Information retrieval (Part 2)

Paijmans, J.J.

Publication date:
1992

Document Version
Publisher's PDF, also known as Version of record

[redacted]

ITK

MEMO

ITK

INSTITUTE FOR LANGUAGE TECHNOLOGY AND ARTIFICIAL INTELLIGENCE

INSTITUUT VOOR TAAL- EN KENNISTECHNOLOGIE

ITK Research Memo
february 1992

Information Retrieval
Part 2:
Document Representations

Hans Paijmans

No. 13

1992/13

Table of Contents

V. Properties of documents.

1. The many faces of the document.

When we are talking about document representations, we should decide exactly which properties of the document are to be represented. The problem is, that in every document we have to distinguish between at least three totally different levels or areas of properties:

1. the properties of the document itself, as an *object*,
2. the properties of the document as a *string*, a *text* or a *collection of characters* and
3. the properties of the *contents* of the document.

The first two groups we will call the *data properties* of the document, the third the *info properties*, to be stored respectively in the DR and the DKR. Of course this last group can be subdivided in a multitude of levels and areas, but that will not concern us here. Overlooking for the moment the textual properties of the document, we see that this partition at least partially reflects the usual division of tasks in museums and libraries: registering and cataloguing:

> *To register an object is to assign to it an idividual place in a list or register [...] in such a manner that it cannot be confused with any other object listed.*
>
> *To catalog an object is to assign it to one or more categories of an organized classification system so that it and its record may be associated with other objects similar or related to it. [Guthe, 1964].*

It should be observed that in museums, where the objects generally display more 'individuality' and often have a bigger value, this division is more pronounced, Nevertheless the museum object has much in common with the document when we try to register or to catalog it.

Below we will show how properties of the document as an object are stored in the MARC catalog format (fig. V.1), which by now is one of the standards for libraries. Then we will turn to the document as a collection of characters and consider the principles of TEI, the Text Encoding Initiative, which tries to formulate rules for the describing of the textual properties of a document. In the next two chapters we will then consider some existing techniques for extracting and storing the *contents* of documents.

2. The document as an object.

The view of the document as an object to be collected, managed and described is generally found in libraries and similar institutions, such as museums (substituting physical objects for documents). These institutions traditionally not only have to

make the documents in their collections accessible on subject, but they also have to keep track of the individual books and volumes for other purposes, e.g. storage, lending or insurance. For this reason many catalogues are very much centered on the objects themselves and even assignments to classification systems have the distinct flavour of sticking just another registration number on it.

Before the electronic age the traditional way to organize documents and their relevant properties, was by systematically storing written descriptions of the document. It was found that a cardfile was very efficient, because of the ease with which the individual cards were handled, inserted or rearranged. Also it became obvious that to reserve fixed areas on the cards for observations of the same kind (e.g. author or title) improved the speed of scanning through the cardfile and so the *fixed format*[1] was born.

This approach works well enough if you consider the documents as objects, to be managed and registered as so many sacks of beans. The cards were sorted on the heading **Author** and possibly on **Title** and so this registration could be used for some minimal information retrieval actions. The main retrieval mechanism on topicality remained the *shelf order*: books of comparable contents were physically stored together and if one book did not satisfy the needs of the user, the volume next to it possibly would. This system has at least the virtue that browsing though adjacent books was ver easy and thus serendipity was ensured. This shelf order obviously admitted only one heading or key; headings on any other attribute, e.g. year of publication could not be represented in the same ordering, although suborderings are sometimes possible. The system also had the inconvenient characteristic that an increase of the volumes on any subject might cause a shift of all subsequent books to other shelves, other rooms or even other buildings.

This situation prevailed until well in the nineteenth century, in fact it was Dewey, who first devised a system that assigned a subject notation to *books* instead of to *shelves*. He published this system anonymously in 1876.

When cataloguing broke away from the shelf order, it was implemented in the same card system that already handled the registration. The topicality of the book or document, i.e. its place in a classification system, was considered as one more characteristic of the object, to be described and stored in its own pigeon hole and to be retrieved in the same way. In the pre-coordinative systems this was sufficient: cards were organized according to the classification system and there was no apparatus to use them for retrieval without the user having to scan at least the records in the adjacent areas. And if the user handles records, be it ever so superficially, he cannot help but interpret its contents. Together with the unavoidable inconsistency in the categorizing of the documents by human effort, this had peculiar effects on the quality of the retrieval systems.

---

1 Everybody who has seen how much text a registrar or librarian can cram in the few square inches of just one card, knows that the expression "Fixed Format" is very relative indeed.

To start with, errors and inconsistencies in the cards, and the fact that many cards were scanned by the user during a search, caused documents to disappear or to pop up unexpectably. Putting a good face on it, librarians called this *serendipity*, meaning that if you go out to search for one thing, you might find something else, as valuable as the original thing or even more.
Serendipity[1] is considered an asset for an IR system, but it is very difficult to introduce it artificially or to measure it and apart from that it should not get in the way. Then, again, the average user displayed a very human tendency to stop searching at the point he felt he had enough information for his needs, a phenomenom akin to the *futility point* as described in chapter II. This caused a certain number of documents to be used repeatedly and others, that happened to be back down in the ordering of the documents, to be consulted rarely or never, even if they were as useful as the documents in front (this problem survived in the electronic age with a vengeance). To counter this phenomenom much research was done on ranking strategies, which should ensure that the order in which the retrieved titles were presented, was one of estimated relevance.

People who use such systems on a regular basis, e.g. the librarians themselves, get to know its contents and that of the collection it represents (subject to the problems and limitations above) and subsequently grow into human IR-systems, able to extract information on a level that was not built into the artificial system. They should not be confused with the *searchintermediary* of modern IR systems, who often is adept only in the index language and handling of an IR system, not in its contents. Indeed the quest of modern information retrieval could very well be described as an attempt to combine the speed and accuracy of the (computerized) artificial systems with the insight of an expert librarian and the 'userfriendliness' of a search intermediary.

When the computer was pressed into service as a filing cabinet, the cards were naturally converted to *fixed format* records (but here the adjective meant exactly what it says). It was found that a computer could sort and select these records better and faster than men, but that these electronic wonders were very finicky about the exact place and contents of the fields. Mixing different attributes in one single field gave you just that: mixed attributes. Taking museum records as an example: if you use the field '**Material**' to store the descriptors *Aluminum*, *Iron* or *Gold*, there is no direct way to find all *metal* objects. Going back to the descriptor *Metal* instead of the more precise descriptors *Aluminum* or *Gold*, would correspondingly degrade the value of the system. The logical consequence was that more and more fields and sub-fields were added to the record formats and the early years of automated catalogues gave birth to some real bizarre description and coding systems. Nevertheless the automatized registering of books may be considered a success and almost all libraries now use computers for their cataloguing. The most popular format seems to be the MARC format of the Library of Congress, part of which we will describe below.

---

1 The name comes from a mythological prince of Ceylon, or Serendip as it was then called.

Control fields
001 Record control number
002 Subrecord directory datafield
008 Information codes

Coded data fields
010 Library of Congres card number
015 British National Bibliography number
017 Correction message
018 Amendment message
021 International Standard Book Number (ISBN)
022 International Standard Serial Number (ISSN)
024 BLAISE number
037 Physical description coded information field
041 Languages
043 Area codes
044 Country of producer
046 Coded data-music
047 Form of composition-music (reserved for future use)
048 Number of instruments or voices-music (reserved for future use)
050 Library of congres classification numbers
080 Universal Decimal Classification number
081 Dewey Decimal Classification number (old edition)
082 Dewey Decimal Classification number (current edition)
083 Verbal feature
085 British Catalogue of Music Classification number
087 National shelf-mark
092 Britsh Library Lending Division shelfmark (reserved for future use)
093 'Back-up'libraries'serial holdings (reserved for future use)
095 Science Reference Library classmark (reserved for future use)

Main Entry Heading Fields
100 Personal name main entry heading
110 Corporate name main entry heading
111 Conference, congres, meeting, etc. name main entry heading

Title fields
222 Key-title
240 Uniform title -excluding collective title
243 Collective title
245 Title and statement of responsability area
248 Second level and subsequent level title and statement of responsibility information relating to a multipart item

Edition field
250 Edition area

Material specific fields
255 Numeric and/or alphabetic,chronological or other designation area (serials)
256 Mathematical data area (cartographic materials)

Imprint field
260 Publication, distribution,etc. area

Physical description field
300 Physical description area

Price field
350 Terms of availability

Series statement fields (cf 800-840)
440 Series area- title of series in added entry heading form
490 Series area- title of series not in added entry heading form

V. 1. MARC format (1).

2. 0. 1. The MARC-format.

