t_i\ occurs\ in\ the\ query \\ 0 & otherwise \end{cases}$$

If all terms in the query are connected by AND: a document is in the result set, if

$$A_i(d) = 1\ \forall i \in \{1, ..., n\}\ with\ q_i = 1$$

If all terms are connected by OR: a document is in the result set, if

$$\exists\ i \in \{1, ..., n\}\ with\ q_i = A_i(d) = 1$$

This result can be expressed using the inner product of two vectors:

***Definition: Inner Product:***

*For* $w_i = (w_{i,1}, ..., w_{i,n}) \in \mathbb{R}^n$ *and* $q = (q_1, ..., q_n) \in \mathbb{R}^n$ *the inner product is defined as*

$$w_i \cdot q = \sum_{k=1}^{n} w_{i,k} q_k$$

*Hence the inner product is a similarity measure* $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$

For vectors containing only 0 and 1 the inner product counts the number of positions in which both vectors have a 1.

If all terms in the query are connected by AND: a document is in the result set, if

$$w_i \cdot q = \sum q_i$$

If all terms are connected by OR: a document is in the result set, if

$$w_i \cdot q \geq 1$$

$\longrightarrow$ ANDed queries select only those documents that are most similar to the query
$\longrightarrow$ ORed queries select all documents that have a similarity larger than zero
$\longrightarrow$ the similarity can be used to deliver a ranked list of documents

## 3.4.3: Term Weighting

Document vectors have been invented to give terms weights according to their importance for a document. Issues are:

- ability to describe the content of a document
- ability distinguish the document from other documents

Methods:

- Intellectual weighting
- automated weighting

Intellectual methods are expensive and not very reliable.

Two kinds of influence can be distinguished in weighting methods: local or context sensitive influences and global or context insensitive influences

### *3.4.3.a Local Weighting Strategies*

Term frequency

Number of appearances of a term in a document.

Rationale: the main topic of a document should cover most of its text. In this text important terms should be used frequently.

Method:

$$w_{i,j} = h(i,j)$$

$$w_{i,j} = \frac{h(i,j)}{1 + h(i,j)}$$

$$w_{i,j} = K + (1-K)\frac{h(i,j)}{\max\limits_{l \in \{1,\ldots n\}} h(i,l)} \;\; if \; h(i,j) > 0$$

with $h(i,j)$ denoting the frequency of term $t_j$ in document $d_i$ and $K \in [0,1]$

Using document structure

Terms can be weighted according to the part of document they occur in.

Terms from the title or the free keyword section should be more important than terms from the body of an article.

### *3.4.3.b Word Frequencies in Language*

Zipfs Law

> *For a text corpus $C$ let $W(C)$ be the set of words occurring in $Ch(w)$ denote the frequency of the word $w \in W(C)$ in the corpus. $r(w)$ denote the rank of $w \in W(C)$ if the words are ranked according to decreasing frequencies. It holds*

$$r(w) \cdot h(w) \sim c \; = \; \textit{constant} \; \forall w \in W(C)$$

### *3.4.3.c Global Weighting Strategies*

Most global strategies use term statistics to determine the usefulness of terms for retrieval:

- very frequent terms tend to appear in many documents and are therefore not well suited to select specific documents
- very rare terms tend to appear in only very few documents and are therefore not well suited to find all relevant documents.

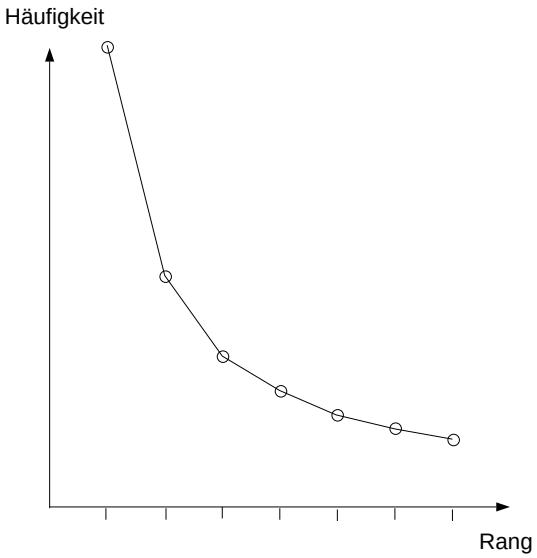
© Ferber GMD-IPSI
ferber@darmstadt.gmd.de

**Figure 18: Zipfs Law applied to the Brown- and LOB-Korpus**

| Rank | Frequency | R*f/100000 | Term |
|---|---|---|---|
| 1 | 138323 | 1.3832 | the |
| 2 | 72159 | 1.4432 | of |
| 3 | 56750 | 1.7025 | and |
| 4 | 52941 | 2.1176 | to |
| 5 | 46523 | 2.3262 | a |
| 6 | 42603 | 2.5562 | in |
| 7 | 22177 | 1.5524 | that |
| 8 | 21210 | 1.6968 | is |
| 9 | 20501 | 1.8451 | was |
| 10 | 19587 | 1.9587 | it |
| 100 | 2043 | 2.0430 | years |
| 500 | 394 | 1.9700 | program |
| 1000 | 207 | 2.0700 | jones |
| 2000 | 105 | 2.1000 | granted |
| 3000 | 67 | 2.0100 | agencies |
| 4000 | 47 | 1.8800 | embassy |
| 5000 | 36 | 1.8000 | vale |
| 10000 | 14 | 1.4000 | poisoning |
| 12034 | 11 | 1.3237 | yell |

```
Minimum: 1.24982
Maximum:  2.55618
Mean:     1.697
Variance: 0.077
Standard deviation: 0.277
```

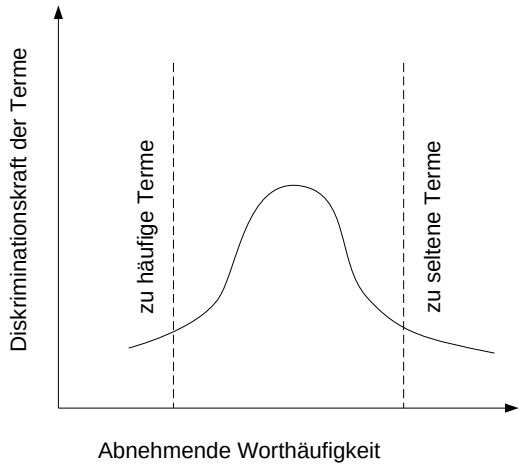
**Figure 19: Qualitative View of Zipfs Law**




Inverted document frequency

To handle the frequent terms: give terms that occur in many documents low weights. The **document frequency** $d(j)$ of term $t_j$ is defined as number of documents a term occurs in.

**Figure 20: Discrimination Power vs. Frequency (from Salton & McGill 1983)**




Method: **Inverted document frequency** (IDF);

$$idf(j) = \frac{1}{d(j)}$$

or related forms:

$$w_{i,j} = \ln\left(\frac{m}{d(j)}\right)$$

$$w_{i,j} = \ln\left(\frac{m - d(j)}{d(j)}\right)$$

where $m$ denotes again the total number of documents in the collection.
Global and local strategies can be combined:

$$\frac{h(i,j)}{d(j)}$$

$$\widetilde{w_{i,j}} = \frac{1}{2}\left(1 + \frac{h(i,j)}{\max\limits_{k \in \{1,..,n\}}\{h(i,k)\}}\right) \ln\left(\frac{m}{d(j)}\right)$$

respectively the normalized version:

$$w_{i,j} = \frac{\widetilde{w_{i,j}}}{\sqrt{\sum\limits_{k=1}^{n} \widetilde{w_{i,k}}^2}}$$

© Ferber GMD-IPSI
ferber@darmstadt.gmd.de

## 3.4.4: Relevance Feedback (Rocchio)

To obtain weights for queries, global methods can be used. L ocal methods are not applicable because most queries are too short. In dialogue settings documents found with a query can be used to construct weights for a new query: users can be asked how useful these documents are.

The user is asked to give **relevance judgements** for the documents in a result set $D_q = \{d_1, ..., d_n\}$ obtained with a query with the query vector $q$. This provides two subsets: $R = \{d_1^+, ..., d_r^+\}$, the set of documents judged relevant and $U = \{d_1^-, ..., d_u^-\}$, the set of documents judged not relevant. A new query vector is constructed based on the document vectors $v_d$ of a document $d$:

$$q' = \alpha q + \beta\left(\frac{1}{r}\sum_{d\in R} v_d\right) - \gamma\left(\frac{1}{u}\sum_{d\in U} v_d\right)$$

with real parameters $\alpha$, $\beta$, and $\gamma$.

## 3.4.5: The SMART System

SMART is one of the best known vector space IR systems.It has been developed for about 30 years by Gerard Salton and his co-workers at Cornell University. It is rather an experimental framework than a single system. It uses automated indexing and relevance feedback among other features.

### *3.4.5.a Automated Indexing*

as described in Salton and McGill (1983)

- Decompose the text into words
- remove stop words
- use a replacement rule based stemmer to generate terms
- these terms are weighted or replaced according to the following steps:
  - for terms with medium document frequency use a weighting scheme like

$$w_{i,k} = \frac{h(i,k)}{d(k)}$$

    or the one given before
  - terms with very high document frequencies are replaced by term pairs built with the other terms in a neighborhood of given size. Weights are constructed based on the frequencies of the two terms of a pair.
  - terms with very low document frequencies are replaced by more general terms from a thesaurus or by groups of related terms

## 3.4.6: Similarity Measures

### *3.4.6.a Inner Product*

The inner product sums the products of the vector entries. Vectors with many non zero entries have a higher probability to achieve high values. Vectors with larger entries get higher values This means that longer documents have a higher probability to get high similarity values.

This leads to weak experimental results.

### 3.4.6.b Cosine Measure

$$cos(w_i, q) = \frac{\sum_{k=1}^{n} w_{i,k} q_k}{\sqrt{\sum_{k=1}^{n} w_{i,k}^2} \sqrt{\sum_{k=1}^{n} q_k^2}}$$

The cosine measure is an attempt to avoid the problems of the inner productIt takes values between $-1$ and $1$. It is insensitive to the length of the vectors and can be interpreted as the inner product of the normalized vectors

$$\frac{w_i}{\sqrt{\sum_{k=1}^{n} w_{i,k}^2}} \quad \text{and} \quad \frac{q}{\sqrt{\sum_{k=1}^{n} q_k^2}}$$

documents are most similar if their vectors have the same directions

### 3.4.6.c Other Similarity Measures

Pseudo cosine:

$$s_p(w_i, q) = \frac{\sum_{k=1}^{n} w_{i,k} q_k}{\left(\sum_{k=1}^{n} w_{i,k}\right)\left(\sum_{k=1}^{n} q_k\right)}$$