The advent of the computer gave birth to several description formats for documents, of which the MARC-format, adopted by the Library of Congress, has been the most successful. MARC (MAchine Readable Catalogue) was developed for the library environment and the eight different MARC-formats cater for several

Notes fields
500 nature, scope or artistic form note
501 "With" note
503 Dissertation note
504 Bibliography and index note
505 contents note
508 statements of reponsability note
511 ISBN and ISSN note
513 Summary note
514 Title proper, parallel title and other title information note
515 Numbering and chronological designation note (serials)
516 Mathematical and other cartographic data note cartographic materials)
518 Change of control number note
528 Publication, distribution etc. note
530 Other versions available note
531 Physical description note
532 Serie note
534 Reference to published description note
536 Characteristics of original of art reproduction, postcard, poster etc. note
537 Program note (machine readable data files)
546 Language of the item and/or anslation or adaptation note
554 Frequency note (serials)
555 Indexes note (serials)
556 Item described note (serials)

Subject heading etc. fields
600 Personal name subject heading
610 corporate name subject heading
611 Conference, congress, meeting etc. subject heading
640 Uniform title subject heading
645 Title subject heading
650 Topical library of congress subject heading
651 Geographical library of congress subject heading
690 PRECIS string
691 subject indicator number
692 reference indicator number
695 Index terms (reserved for future use)

Added entry heading fields
700 Personal name added entry heading
710 corporal name added entry heading
711 Confference, congress, meeting etc. added entry heading
740 Uniform title added entry heading
745 Title added entry heading - excluding uniform titles

Tracing field
790 Tracing data

Series added entry heading fields
800 Personal author series added entry heading
810 Corporate series added entry heading
811 Condference, congress, meeting etc, series added entry heading
840 Series title added entry heading

Reference fields
900 reference from a personal name
910 Reference from a corporate name
911 Reference from the name of a congress, confderence, meeting etc.
945 Reference from a title of a work

V. 2. MARC format (2)

types of material, including monographs, scores, sound recordings, manuscripts, maps, audiovisual materials and machine readable data files.

It must be stressed here that the MARC format was and is not aimed especially at informationretrieval. Still, with eight different types of material and hundreds of fields and subfields, there is an enormous number of discrete data that may be stored on MARC-records and individual libraries will only enter a subset of them. Their task is made easier by the fact that, contrary to e.g. a museum registrar, the librarian may copy skeleton records from another library or from a vendor and fill in the fields that are relevant for his collection.

A closer inspection reveals that there are no direct descriptions of 'aboutness' in the MARC format: there is a *'(505) contents note'* though. The great number of fields, where the *signature* (code) of a classification system may be stored, more than makes up for this omission.
For a full discussion of the MARC format see [Reynolds, 1985] and [Attig, 1983].

3. The document as a string.

If the managing of documents as objects poses no particular problems any more, the same cannot be said of the managing of documents as strings or pieces of text. This is not because the correspondent properties are not easily extracted (most of them are readily isolated by computer programs: e.g. length in characters, number of sentences etc.) but because no particular need was felt to store them explicitly: not many users will want to retrieve a document on the number of sentences in it or on the statistical division between vowels and consonants and such properties certainly have no direct bearing on its topicality. However, in the next chapter we will see that there certainly are statistical relations between such properties and the contents of the document, as in for instance the *relative document frequency* (see next chapter).

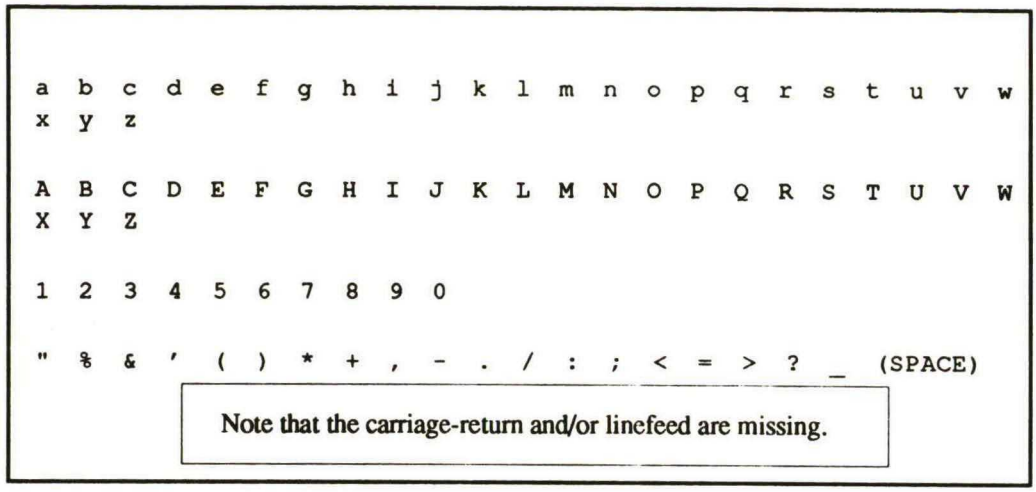
More difficult to identify, but still recognizable for a machine are document parts like the TOC-structure, bibliography and bibliographic references, front-matter, back-matter, to mention a few. Automatic syntactic parsing may be considered, if not solved, then not any more the most important obstacle to text-analysis and a reasonably exact likeness of the syntactic structure of the document (or at least from relevant parts of it) may be generated and stored[1], that is: many researchers report experiments in which syntactic parsing is used (see next subparagraph (3.2) for a short discussion). All these properties may be considered to belong to the document-*object* and find a place in the Data Representation of the document. The document may be seen as the union of all these representations. So we will discuss here respectively the visual properties of the document, its properties as a collection of strings (syntax) and finally we will consider how these (textual) properties may be described, taking the Text Encoding Initiative as an example.

3. 1. Visual structures and clean text.

When talking about text and documents, the word attribute may have a different meaning from that mentioned above: it then indicates the visual properties of the text or parts thereof, which serve to emphasize certain parts of the text or which organize the document by distinguishing its logical parts. These attributes often cause a noticeable shift in the meaning of the sentence. Compare for instance

---

1 My collegas, who are working on syntactic parsing will not agree with this statement.



V. 3. ISO 646 character set.

John ate the apple

with

John ate the *apple*.

where the attribute *italics* enables inferences about respectively the number of people and the number of consumable objects in the room. Another and similar function of this mark-ups would be of emphasizing words (e.g. underlining or italics) or even characters (actually syllabi) by the adding of emphasizing accents:

Jóhn was here, not Killroy!

which should not be confused with real diacritics.

More important are the possibilities of using similar attributes dividing the document in an hierarchical structure of chapters, paragraphs etc., with words at the buckets. Before we embark on the description of general document representations, we will first turn to the information that may be gleaned from the visual representation or lay-out of the *paper* document and the corresponding features of the electronic document.

Documents may be presented in several forms. To start with, of course, there is the facsimile (xerox-copy), which for all goals and purposes is the document itself. This facsimile may consequently be stored in several ways (e.g. micro-fiche), of which the bit-image of the printed page in a file on some magnetic medium is so far the most advanced way. Somewhere along this road the document may have been processed by an Optical Character Reader (OCR), which extracted from it an ASCII representation, to be stored separately[1]. This representation may or may not

1 When we mention the ASCII-code here we mean any code, in which the characters of the alphabet may be represented. The ASCII-code represents the characters of the alphabet in one single byte, which was sufficient

```
<!DOCTYPE Researchpaper [
<!ELEMENT Document          (front, body,back)>
<!ELEMENT Front             (title,author+,abstract)>
<!ELEMENT Abstract          (Paragraph*)>
<!ELEMENT Body              (Section*)>
<!ELEMENT Section           (Heading,(Paragraph+|(Paragraph*,Subsection+)))>
...
...
<!ELEMENT Citation          (#PCDATA)>
]>
```

V.4. A Simplified DTD.

include information about the original layout in one notational convention or another (e.g. SGML or TEX). If it only exists of the printables, spaces and carriage returns of a normal typewriter, we call it clean text or pure ASCII (which is incorrect, but has nevertheless become common usage. Correct would be: ISO 646, see fig. V. 3). Of course the lay-out information may have been added by other means, e.g. by the wordprocessor of the author himself.