Dice:

$$s_d(w_i, q) = \frac{2 \sum_{k=1}^{n} w_{i,k} q_k}{\sum_{k=1}^{n} w_{i,k} + \sum_{k=1}^{n} q_k}$$

Overlap:

$$s_o(w_i, q) = \frac{\sum_{k=1}^{n} \min\left(w_{i,k}, q_k\right)}{\min\left(\sum_{k=1}^{n} w_{i,k}, \sum_{k=1}^{n} q_k\right)}$$

Jaccard:

$$s_J(w_i, q) = \frac{\sum_{k=1}^{n} w_{i,k} q_k}{\sum_{k=1}^{n} w_{i,k} + \sum_{k=1}^{n} q_k - \sum_{k=1}^{n} w_{i,k} q_k}$$

A description of these measures is given in Jones and Furnas (1987).

ferber@darmstadt.gmd.de

## 3.5: Evaluation of IR Systems

The goal of the evaluation of a IR system should be to determine how useful a system or a specific method within a system is to solve the information problems of users. A sound experimental evaluation of this question would need representative samples of problems and users. Within experiments many potentially influencing factors should be examined including

- collection characteristics (selection strategies, document formats, ...)
- indexing and representation formats (manual indexing, use of a controlled vocabulary, classifications, ...)
- search methods and tools
- presentation of results
- interaction strategies and interface design

Such an approach is not feasible. Therefor most evaluations are restricted to a limited but central sub task.

Given:

- a collection of documents
- a set of queries
- for each query the subset of relevant documents

the goal is to find as many relevant documents as possible.

### 3.5.1: Measures

A more detailed description can be given in the following way:

***Definition: Relevance***

*Let $D = \{d_1, ..., d_m\}$ be a set of documents and $Q$ a set of queries. The relevance of a document $d_i$ for a query $q$ is defined by a relation*

$$r : D \times Q \rightarrow U$$

*where $U$ denotes a set of "relevance values" in most cases being $U = \{0, 1\}$.*

According to this definition relevance depends only on the query and the document. Other influences like the documents already seen or the knowledge of the user cannot be taken into account.

***Definition: Precision and Recall***

*Let $D = \{d_1, ..., d_m\}$ be a set of documents, and $D_q$ be the sub set of documents found in $D$ for the query $q \in Q$. Let further $R_q = \{d \in D \mid r(d, q) = 1\}$ denote the set of documents relevant for $q$.*

$$P(q, D) := \frac{\mid D_q \cap R_q \mid}{\mid D_q \mid}$$

*is called* ***Precision*** *and*

$$R(q, D) := \frac{\mid D_q \cap R_q \mid}{\mid R_q \mid}$$

***Recall*** *of the result set $D_q$*

Precision gives the fraction of relevant documents within the retrieved documents, recall gives the fraction of relevant documents that were retrieved.

Best values for precision are obtained if all retrieved documents are relevant, for recall if all relevant documents were retrieved.

Extreme values: one single relevant document: precision = 1; All documents: recall = 1.

Precision and recall are antagonistic: A small result set from a specific query will result in high precision and low recall; a large result set from a very general query will result in low precision an high recall.

Mean Precision and Recall values can be calculated for a set of queries in two ways: user oriented

$$P_u(D) := \frac{1}{N}\sum_{i=1}^{N}\frac{\mid D_{q_i} \cap R_{q_i} \mid}{\mid D_{q_i} \mid}$$

$$R_u(D) := \frac{1}{N}\sum_{i=1}^{N}\frac{\mid D_{q_i} \cap R_{q_i} \mid}{\mid R_{q_i} \mid}$$

or system oriented

$$P_s(D) := \frac{\sum_{i=1}^{N} \mid D_{q_i} \cap R_{q_i} \mid}{\sum_{i=1}^{N} \mid D_{q_i} \mid}$$

$$R_s(D) := \frac{\sum_{i=1}^{N} \mid D_{q_i} \cap R_{q_i} \mid}{\sum_{i=1}^{N} \mid R_{q_i} \mid}$$

To compare two systems both measures have to be taken into account. Only if one has higher values for Precision and Recall, it is “better” than the other.

In case of ranked result sets this antagonism can be displayed:

### *Definition: Precision-Recall-Diagram*

*Let* $D_q = (d_{s_1}, ..., d_{s_k})$ *be a completely ordered result set and* $R_q = \{d \in D \mid r(d,q) = 1\}$ *the set of relevant documents for the query* $q$*. Let further* $(d_{t_1}, ..., d_{t_l})$ *be the intersection* $D_q \cap R_q$ *ordered according to* $D_q$*. The sequence* $(R_i(q,D), P_i(q,D))_{i=1,...,l}$ *with*

$$R_i(q,D) := \frac{\mid (d_{t_1}, ..., d_{t_i}) \mid}{\mid R_q \mid}$$

*and*

$$P_i(q,D) := \frac{\mid (d_{t_1}, ..., d_{t_i}) \mid}{\mid (d_{s_1}, ..., d_{s_j}) \mid}$$

*with* $d_{s_j} = d_{t_i}$ *is called the* ***Precision-Recall-Diagram*** *of* $q$*.*

*It can be displayed by points in the square* $[0,1]^2$*.*

Comparing two systems: A system is “better” if its precision values are higher at all recall levels.

One dimensional comparison of systems with ranked results:

- Mean precision values at fixed recall levels
- Break even point: Value where Recall=Precision (not always uniquely defined!)

**Figure 21: A Precision Recall Diagram**

The following sequence represents a completely ordered result set following the lines. R denotes a relevant document U a document that is not relevant.

```
RURRURRRUU URRURRRUUU URUUUURUUU UUUUURUUUU
RURUUUUURU UUURUUUURU UUUUURUUUU UUUUURUUUU
UUUUUUUURU UUUUUUUURU UUUUUUUURU UUUUUUUUUU
UUUUURUUUU UUUUUUURUU UUUUUURUUU UUUUUUURUU
UUUUUUUUUU RUUUUUUUUU UUUUUURUUU UU...
```

The sequence $(R_i(q, D), P_i(q, D))_{i=1,\ldots,30}$ looks like this:

$$\left(\frac{1}{30}, \frac{1}{1}\right), \left(\frac{2}{30}, \frac{2}{3}\right), \left(\frac{3}{30}, \frac{3}{4}\right), \left(\frac{4}{30}, \frac{4}{6}\right), \left(\frac{5}{30}, \frac{5}{7}\right)$$

$$\left(\frac{6}{30}, \frac{6}{8}\right), \left(\frac{7}{30}, \frac{7}{12}\right), \left(\frac{8}{30}, \frac{8}{13}\right), \left(\frac{9}{30}, \frac{9}{15}\right), \left(\frac{10}{30}, \frac{10}{16}\right)$$

$$\left(\frac{11}{30}, \frac{11}{17}\right), \left(\frac{12}{30}, \frac{12}{22}\right), \left(\frac{13}{30}, \frac{13}{27}\right), \left(\frac{14}{30}, \frac{14}{36}\right), \left(\frac{15}{30}, \frac{15}{41}\right)$$

$$\left(\frac{16}{30}, \frac{16}{43}\right), \left(\frac{17}{30}, \frac{17}{49}\right), \left(\frac{18}{30}, \frac{18}{54}\right), \left(\frac{19}{30}, \frac{19}{59}\right), \left(\frac{20}{30}, \frac{20}{66}\right)$$

$$\left(\frac{21}{30}, \frac{21}{76}\right), \left(\frac{22}{30}, \frac{22}{89}\right), \left(\frac{23}{30}, \frac{23}{99}\right), \left(\frac{24}{30}, \frac{24}{109}\right), \left(\frac{25}{30}, \frac{25}{126}\right)$$

$$\left(\frac{26}{30}, \frac{26}{138}\right), \left(\frac{27}{30}, \frac{27}{147}\right), \left(\frac{28}{30}, \frac{28}{158}\right), \left(\frac{29}{30}, \frac{29}{171}\right), \left(\frac{30}{30}, \frac{30}{187}\right)$$

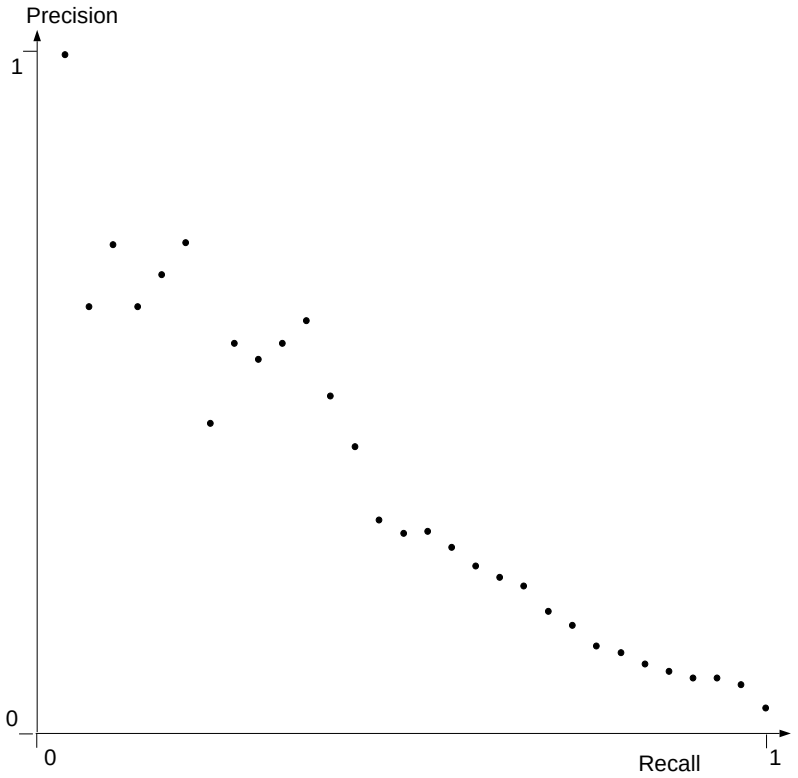
## 3.5.2: Test Collections

The relevance relation cannot be generated automatically. It has to be determined by humans knowing the domain. This means that for every query all documents of a collection have to be known or inspected. For large collections this is very expensive. Therefor collections with queries and relevance data that had once been generated were often reused.

Test collections have the advantages to be available and to offer comparable test conditions. But there are also some p roblems:

- the collections are small compared to real collections
- it is often not clear how they were generated
- some collections are quite old
- only few collections have full texts
- different collections yield differing results for the same systems
- systems can be optimized for a single collection
- collections might be selected for evaluation that yield best results.

**Figure 22: Precision – Recall – Diagram Displayed in the Plane**




## 3.5.3: The TREC Experiments

Since 1992: Seven Text REtrieval Conferences (http://trec.nist.gov/).

The goal of the conference series is to encourage research in information retrieval from large text applications by providing a large test collection, uniform scoring procedures, and a forum for organizations interested in comparing their results.