The lay-out of a document serves two functions: esthetics and additional semantics. Sometimes the two are difficult to separate: the iuxtaposition of data in a table or emphasizing by italics are examples of semantics; an elaborate initial (first character of a chapter) clearly has an esthetic function, but the centering of a title may or may not serve both functions. Decisive in such cases is whether or not a native reader[1] would recognize the additional semantics of the lay-out in the pure ASCII-text.

The point is, that these additional semantics are not described in the pure or typewriter-representation of the text, but in its visual appearance. The human reader is trained to add this information to the ASCII-information, so completing the semantics of the document. Therefore it should be the first step in the processing of a document in a FTIR system to generate its ASCII-representation, including mark-ups in one of the several popular mark-up languages.

The Text Encoding initiative (subparagraph 3.3 below) covers the mechanics of font-shifts, especially characters, in depth. Needless to say that a text may have an intricate structure without having as much as one single mark-up code. In that case other techniques, e.g. heuristics, must be applied to recognize the visual structure of the document. Some work on the heuristics of title pages of books has been done by Davies [Davies, 1990].

---

for the needs of the western (latin) alphabet. Attempts are made to introduce an new standard, Unicode, which uses two bytes and more than 27,000 characters (Computerworld, 27 febr.1991, p.1).

1 Native reader: The reader of a text that was written with somebody of his general level of knowledge, education etc. in mind. Of course he must at least have sufficient competency in the language to be able to read the document.

Using these mark-ups a document representation may be constructed, which isolates and preserves the hierarchical structure of chapters, headings, paragraphs etc., that is inherent in almost every document ([MacLeod, 1990]). See fig. V. 4 for a SGML-encoded document structure. He uses this structural representation for (a kind of) field control, although the semantic meaning of these fields is not nearly as well defined as that of the fields in an orthodox data base. See also [Burkowski, 1991].
We will call these structures the visual structures or the visual syntax, because this structure is not contained in the semantic/syntactic correctness of the sentences or the orthograpy of the words, but in visual additions/changes to them.

There exists a problem here. The meanings of words and syntactical constructs are relatively easy to define and they are more or less axiomatized by dictionaries and grammars. The visual syntax and semantics traditionally are less stringently defined, i.e. there exists no universal grammar for it.
In the last few years some conventions in mark-up languages have become ad hoc standards, e.g. LaTeX in a part of the scientific community. Nevertheless the majority of printed material will follow any number of conventions or even make up totally new lay-outs. And even if by any chance one convention became the absolute standard, this would only help in deciding on the exact structure of the document in chapters, paragraphs and the like and that only in documents published afterwards. Anyway, there is no way of adding a clear meaning to the fact that a word in a document is in the first sentence of the second paragraph of the third chapter of the document, except for the very tenuous statistics as mentioned below.

Although these visual semantics at first sight are perhaps not very useful in the searching of information, we may yet use it in this process. Apart from that it may have a positive effect on the reporting part of the retrieval process (see also [Holstege&Inn&Tokuda, 1991] for attempts to capture semantic and pragmatic contents from visual representations).
To start with we have a hierarchically structured representation of the document in its visual structure as explained above. Although the semantics are not clear (the visual structure is very much syntaxis), it certainly adds semantic value to the document.
For instance the following question could be constructed ([MacLeod, 1990], p.203):

```
text = list gets Subsection (having any Paragraph where =
"database" in first Sentence) of Section where "retrieval"
in Heading.
```

If this question is placed along a SQL-query like

```
text = select Namefield from datafile where Salary=$30.000;
```

it should be clear how imprecise such a structure is compared to the clarity of the relational fields. The capitalized words in both examples function as fieldnames, but whereas the fieldnames in the SQL-statement are decisive for the semantics of the fields, the 'fieldnames' of the MacLeod question have at best a very tenuous connection with its contents.

3. 2. Syntactic structure.

Fools rush in where angels fear to tread and the same may be said for the cavalier fashion with which syntactic parsing is treated by information retrieval scientists. I will follow this tradition wholeheartedly.

Although many issues in syntactical parsing are not yet solved, there are perhaps some parts of it, which may be considered sufficiently mature to be used in informationretrieval. Literature shows many places where IR strategies use parsing to select parts of a document, or to decide on the relative importance of sentences. These techniques generally do not take semantics in account, and for this reason we will mention them here.

One early attempt to use syntactic information is the research by Earl [Earl, 1970]. She tried to test the hypothesis, put forward by Dolby and Resnikoff, that the syntactic form of a sentence might by itself be an indicator of sentence significance, assuming that as the letter strings of a word were indicative of the part of speech of a word, analogously, the part-of-speech strings of a sentence might well be indicative of sentence significance. A parser was developed as part of her experiments and although no significant results were obtained, the parsing was reported sufficiently accurate and reliable for this kind of work (Op.cit p.316). Using hindsight it is easy to say that this particular hypothesis never had much promise, but the importance of syntactic information and syntactic parsing remains clear.

Another system, that uses syntactic parsing extensively is CLARIT [Evans, 1991], see also chapter VI. This system works on the assumption that, from an information-theoretical point of view, NPs are among the most interesting units in a document and that, consequently, the matching of such units with known 'interesting' terms, offers a way to succesful retrieval. The CLARIT indexing system thus consists essentially of syntactical parsing, aimed at extraction of the NP's, combined with a thesaurus for the semantic contents. Here too, the syntactical parsing is not seen as a problem: it clearly is sufficient for NP-extraction.

3. 3. The Text Encoding Initiative.

It should be clear from the above that the very shape of the text carries a semantic message and while we will not go all the way with McLuhan and call the medium the message, it should be clear from the last pages that the medium at least describes part of the message. Seen from an IR point of view it remains the question which parts of this medium level should be made explicit and how to do it.

A possible approach would be that of the Text Encoding Initiative (TEI), which proposes a standard for the describing of text with all its properties, doing for text more or less what the MARC format does for bibliographical objects. Originally conceived as a method to exchange texts between linguists, it is rapidly growing into an exhaustive analysis of all possible properties of text, covering such diverse subjects as the shape of characters, the design of the layout and the syntactic structure. The most interesting parts of the TEI from the viewpoint of IR are the treatment of

1. the bookkeeping type data (bibliographic control)
2. the textstructures and the
3. analytical and interpretative information,

although many more properties of the text are made explicit and encoded, notably the visual information as described in subsection 3.1 above.

We already mentioned several times that the automatic extraction of meaning from documents is one of the most important research areas in IR-science. The markups of the TEI might be an important vantagepoint, even if many text-propertiesthat are described by it, as yet have to be coded by hand.

In the following paragraphs we will give a short overview of the salient features of the TEI-draft of 1990; however, it is by no means a complete summing up.

3. 3. 1. Bibliographic control, encoding declarations and version control.

The TEI recognizes three kinds of bookkeeping type data of a text. To start with there is the bibliographical information both about the original text, of which the machine-readable text is created, and the file as an object in its own right. Questions of "who did the transcription" or "what is the key to the transcription scheme" are also put in this section. The second section concerns itself with questions about tags and coding conventions, among others whether typographical errors in the original were corrected in the transcription or spelling was modernized. Third comes a history of the textfile (later modifications and who is responsible for them (version control).

In a FTIR system, that is working with documents in the sense of books, articles and similar publications, the most interesting part is the *source description.*, in which the original document is described. The TEI suggests that this description has a format (i.e. tags) similar to either the bibliographical description of the file itself or to the in-text bibliographical citations (see next paragraph (C2)).

The bibliographic description of the electronic file then consists of a very abbreviated set of bookkeeping type data, of which the most important are:

<title.statement>
Title and statement of responsibility (split in author, sponsor, funding agency and principal researcher),

<edition.statement>
An edition is the set of all copies produced from a single master and issued by a particular publishing agency.

<publication.statement>
The person or institution by whose authority this edition is made public.

<notes.statement>
As is usual, a notes field acts as a general repository for observations, which are difficult to fit in a rigid structure. Of special interest from the information retrieval point of view are the "*Nature, scope, artistic form or purpose of the file*" and the '*summary description providing a factual, non-evaluative account of the subject content of the file*" (TEI p.64).

Compared to the extensive set of attributes covered by the MARC-format, this may hardly be called superfluous. The writers of the TEI-guidelines remark themselves that the fileheader, in which these statements have their place, is not intended as a library catalogue record. It would have been a good idea though, to follow the MARC-record more closely, even if many fields would have been empty.

3. 3. 2. Text structures (features common to many text types).

"*By a text we understand an extended stretch of natural discourse, whether written or spoken*" (TEI p.71). Strictly taken, this definition should not cover corpora, the contents of which often do not consist of *extended* stretches, but rather contain isolated fragments. And the describing of corpora like the Eindhoven Corpus (fig. IV.2) certainly is one of the aims of the TEI.

Nevertheless this definition is close enough to the concept of a document as discussed in chapter IV of this publication and the TEI gives a complete set of tools, to tag almost every distinctive part of a text that might be imagined.

The TEI distinguishes two kinds of markups: the descriptive markup, that tries to distinguish underlying textual features, and the presentational markup, that simply marks the typographical features. Presentational markup is easier to apply; descriptive markups allows for more sophisticated analysis of the text, but is more costly in terms of time and effort, runs the risk of introducing subjective or erroneous decisions and certainly is more difficult to implement for automated systems.

Also these tags offer entry points for reference systems, which also is a very important feature, if the document is to be used in a FTIR system. Below we wil discuss most of the structures that are recognized by the TEI.

1. Core structural features.

Most texts, especially documents etc. in a library system, conform to a very basic tripartition.: the *front matter* (title, author, imprimatur etc.), the *body matter* (e.g. chapters, section and paragraphs) and the *back matter* (in which may be found an index and/or a bibliography and/or other distincive parts). Many elements have a distinctive value in information retrieval, notably the title (often the only part of the document that contains retrievable items), the index or the bibliograpy, which is the subject of many experiments in IR, e.g. [Lee pao&Worthen, 1989]).

McLeod and Burkowsky (see above, subsection A) have done much work on the subject of information retrieval in structured texts. But already Luhn and other early scientists have commented on the relative importance of different parts of texts and even the position of text relative of other text.

2. Basic non-structural Features

The basic non-structural features are those features, that occur freely in texts and may form part of many other structures. Most have no consistent internal structure and often they contain simple embedded structures, which are called *crystals* in TEI terminology.

Paragraphs are the most important of these non-structural features as they make up most of the text. The TEI gives no definition of paragraph-boundaries, but a paragraph generally is a unit consisting of a relatively small number of sentences, separated from other units by one or more (hard) carriage-returns. In these paragraphs may be found text-elements that may be tagged as highlighting, quotations, names and the crystals as mentioned above, although the TEI does not make clear what exactly distinguishes non-crystals like *names* from crystals like *numbers and dates.*

Crystals are text-elements like *Lists, Notes, Index entries, Numbers and Dates,* each of which may be tagged as such. The importance of the fact that elements like names in the text may be made explicit and recognizable in a text is evident from an FTIR point of view. The same goes for quotations, index entries and other elements, though sometimes less so than the writings of McLeod and Burkowsky (op.cit.) suggests.

3. Bibliographic citations and references.

The TEI provides a complete set of tags for the handling of bibliographic references, both as references in a running text or as lists in the back matter. The importance of these references from the information retrieval point of view already has been commented upon.

4. Links, cross references and reference systems.

Reference systems are necessary to mark a particular place in a text. Of course the structural units (chapters, sections etc.) may serve as a refential frame, or the more traditional page and line structure. However, often a more precise entry is useful, especially in electronically accessible files.

The links, that accompanies the hypertext system, of course need markups of their own. The concept of hypertext and similar navigation systems for textfiles is of obvious importance for information systems, although perhaps less so for the information retrieval in a narrower sense.

5. Formulas, Tables and Figures

Formulas, tables and figures are also considered by McLeod and his collegues to be important items in information retrieval. As we have seen, it is perhaps not so much pertinent to information retrieval as to data retrieval.

3. 3. 3. Analytic and Interpretative information.

The structures for bibliographic control and text structures as mentioned in the paragraphs C1 and C2 above, do not pose any particular problems in the

```
<f.struc t  id=sample>
    ...
    <feature>
        <f.name>  category </f.name>
        <f.struct>  noun </f.struct>
    </feature>
    ...
</f.struct>
```

V. 5. Suggested noun-tag in TEI

interpretation, except perhaps in the descriptive text markup as opposed to the presentationalmarkup. The TEI also proposes markups for the linguistic analysis of a text and holds out the possibility of yet other types of analysis and interpretation (TEI, p.129).

Restricting themselves to the linguistic properties, they rightly state that a notational system like the TEI, that tries to offer a wide hospitality to all possible theories, should not implicitly privilege certain schools of linguistic thought, although it cannot be avoided that some systems may be easier implemented in a given notation than others. They note on page 130 of the draft that the TEI markup system certainly is more hospitable to Lexical-Functional Grammar and Generalized Phrase Structure grammar than to Governement Binding or Categorial Grammar, although it is sufficiently general to accommodate them in one way or another.

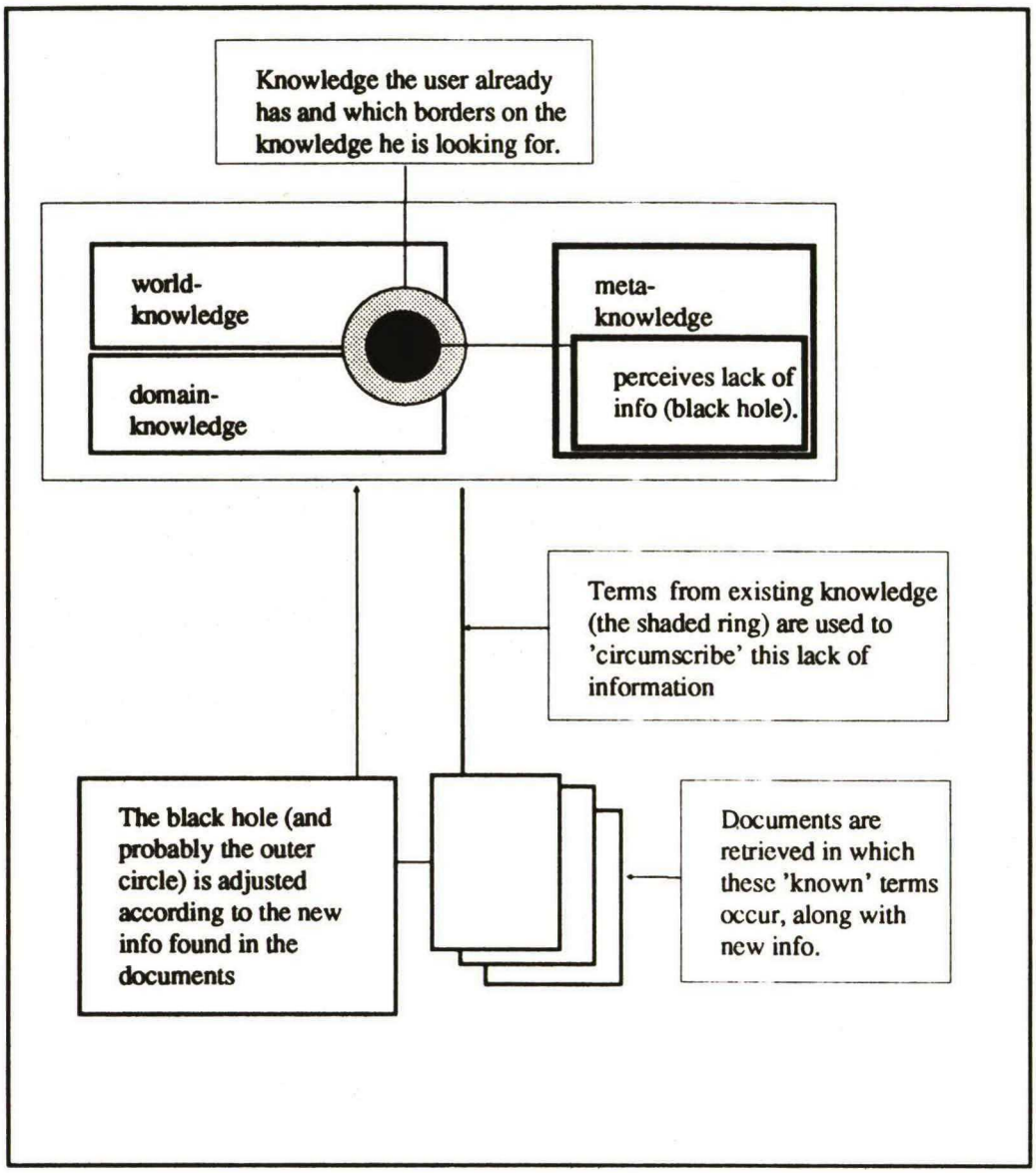
If we look at the example in fig. V.5 we may see how the TEI refrains from associating specific elements of linguistic theories with specific SGML tags and attributes. In stead of supplying tags for <noun> or <verb>, they have chosen for the much more involved approach of defining categories and their values in the very general *feature*-stucture.