Material provided for preparation:

- training data: 2 Gigabyte of text (> 1 mio docs)
- training queries (topics)
- relevance judgements for the queries.

Material provided for testing:

- collection of new documents (~ 1 GB of text)
- 50 new queries

Participants can adapt their systems to the material provided for preparation. With the new material for testing two tasks are given:

- Ad hoc: search the old collection with new queries
- Routing: distribute the new documents according to the old topics (queries)

Participants provide ranked lists of 1000 hits per topic. Evaluation is done centrally at NIST (National Institute of Standards & Technology, USA). Measures are Precision Recall Diagrams and the mean precision over a fixed set of recall values.

**Figure 23: Test Collections (according to Griffiths Luckhurst & Willett 1986 and Dumais, 1991)**

| Name | Domain | Content | 4. Documents | 5. Terms / Doc. | 6. Queries | 7. Terms / Query | 8. Rel./ Query |
|---|---|---|---|---|---|---|---|
| Keen | Librarianship, Information Science | title, Indexterms (manual) | 800 | 9,8 | 63 | 10,3 | 14,9 |
| Cranfield | Aerodynamics | Indexterms (manual) | 1400 | 28,7 | 225 | 8,0 | 7,2 |
| Evans | INSPEC | title | 2542 | 6,6 | 39 | 27,5 | 23,1 |
| Harding* | INSPEC | title, abstracts | 2472 | 36,3 | 65 | 32,4 | 22,6 |
| LISA** | Library and Information Abstracts Database 1982 | title, abstracts | 6004 | 39,7 | 35 | 16,5 | 10.8 |
| INSPEC | INSPEC | title, abstracts | 12684 | 36,0 | 77 | 17,9 | 33,0 |
| UCKIS | Chemical Abstract Service | title | 27361 | 6,7 | 182 | 7,4 | 58,9*** |
| MED | Biomedicine | (title?) abstracts | 1033 | 50 | 30 | 10 | 23 |
| CISI | Library , Information Science | (title?) abstracts | 1460 | 45 | 35 | 8 | 50 |
| TIME | World news (Time Magazine) | articles | 425 | 190 | 83 | 8 | 4 |
| ADI | Library science | abstracts | 82 | 16 | 35 | 5 | 5 |

*Columns: 4 number of documents; 5: mean number of terms in a document; 6: number of queries; 7: mean number of queries; 8: mean number of relevant documents per query.*

** Harding is a subset of Evans, enriched by abstracts and additional queries. ** Relevance judgement for LISA were obtained from manual searches in the printed version of the data base, supplemented in some cases by online or exhaustive manual searches. *** UCKIS suffers from a lack of exhaustive relevance judgements.*

### *3.5.3.a Relevance Judgements*

Relevance judgements are made after the experiments using a pooling method: The top 100 documents of all lists submitted for a query are taken together to form a “document pool”. This pool is intellectually checked for relevant documents by an expert. This method was used to reduce the number of documents to be checked. It assumes that most relevant documents are ranked within the top 100 documents by at least one of the retrieval systems. This is realistic only for a large number of participating systems. If the result lists are very similar, the probability that a relevant document is missed by the pooling method increases.

The quality of the pooling method for relevance judgement was tested in TREC 3: In addition to the pool with 100 top ranked documents a second pool with the 200 top ranked

documents was used for relevance judgement. Whereas in the first pool on average 146 relevant documents were found, this number was 196 for the second pool. This shows, that a remarkable number of relevant documents was probably missed. (More detailed results are given in fig 24.) This leads to an overestimation of recall and a underestimation of precision. But this is true for all participating systems.

**Figure 24: Size of Document Pools Used for Relevance Assessment (from 1995–WWW, 1996–WWW)**

| | Adhoc | | | Routing | | |
|---|---|---|---|---|---|---|
| | max. possible | actual | relevant | max. possible | actual | relevant |
| TREC-1 | 3300 | 1279 (39%) | 277 (22%) | 2200 | 1067 (49%) | 371 (35%) |
| TREC-2 | 4000 | 1106 (28%) | 210 (19%) | 4000 | 1466 (37%) | 210 (14%) |
| TREC-3 100 | 2700 | 1005 (37%) | 146 (15%) | 2300 | 703 (31%) | 146 (21%) |
| TREC-3 200 | 5400 | 1946 (28%) | 196 (10%) | 4600 | 1333 (35%) | 187 (14%) |
| TREC-4 | 4000 | 1345 (34%) | 115 (8.5%) | 2600 | 930 (35%) | 131 (14%) |

*The columns within the Adhoc and Routing blocks contain the following information: 1: largest possible number of documents in the pool (in case non of the result lists shares a document in the top ranked documents with another result list); 2: the average of the actual size of the pools and the percentage of the maximal possible size; 3: the average number of relevant documents found in the pool and the percentage of this within the actual pool size.*

### *3.5.3.b Tracks*

Beginning with TREC 5 several sub tasks (called tracks) have been included in the TREC experiments:

- the **Confusion** track uses documents degraded by optical character recognition (OCR)
- the **Database Merging** track uses several collections. Problems are: selection of appropriate databases and merging of ranked result sets from various collections
- in the **Filtering** track participants have to deliver unordered result sets. A specific cost function is used for evaluation.
- in the **Interactive** track real user interaction is allowed
- the **Multilingual** track uses documents in other languages than English
- in the **Cross Lingual** track documents in one language are being searched with a query in a different language
- the **NLP** track is dedicated to systems using natural language processing techniques

## 3.6: Advanced Vector Space Systems

The first two TREC experiments were mainly used to adapt the systems to the large amount of data. In TREC 3 and 4 several ideas and methods were used by several systems.

### 3.6.1: Pseudo Relevance Feedback

Idea: use documents found in a first run to enhance the query for a second run. This idea is based on the aassumption that the top ranked documents are likely to be relevant i. e. that the system already performs quite well. In contrast to the Rocchio relevance feedback most systems did not use the complete document vectors to modify the query, but they selected a limited number of terms occurring most frequently in the top ranked documents to enhance the query.

In TREC 3 SMART for example used the following procedure: After a first query operation the top 30 documents of the ranked list were used for feedback. All terms from these documents were ordered according to their frequency in the top 30 documents. The 500 most frequent terms were selected, the 30 document vectors were restricted to these 500 terms and added to the query vector according to the Rocchio formula with parameters (8, 8, 0).

Pseudo Relevance Feedback was quite successful

### 3.6.2: Pairs of Terms

Many systems used pairs of terms for indexing. The idea is that such pairs should be more specific then single terms. For example SMART allowed in TREC 3 term pairs for indexing if they occurred in more than 25 documents. In the pseudo relevance feedback step the 10 most frequent term pairs were used in the same way as the 500 most frequent single terms.

The “INQUERY” system extracted “phrases” of two or three words from the documents of a large sub collection and added the most similar phrases to the query.

The use of term pairs seems to have only little or even no positive effect on the results. In TREC 4 several systems reduced the number of pairs used in their queries.

### 3.6.3: Passage Retrieval

Several systems tried methods that divide the documents in blocks or overlapping windows of fixed or limited size to calculate similarities and to do pseudo relevance feedback. The idea behind this approach is that within longer documents specific topics are dealt with in subsections. Similarity measures that are based on subsections of documents should easier find these topics.

For example in SMART the documents were divided into overlapping blocks of 200 terms. The similarity measure was defined as follows:

$$s(w_i, q) = w_i \cdot q \left( 1 + 2 \frac{\max_{b \in B_d} (b \cdot q)}{\max_{b \in B_D} (b \cdot q)} \right)$$

with $B_d$ denoting the vectors of blocks of the document $d \in D$. $B_D$ denotes the vectors generated from all Blocks/ from documents of $D$

The use of passage retrieval did in general not yield the expected success.

## 3.6.4: Similarity Measures

Many groups tried to optimize their similarity measures. Often these measures are very complicated using many parameters. Two basic methods will be shown

### *3.6.4.a Adapted Cosine*

The cosine is insensitive to the length of the document.

For the results of TREC 3 the relative frequencies to be judged relevant and the relative frequency to get a high ranking by the cosine similarity measure were compared. It turned out that short documents are ranked too high compared to the relevance judgement while long documents are ranked to low compared to the relevant judgement. To compensate this effect the similarity measure was changed in such a way that these differences vanished (for details see: Singhal, Buckley & Mitra 1996)

### *3.6.4.b Robertsons-Spark Jones Formula*

This formula uses relevance feedback data in a different way to determine a query vector:

$$v_k = \ln \frac{(R(q,k)+0.5)/(R(q)-R(q,k)+0.5)}{(d(k)-R(q,k)+0.5)/(N-d(k)-R(q)+R(q,k)+0.5)}$$

with $N$ being the number of documents in the collection, $R(q)$ the number of documents judged relevant for $qd(k)$ the number of documents that contain term $t_k$ and$R(q,k)$ the number of relevant documents that contain term $t_k$

Details can be found in Robertson, Walker, Hancock-Beaulieu and Gatford TREC 3; Robertson, Walker, Beaulieu, Gatford and Payne TREC 4.

To compute the weight of term $t_k$this formula distinguishes the set of documents according to two criteria: If a document contains term $t_k$, and if it is judged relevant for the query.

The nominator deals with the relevant documents comparing those containing the term with those that do not. The denominator does the similar comparison for the documents that are not relevant.

**Figure 25: Number of Documents in a Collection when Classified According to the two Criteria: "Contains a term $t_k$" and "Is Relevant to Query"**

| | contains term | does not contain term |
|---|---|---|
| relevant | $R(q,k)$ | $R(q)-R(q,k)$ |
| not relevant | $d(k)-R(q,k)$ | $N-d(k)-R(q)+R(q,k)$ |

*The cells of the table give the number of documents satisfying both conditions with the notations from the definition of the Robertson-Spart Jones formula. The numbers can be found in the quotient of the numerator and the denominator in the formula.*

- ⟶ a term that occurs in most relevant documents and does occur in few non relevant documents gets a high value for the nominator and a low value for the denominator, i. e. altogether a high weight
- ⟶ If the term also occurs in many non relevant documents the denominator will increase leading to a smaller weight.
- ⟶ If the term is contained in only a few relevant documents but in many non-relevant documents the nominator will be small and the denominator will be big leading to a small weight.

## 3.7: Advanced Models

### 3.7.1: Inference Networks

An inference network consists of

- a directed graph with
  - nodes representing content (assertions) and
  - edges representing relations (dependencies) between nodes;
- nodes
  - can take an activation value (between 0 and 1)
  - have a function that takes the activation of nodes connected by an edge as input to compute a new activation value.
- To generate a new pattern of activity values, the function of all nodes are applied in parallel.
- This can be repeated until the patterns do no longer change.