One might wonder if the penalty of such verbose circumscriptions would not prohibit the use of the TEI system for all applications but the direct interchanging of textfiles between different systems, but then that is its professed goal.

3. 4. The document as container of info.

The contents of a document in terms of topicality, "aboutness", as found in abstracts and rightfully belonging in the Document Knowledge Representation, are not so easily identified, extracted and described. Also there rarely is a direct link between the data in the DR and the DKR, either causal or statistical; that is: there are many possible links, but no rules to choose the correct ones.

The question is which properties of the document should be extracted and made explicit for retrieval purposes. It is on this info, combined with his previous knowledge, that the human user decides whether he wants to see the whole document, or at least more of it.



V. 6. The 'black hole' retrieval process.

3. 4. 1. The retrieval process.

What happens when a human user tries to retrieve a document on content? To start with, he has a set of internal knowledge representations, which cover respectively his general world knowledge, his domain knowledge (related to the domain of his query) and meta-knowledge about the state of his knowledge (fig. V. 6). This meta-knowledge detects a hole in the user knowledge and causes him to search for and consult certain documents (actually the decision to search for documents and the selection of a place/library/collection where to look, comes first and rightfully

belongs to the total retrieval process). Now although new knowledge (i.e. knowledge that causes something in the KR of the user to change) may be present in a document, it is the *matching* of the document knowledge and existing user knowledge that causes retrieval in an IR system. In other words: the user can only search in terms he already knows.

So if we want to single out certain properties of the document (be it from the document as object or as container of info) and from this create representations of documents, they should be organized for access according to the existing knowledge of the prospective user rather than according to the new knowledge that is contained in the document. Speaking in terms of assigned vs. derived indexing, the document again has to be *assigned* to a niche in an existing system and it is this system, not its contents, that communicates with the user.

It goes without saying that such systems will be far more complicated than the old classificationsystems. This orientation on assignment and existing knowledge does not mean that no new data or relationships could be entered in the DKR, only that this new knowledge should be presented where possible in known structures and terminology.

When we consider the creation of document knowledge representations as primarily an assignment-activity instead of a rebuild-activity, this has the advantage that we can use recognition instead of cognition, checking and ticking off the arguments, which decide how and where the document has its place in the system. The system itself can be largely pre-built. Of course these arguments and the structures to be recognized, have to be *derived* from the document, i.e. from the data representations of the documents. The FRUMP system [Dejong, 1979], although rigid and ungainly, should be taken as an example, rather than the ubiquitous keyword systems, however subtle their probabilistic theories.

VI. Document representations.

In this chapter we will try to describe some routes that lead from the original document to the document representation(s) that is (are) used by the system. A major problem in spelling out these descriptions is the multi-stage character of many of these conversions. As we have mentioned before, it is not unusual to talk about e.g. full-text retrieval systems in cases, where by 'full-text' in reality an abstract of the original document is meant, so the ultimate *docrep* may well be a representation of a representation, the first of which (document -> abstract) is generated by hand and the second (abstract -> keyword-representation) is done using a computer. In such circumstances the results of performance tests as measured by user satisfaction are dubious as a measure of the relative success of each of the two translations, This is because the user satisfaction is generally not based on the *abstract*, from which in such cases the keywords derive, but on the whole document.

Another problem, that pops up when a systematic description of different kinds of document representations is attempted, is a marked tendency for the more structured docreps to merge with each other into a general representation of objects and concepts in the general domain of the system. Imagine a database of texts in the domain of, say, cars and its mechanical components. Systems like RESEARCHER or SCISOR (see chapter VII) would extract the information from the documents and build hierarchical representations of the *cars* rather than representations of the individual documents in the database. Use of the general term *Document Representation* would here be misleading, therefore we will in such cases use the narrower term *Document Knowledge Representation*, because it is the *knowledge* that is represented, rather than the document.

In this chapter the Document Data Representations that concentrate on the individualdocuments are described, together with some retrieval techniques that go with them. In the next chapter we will describe some of the more structured representations and the systems in which they are used, i.e. the Document Knowledge Representations. By the latter, as we have said, we mean those representations in which the contents of the document are described in symbols, that relate to each other in some non-trivial way and so represent information or even knowledge (together called *info*) about the underlying domain rather than about the documents themselves. Document Data Representations, then, are those representations, where the symbols, when taken from the document, have no relation to each other except for the membership of the set of symbols, extracted from that individual document by that individual method.

The boundaries between the two classes are not all together sharp. We will see at the end of this chapter about document representations two examples of systems, that might as

well have been placed in the next chapter. They have been placed here, because they mark some aspects where the transition from one class to the other occurs.

The purpose of the extractions or abstractions is to make such info as is the ultimate goal of the retrieval activity, explicit and to store it in such a shape as is most appropriate for query-operations. Processing techniques are generally aimed at whittling away those parts of the original document that are irrelevant for that purpose.

We suggest a division in three different methods:

1. indexing, which will occupy most of this chapter. The end result of an indexing operation is a set of keywords or keyphrases.
2. extraction, that may be considered a special case of indexing, but which aims at a coherent description of (the contents of) the document, rather than a set of keywords.
3. subtraction, a method that in itself does not make much info explicit, but which may be used by other techniques.

As we have said, in this chapter we will try to touch on some of these representations and techniques and on their worth for information retrieval purposes.

1. Indexing.

It is felt by most researchers and system builders that the easiest representation of the document is a set of keywords or key phrases, either assigned or derived by a particular technique. These keywords may be flat indices or they may be organized according to some classification system or according to a thesaurus-like construction. The classification system or thesaurus are generally created by hand: some attempts to generate classes automatically are described below.

1. 1. Derived indexing.

We may formulate part of these representations in terms of *derived* and *assigned* indexing. Starting with derived indexing, in which the terms in the document representation are derived directly from the document, we will distinguish the *keyword* representation, the *key-phrase* representation and finally the *extract* (in which complete sentences are selected and taken from the original document, see at the end of this chapter). Keywords and sentences have in common that they are easily identified by typography. Representation by selected phrases is more difficult because of the additional parsing problems. An additional problem with sentence-extracts is, that for reason of textual cohesion the dangling anaphores have to be resolved, but then again, anaphores are a major problem anyway.

1. 2. Formatted indexing.

A less known, but nevertheless interesting application of indexing is the case where a form has to be filled with facts, that themselves have to be extracted from more extensive texts, e.g. medical records that are composed from radiography reports [Sager, 1981]. In such circumstances the domain is restricted and the language used in the reports is a very small subset or sublanguage, which

facilitates processing. In [Liddy, 1991] a similar project for insurance companies is described.