To apply the model in IR Turtle and Croft (1990, 1991) use a network consisting of two parts. The "document network" consists of three layers:

- a document layer
- a layer of text representations
- a layer of concept representations

Edges are only leading from one layer to the next one. The document network is determined by the documents of the collection. The layers represent various levels of abstraction. All paths are starting at the document nodes and lead to the concept nodes.

The second part of the inference network is the query network. It is connected to the layer of concept representations and may consist of several layers detailing various levels of abstraction of a query. All paths in this part of the graph lead to a single node, that represents the importance of a document for the specified query. To obtain this value the node of the document is activated and this activation is propagated through the network until it reaches the last node of the query network. During this propagation several paths of "evidence" contribute to the final value.

The actual implementation in the IR system INQUERY is much simpler. It consists of only one layer representing terms, i. e. a vector space model with a sophisticated similarity measure implemented by an inverted list.

### 3.7.2: Co-occurrence Based Methods

The weighting strategies for terms in a document vector are applied to single words. Dependencies between words were not taken into account for the selection of index or query terms. But the occurrences of a term in a document is not independent of the occurrence of other terms in the same document. These dependencies can be used to select index and query terms.

The associative model assumes that the meanings of terms occurring frequently together in documents are related. This principle is a rather old one:

> *"objects once experienced together tend to become associated in the imagination, so that when any one of them is thought of, the others are likely to be thought of also"*

(James, 1890, Vol 1, page 561)

Several approaches have used similarities between terms based on their co-occurrence in documents to

- model human associations and memory
- construct associative thesauri based on the similarity of terms
- use these thesauri for automated indexing
- use these thesauri for query expansion

**Figure 26: Associations Automatically Generated from the Lob– and the Brown Corpus Using Co-occurrences of Terms**

| | | |
|---|---|---|
| tax: | fruit: | sin: |
| income: 71.81 | eggs: 56.69 | crime: 107.11 |
| fiscal: 66.96 | meat: 56.69 | doctrine: 98.31 |
| taxes: 61.99 | foods: 55.09 | morality: 92.00 |
| profits: 56.67 | fresh: 54.99 | adam: 87.57 |
| revenue: 56.35 | seed: 52.16 | christ: 57.57 |
| sales: 42.85 | sugar: 42.99 | jesus: 54.83 |
| reduction: 41.33 | milk: 40.12 | suffering: 52.18 |
| file: 39.18 | meal: 32.20 | flesh: 50.89 |
| paying: 35.93 | tree: 31.55 | original: 47.89 |
| payments: 32.81 | believes: 31.42 | born: 45.49 |
| collection: 31.20 | soft: 31.30 | burden: 39.32 |
| towns: 30.29 | tea: 30.09 | consequently: 36.19 |
| estimated: 29.64 | expenditure: 29.98 | heaven: 36.19 |
| finance: 29.06 | wine: 29.64 | god: 35.91 |
| net: 28.81 | fish: 26.98 | creation: 34.78 |
| uniform: 27.23 | breakfast: 26.89 | requires: 34.35 |
| corporation: 26.70 | containing: 26.08 | grace: 30.84 |
| purchase: 26.37 | eat: 25.82 | death: 30.78 |
| spending: 25.13 | referred: 25.08 | moral: 30.70 |
| excess: 24.21 | parks: 23.29 | darkness: 30.16 |

*These associations were generated from the terms that occur with a frequency between 100 and 3100 in the corpus. If the occurrence of terms were independent of each other, there should be no similarity between the terms in one column*

Similarity of terms can be gained from the document vectors of a collection: they form a so called **Term-Document-Matrix**

$$W = \{w_{i,j}\}_{i=1,..,m;\, j=1,..,n}$$

This matrix can be multiplied by its transposed leading to a $n \times n$-matrix $W^t \cdot W$ called the **Term-Term-Matrix**. The entries in this matrix are the inner products of vectors that are composed of the weights that the respective term is given in the documents of the collection; i. e. “term vectors” that describe the meaning of a term by the documents it occurs in. The inner product of the term vectors of two terms cn be used as a similarity value for the two terms.

### *3.7.2.a Associative Indexing and Query Expansion*

In an experimental setting we used co-occurrence data for automated indexing (Ferber 1997): A corpus of bibliographic records intellectually indexed with the OECD thesaurus was used to extract similarities between the words of the titles and the descriptors of the thesaurus. Two sets of records were kept aside: one to optimize the parameters of the system, the other to test it. The word - descriptor similarities were used to predict the intelectually given descriptors of the test records.

### *3.7.2.b Cross Language Retrieval*

A “parallel” corpus of documents in two different languages can be used to search for documents in one language using a query formulated in the other language: To this end documents in the two languages with the same content are used to compute similarity values between the terms of the two languages. These similarities are used to expand the query with terms of the other language. These new terms can be used to search in the collection.

Sheridan and Ballerini (1996) used 93 229 Italian news stories of the Swiss News Agency indexed by time, location, and a content category (out of 50 possible classes). From the German service of the same agency 10 293 articles could be identified with the same indexing. Based on these pairs of identically indexed articles the similarity values for the occurring terms were computed and used to expand German queries. The results were compared with results obtained with intelectually generated translations of the queries.