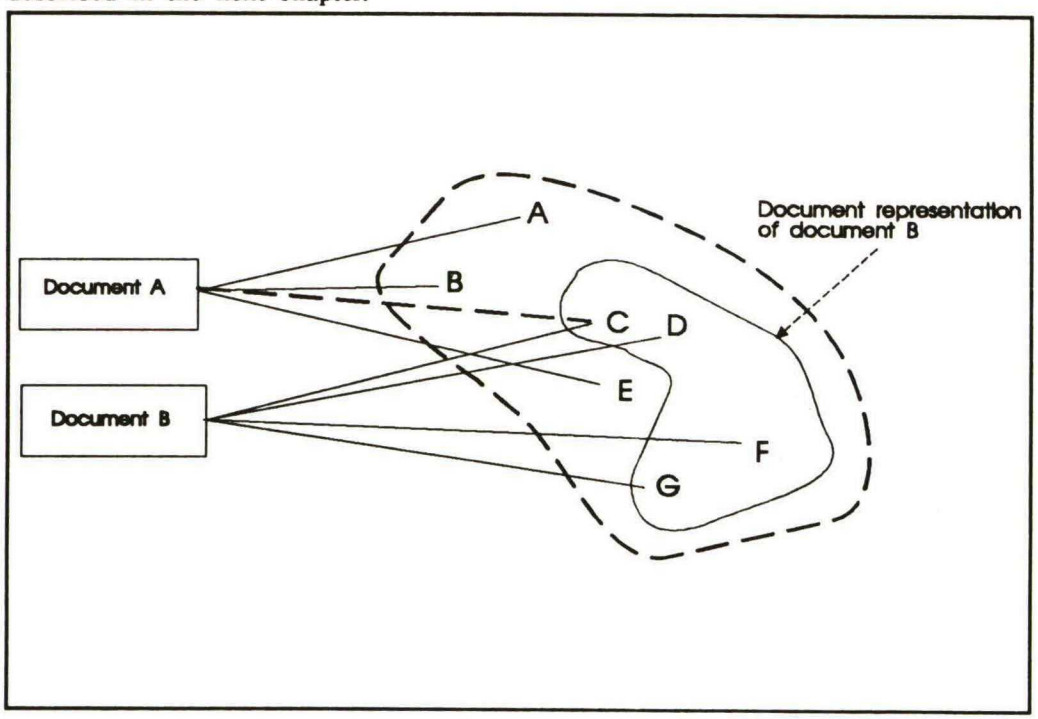
1. 3. Assigned indexing.

In assigned indexing we normally have an human indexer, who assigns the documents to a classification system. This classification system may be highly structured; it may also consist of a rather loose list of keywords (controlled dictionary), which the indexer may assign more or less as he sees fit. The more elaborate classification systems, including thesauri, may be considered as knowledge representation systems.

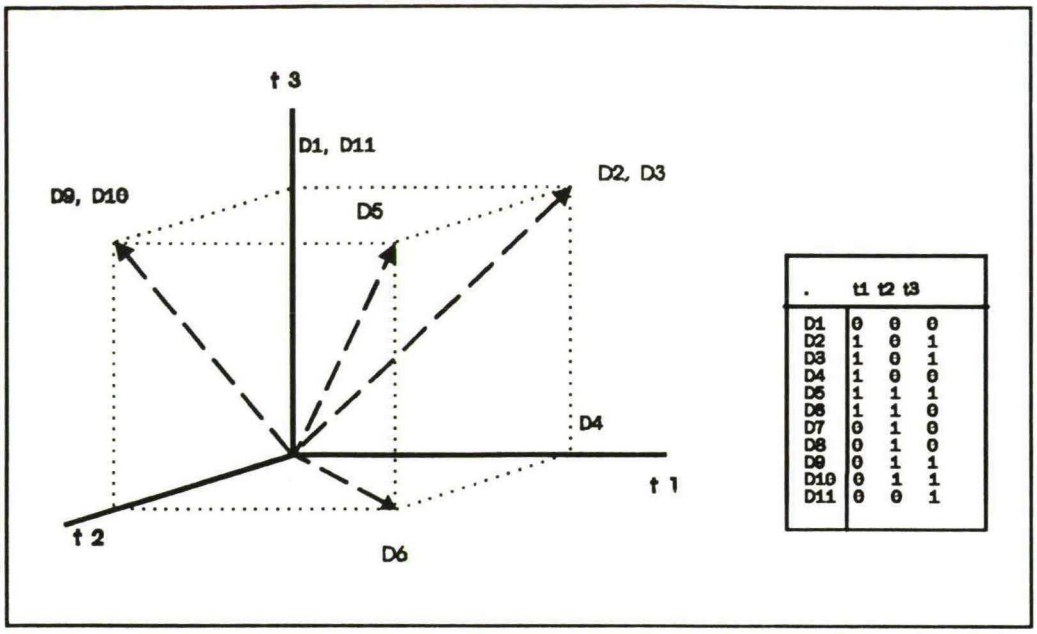
Automatic assignment to documents of index terms from pre-established lists is possible, although experiments in this direction have not been encouraging when applied to databases with abstracts or documents, according to [Lancaster, 1972]; [Borko/Bernick, 1963] and [Maron,1961]. Of course the youngest of these experiments is twenty years old, but recent research in the automatic classification of books ([Enser, 1985], see also next page) seems to confirm the earlier findings.

If we compare these results with those of the following paragraph, this seems to point to a general incompatibility between human assignments (as by a pre-established list) and the results of computational methods on texts.

In any case: systems that try to assign documents to classifications that appeal to human thinking, generally need some knowledge about the domain of document and would therefore belong to the Document knowledge representations as described in the next chapter.



VI. 1. Document representation in keywords.



.	t1	t2	t3
D1	0	0	0
D2	1	0	1
D3	1	0	1
D4	1	0	0
D5	1	1	1
D6	1	1	0
D7	0	1	0
D8	0	1	0
D9	0	1	1
D10	0	1	1
D11	0	0	1

VI.2. Document vectors.

1. 4. Clustering and Automatic generation of classes.

The generation of inverted files and the power of modern computers gives us the opportunity to try and identify groups of terms on the basis of their statistical characteristics. If, for instance, two words tend to co-occur in documents, they are likely related to each other in some way or another (and may be substituted for each other when searching).

This works in two directions:

1. The clustering of documents on the basis of the terms (see section 4 below).
2. The clustering of terms on the basis of the documents.

This second operation is of interest in building a kind of 'automatic thesaurus', which terms that relate statistically rather than semantically. It was found that (statistically spoken) some terms are *near-synonyms*, others relate in a *genus-species* set, while yet others will be related similarly to the *related term* in a conventional (semantic) thesaurus. Extensive work on this subject was done in Cambridge by [Sparck-Jones/Jackson, 1967], trying to establish links between these statistic thesauri and the semantic ones. Other investigations, however, indicated that these clusters tended to differ from the conventional, human-made thesaurus. In this respect perhaps an experiment should be mentioned, which was conducted on a small corpus of books represented by their title, BOB-index and TOC [Enser,1985]. An attempt was made to create an automatic classification on the basis of co-occurring terms and although it was found that these automatic classifications were markedly superior (as retrieval classes) to manual classifications, the general effectiveness was not high enough to justify the costs of storage and manipulation of the entire index.

Clustering is not a representation of individual documents, but is a way to represent groups of documents in such a way that their resemblance with each another is made explicit. Thus clustering does not conflict with inversion - the latter essentially is a way to solve access by storage, the former a technique to identify 'similar' records.

The first step in clustering is the location of each document in a *t*-dimensional vectorspace, where *t* is the total number of keywords, and the absence or presence of a keyword in a document is indicated by 0 or 1, respectively by a positive number for weighted terms (see fig. VI.2) The second step is the analysis of the points in this vectorspace to see if clusters can be pointed out and partitioned off. In a similar way the keywords in an IR-system may be clustered to discover groups of co-occurring and possibly related terms.

The distance between documents or between query and document(s) can be measured by the angle between the respective vectors or by measuring the euclidean distance between the endpoints.

Several attempts are made to adjust the vectors in such a way that in a query q in a vector space with relevant documents D and irrelevent documents d (relevancy reckoned by the query), the relevant documents are moved closer to the query vector and farther away from the irrelevant documents, e.g.:

$$\vec{D}' = \vec{D} + \alpha(\vec{q} - \vec{D})$$

$$\vec{d}' = \vec{d} + \alpha(\vec{d} - \vec{q})$$

where alpha is a constant.

1. 5. Some weighing techniques for indexing.

Traditionally the document representations (docreps) in an IR-system are limited to two classes. One is the bibliographic description of the document (which we define here as the bookkeeping data: author, editor etc.). The other is the set of keywords (postings), extracted from the document by one method or another. These keywords act as access points to lists with record-identifiers: the relation to the records is that the keywords are derived from or assigned to it. In an exhaustive inverted file, in which each and every wordtoken in the documents is contained, there is no further relation between the keyword and the record. If e.g. a stoplist is applied, or some form or another of weighting is applied, there immediately is an added relation between the document and its keyword-representation.

So a keyword representation of a single document in a database of several documents may be described as

$$R = \{<k,o> \mid k \text{ selected by some method } \}$$

where k = keyword and o = list of occurences.

Note that the occurences point to the smallest addressable text segment, not to the exact place of the keyword - if it exists at all in that text segment (it almost

Some examples:

(a) R = { <k,o> | k not in stoplist }

(b) R = { <k,o> | k occurs in defined part of the document}

(c) R = { <k,o> | X> *freq(x)* >Y }

where *X* and *Y* are upper and lower cut-off (Fig. VI. 3).

(d) $R = \left\{ <k,o> \mid \frac{freq(k)\ in\ document}{freq(k)\ in\ database} > C \right\}$

where *C* is a treshold.

(e) R = { <*t*,*p*> }

where *t* = any wordtoken or punctuation and
p = its exact place in the document.

always does occur there in derived indexing, but rarely in assigned indexing). Also, the list of individual occurrences of the keyword in the document is often omitted and the membership of the set just notes that the keyword occurs at least once in the document.

Above are some examples. Note example (d) describing the inverse document frequency weighting method (IDF); simple but popular and effective. The IDF is explained below.

So as the most fundamental document representation we have an exhaustive inverted file of wordtokens, punctuation and mark-up codes, in which no other information is contained than the list of documents where they occur. This fundamental docrep is itself the departing point for a whole series of representations, where different strategies are used to extract keywords from the document and/or to indicate the importance of a keyword for a particular record. The relation between the representation and the document changes accordingly.

Also we may add information to the keywords, other than the occurence-information. If in an exhaustive concordance as described above, we add not only the document number, but also the relative place in the document of each posting (as in example (e) above), we implicitly copy the document itself in the docrep, because it may be reconstructed using this information. If the document itself also is present in the index language, this effectively creates a redundancy.

1. 6. Weighing of words and phrases.

When talking about documents, we will generally refer to documents that exist of the ASCII-text, including printables, carriage returns and pagefeeds, but little else, although the visual structure as described in the last chapter may be used as an additional factors to weigh the words, as indeed is often the case. We will mention a few methods and measures below.

1. 6. 1. Frequency, distribution and other statistics.

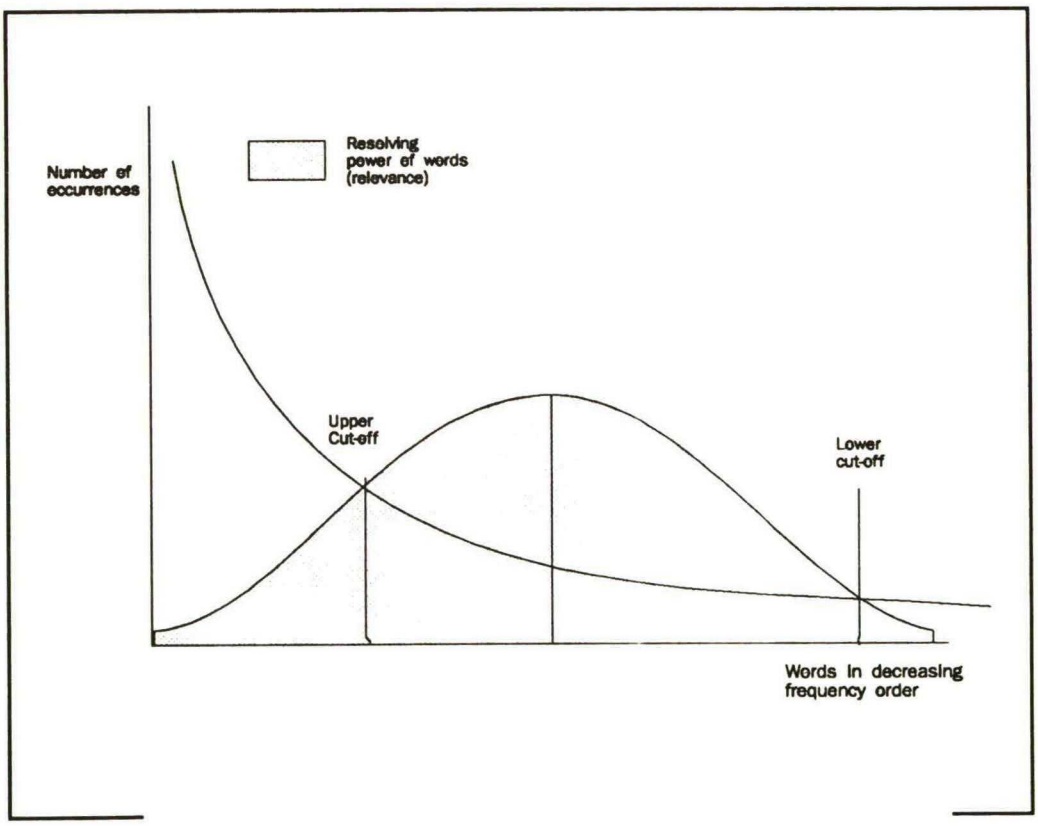
There has been extensive research in probabilistic weighting, notably by Salton and hiscooperators[Salton/McGill, 1983] and van Rijsbergen [Rijsbergen, 1979]. There are two reasons for considering word frequency as weighing factors in information retrieval. One is the well known rank-frequency law of Zipf, stating that

Frequency . rank = constant.

while the other is the seemingly contradictory intuition that words, that occur more often in a text are better indicators of what the text is about. This too already was signalled in the fifties: *"A notion occurring at least twice in the same paragraph would be considered a major notion..."* [(Luhn, 1957]. Other research along these lines was carried out by [Oswald, 1959] and [Edmundson, 1969].

Applying the rank-function law to the words in documents, we will see that the highest scoring words are function-words. A relatively short 'stoplist' may be used to exclude these function words from further processing, as they have no direct value for information retrieval purposes. But even when limiting the list to content-bearing words, cut-offs have to be used at both ends of the list.

In the figure below and formula (c) we see an example of the use of this approach. It was found that the importance of a word as a content-describing



VI. 3. Low and high cut-off in word-frequency.

word, compared with the relative frequency of the word, exhibited a normal-curve. By choosing appropriate upper and lower cut-off points it is possible to limit the words in the dictionary to those with the greatest weight. This captures the experience that both words that are to be found in almost every document and words that occurr in only one or two documents have less value in discriminating between documents. We will give two very short examples of the important probabilistic measures for relative keyword weights. The interested reader may consult [Salton&McGill, 1983] or [Rijsbergen, 1979]. See also [Evans, 1991] for recent applications.

a. The inverse document frequency.

A well-known and popular measure for the relative importance of an index term is the inverse document frequency (see formula (d), also [Bar-Hillel,1959] and [Oswald, 1959]). For each term k and document i (or query j) it is possible to compute the frequency with which it occurs, f_{ik}, and the *collectionfrequency* of term k for the N documents of the collection:

$$F_k = \sum_{i=1}^{N} f_{ik}$$

and similarly the document frequency, B_k, which is the number of documents in a collection to which a term is assigned:

$$B_k = \sum_{i=1}^{N} b_{ik}$$

where b_{ik} is defined as 1 whenever the corresponding f_{ik} is greater than or equal to 1 and b_{ik} is 0 when f_{ik} is 0.

The inverse document frequency postulates that a good term exhibits a high occurrence frequency in a specific document and a low collection frequency or document frequency. This leads to the function

$$w_{ik} = f_{ik}/B_k$$

where w_{ik} represents the weight of term k in document i. For constant values of f_{ik}, the weight of a term will vary inversely with its document frequency B_k.