**Figure 27: Results of the Study on Cross Language Retrieval Using Similarity Measures Generated from Parallel Corpora. (from Sheridan and Ballerini, 1996)**

| | | Relevant Docs Found | Mean Precision |
|---|---|---|---|
| Cross Lingual: | Number of Query Terms | | |
| | 10 | 525 | 0.212 |
| | 25 | 694 | 0.278 |
| | 50 | 638 | 0.275 |
| Italian only: | Stemming | | |
| | no | 488 | 0.231 |
| | yes | 898 | 0.527 |

## 3.8: Meta Data

IR systems can use information about documents like bibliographic data, classifications, or access information that are not really part of the document. Such data are called **Meta Data**. The destinction between data and Meta Data is often fuzzy: in some cases the bibliographic information is printed on the title page of articles, or key terms and classification are given in printed versions.

Meta Data can be included in documents encoded in appropriate structured formats like SGML, or they can be provided separately like done in reference databases or by web search engines. For resource discovery Meta Data should be provided in machine readable form. There are many complex formats for various domains and types of documents. The diversity and complexity of these make it difficult to handle them in a uniform way.

### 3.8.1: Dublin Core

The “Dublin Core” initiative is an approach to unify the storage of Meta Data for digital document like objects. It is named after the first meeting in Dublin (Ohio) in 1995.

It proposes a set of elements to describe a digital document that is

- simple and intuitive enough such that non specialists can use it
- general and flexible enough such that it can cover many domains
- powerful enough such that also elaborated descriptions can be included

To accomplish these goals a number of principles where set up for the elements:

- The Dublin Core element set is **extensible**: new elements for specific documents can be added
- all elements are **optional**: there are no mandatory elements
- all elements are **repeatable** within a description
- all elements are **modifiable**: attributes can specify how the content should be interpreted. This means that existing formats and content description systems can be used within Dublin Core.

A description of the initiative and the proposed 15 elements can be found at http://purl.oclc.org/dc.

Systems using Dublin Core should adhere to these principles. This means especially that they should be able to ignore elements and attributes they do not know. It does not mean that they have to be able to use all information provided.

## References

Burkhart, M. (1997). Thesaurus. In *Grundlagen der praktischen Information und Dokumentation*. München: K. G. Saur, ch. B 6, 160-179.

Dumais, S. T. (1991). Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments, & Computers 23*(2), 229-236.

Ferber, R. (1997). Automated indexing with thesaurus descriptors: A co-occurrence based approach to multilingual retrieval. In *Research and Advanced Technology for Digital Libraries. First European Conference (ECDL’97). Proceeding* (1997), C. Peters & C. Thanos, Eds., vol. 1324 of *Lecture Notes in Computer Science*, Springer.

Ferber, R., Wettler, M., & Rapp, R. (1995). An associative model of word selection in the generation of search queries. *Journal of the American Society for Information Science (JASIS) 46*(9), 685-699.

Fox, E. A., & Lee, W. C. (1991). FASIT-INV: A fast algorithm for building large inverted files. Technical report, VPI&SU Department of Computer Science, Blackburg, VA.

Frakes, W. B., & Baeza-Yates, R., Eds. (1992). *Information Retrieval Data Structures & Algorithms*. Prentice Hall, Englewood Cliffs, New Jersey.

Griffiths, A., Luckhurst, H. C., & Willett, P. (1986). Using interdocument similarity information in document retrieval systems. *Journal of the American Society for Information Science 37*(1), 3–11.

Harman, D. Overview of the Third Text REtreival Conference (TREC-3). WWW-Page: http://trec.nist.gov/pubs/trec3/overview.ps, 1995.

Harman, D. Overview of the Fourth Text REtreival Conference (TREC-4). WWW-Page: http://trec.nist.gov/pubs/trec4/overview.ps, 1996.

Harman, D., Fox, E., Baeza-Yates, R., & Lee, W. (1992). Inverted files. In *Information Retrieval Data Structures & Algorithms*, W. B. Frakes & R. Baeza-Yates, Eds. Englewood Cliffs: Prentice Hall, ch. 3, 28–43.

James, W. (1890). *The Principles of Psychology*. New York: Holt, Reprinted New York: Dover Publications, 1950.

Jones, W. P., & Furnas, G. W. (1987). Pictures of relevance: A geometric analysis of similarity measures. *Journal of the American Society for Information Science 38(6)*, 420-442.

Kuhlen, R. (1977). *Experimentelle Morphologie in der Informationswissenschaft*. Verlag Dokumentation: München.

Lezius, W. (1995). Morphologie und tagging. Technical report, Universität -GH - Paderoborn, FB 2. http://www-psycho.uni-paderborn.de/lezius/morpho.html.

Mannecke, H.-J. (1997). Klassifikation. In *Grundlagen der praktischen Information und Dokumentation*. München: K. G. Saur, ch. B 5, 141-159.

Robertson, S. E., Walker, S., Jones, S., Beaulieu, M. M., Gatford, M., & Payne, A. Okapi at TREC-4. WWW-Page: http://trec.nist.gov/pubs/trec4/papers/city.ps, 1996.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. Okapi at TREC-3. WWW-Page: http://trec.nist.gov/pubs/trec3/papers/city.ps, 1995.

Salton, G., & McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. New York: McGraw-Hill.

Sheridan, P., & Ballerini, J. P. (1996). Experiments in multilingual information retrieval using the SPIDER system. In *Proceedings of the19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '96* (1996), 58-65.

Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR 96* (1996), 21-29.

Turtle, H., & Croft, W. B. (1990). Inference networks for document retrieval. In *Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1990), 1-24.

Turtle, H., & Croft, W. B. (1991). Efficient probabilistic inference for text retrieval. In *Proceedings of the RIAO'91* (1991), ACM, 644-661.