When an IR-system is used to query a collection of documents with t terms, the system computes a vector Q with terms(q_1,q_2,...q_t) as weights for each term. The retrieval of document D_i with document vector (d_{i1}, d_{i2},...,d_{it}) may be effectuated by a similarity function like

$$sim(Q,D_i) = \sum_{j=1}^{t} w_{qj} \cdot d_{ij}$$

b. The signal-noise ratio.

A related way to decide on the relative weight of a keyword on the basis of its frequency uses information theory. It is akin the intuition that the higher the probability that a word occurs, the less information it contains. The information content of a word then is INF=-log2p, where p is the probability of the occurrence of the word. This gives us a measure of reduced uncertainty, because every term we assign to a document, decreases the uncertainty about its contents. So if a document is characterized by t possible keywords, each of which has the probability p_k, the average reduction of uncertainty about the document is

$$AVERAGE\ INF=-\sum_{k=1}^{t} p_k \log p_k$$

and the noise of an indexterm k for a collection of n documents may be expressed as

$$NOISE\ k=\sum_{i=1}^{n} \frac{FREQ_{ik}}{TOTFREQ_k} \log_2 \frac{TOTFREQ_k}{FREQ_{ik}}$$

This covers the intuitive notion that a word, that is distributed evenly over the database, i.e. occurs an identical number of times in each document, is a bad keyword. The noise is maximized in such cases. On the other hand, if a keyword only occurs in a single document, the noise is zero.

1. 6. 2. The title-keyword approach and the location method.

Words in titles of documents, chapters and paragraphs are 'heavier' compared to words in the middle of the text (Edmundson). This observation is akin to the observation that in a paragraph the first sentence is usually the most central to the text [Baxendale, 1958]. Edmundson elaborated on this principle and research by [Kieras, 1985] confirmed the psychological assumptions. Both methods seem to fit in the approach taken by McLeod [op.cit] and Burkowski [op.cit], who, as we have seen in the previous chapter, divide documents in logical parts similar to the division of a fixed format record in fields and subsequently try to use these logical parts in a kind of field control.

1. 6. 3. Syntactic criteria.

An hypothesis of Earl was that the weight of a sentence might be correlated with its syntactic structure. Experiments conducted by her, however, did not bear this out [Earl, 1970]. Earl herself expressed disappointment about the results of her study and the general feeling is that this approach cannot lead to substantial results.

However, Earl worked with complete sentences. It is generally accepted that different parts of a sentence do have different weights, hence the fact that almost all sophisticated indexing methods try to limit themselves to NPs and, indeed, the identification of NP's consisting of more words, is a major concern (e.g. CLARIT). It is tempting to think that NPs that are part of a modifier, e.g. a prepositional

phrase do have different weights than NPs that are the head of a phrase, although we never saw research in this direction.

1. 6. 4. The cue method and the indicator phrase method.

The cue method and the indicator phrase method are very similar in that they signal important sentences by cue words or phrases like "our work", "purpose", "The main aim of this article is...". Words following these cues are weighted accordingly. Compared with the three other methods mentioned above, this last method may lay claim to the fact that it uses semantics, be it in a crude way.

1. 6. 5. Relational criteria.

Skorokhodko proposed a very interesting method of weighing sentences. He proposed the creation of a 'semantic structure' for the document, in which the relations between the sentences are visualized in a graph, with the sentences as the nodes and the inter-sentence relations as arcs. The number of arcs, that meet in a node is the weigh factor for the sentence; sentences are related when they contain references to the same concept.

The relations between the sentences, i.e. the question if they refer to the same concepts, is decided on word-word similarity or by using a thesaurus. Nevertheless here as in other NL-applications the solving of anaphores is crucial.

2. Retrieval with weighted terms.

Using any of the weighing strategies mentioned, we may construct an inverted file of keywords and/or keyphrases. Retrieval of documents becomes a matter of predicting which keywords are used in exactly the documents we are interested in.

Modern computing and storage techniques have created the possibilitiy of adressing hundreds of megabytes of text on-line and orthodox inverted file systems will inevitably break down when confronted with even smaller quantities [Blair/Maron, 1985]. If the reason for such breakdowns is the futility point or predicted futility point, ranking and weights may offer a solution; the other solution lies in the creation of knowledge representations, such as a thesauri.

2. 1. TOPIC

An approach combining both is offered by the RUBRIC-system [Cune/Tong/Dean, 1985], which evolved into a commercial system called TOPIC[1].

The RUBRIC/TOPIC system essentially is an front-end to full text databases of the type in which each wordtoken and its location in the text exists in the document representation. Orthodox fields with formatted information are accessible too.

1 These names cause no end of confusion, because there exists another system called TOPIC [Hahn/Reimer, 1988] and another RUBRIC [Loucopoulos/Layzell, 1989], both systems, that are also adressing problems in Information retrieval, but not connected with each other or with the TOPIC-RUBRIC pair mentioned here. The second RUBRIC as described by Loucopoulos, will not concern us here, but as we will consider both TOPIC-systems, we will use TOPIC, RUBRIC/TOPIC or just RUBRIC when talking about the system described by Cune, Dean and Tong and use the adjective *German* for the TOPIC of Hahn and Reimer, whenever confusion may occur.

```
Team | event => World_series
St._Louis_cardinals | Milwaukee_brewers => team
"Cardinals" => St._Louis_Cardinals (0.7)
Cardinals_full_name => St._Louis_Cardinals(0.9)
Saint & "Louis" & "Cardinals" => Cardinals_full_name
"St." => saint(0.9)
"Saint" => saint
"Brewers" => Milwaukee_Brewers (0.9)
"Milwaukee Brewers" => Milwaukee_Brewers(0.9)
"World Series => event
baseball_championship => event (0.9)
baseball & championship=> baseball_championship
"ball"=> baseball(0.5)
"baseball" => baseball
"championship" => championship (0.7)
```

VI.4. RUBRIC's rulebase for topic of world_series

However, the normal querying of the database by ad hoc boolean combinations of keywords is replaced by a system, where the burden of building a knowledge representation is on the user. This is effectuated by enabling him to build '*topics*', essentially self-made thesaurus entries, where concepts in the documents are characterized by the occurrence of keywords combined by various operators admitting the attachment of weights to the individual keywords (fig. VI.4.).

When processing a query like the *topic* world_series as above, RUBRIC searches the rulebase for all definitions of this topic, finding *team* and *event* as definitions. It then recursively searches all definitions until every leaf-node of the tree contains textual patterns.

Following this activity a calculus is applied to the weights, that in the figure are shown as reals between 0 and 1 between parenthesis. It then ranks the documents found according to these figures and presents them to the user.

These topics may also be built by experts in the domain of the database and they may freely be shared among the users. The net effect is that RUBRIC/TOPIC acts as a kind of SDI (Selective Dissemination of Information) system, suited for queries that evolve over longer periods of time in a rather constricted domain, rather than an all-round query system for big databases and general libraries.

EssentiallyRUBRIC organizes well-known retrieval tools as weighting and various operators in a new way. Enthousiast results are published and its commercial successor, TOPIC is rapidly becoming a very popular full text database management system. However, it is felt by the author of this memo that the tests and reports so far seem to labor under a fallacy: knowledge of the contents of the database seems to be necessary before the weights can be built in. Queries and topics may be superficially almost identical, but may return different documents

again depending on the weights that the individual user attaches to them. So if the user is not already an expert with extensive knowledge of the underlying database, his IR results may very possibly have the same bad precision and recall ratio as the more orthodox retrieval systems.

The keyword systems of the previous chapter, with or without sophisticated user-ends such as RUBRIC, will remain the backbone of information retrieval in libraries for a long time, by virtue of the fact that the information need of the users is very unpredictable and the domain in which the questions are put, is very wide. There are other environments, where either this information need or the underlying domain (or both) is more circumscribed and here more involved document representations may be created.

3. Phrase indexing.

It was long felt by various scientists that separate keywords are very inefficient vehicles for the "aboutness" of documents. "*An obvious shortcoming of the document representation models used in most automatic systems is, that the contents of each document is represented by an unstructured collection of simple descriptors*" ([Fagan, 1989], p.115). However, if the keywords could only be replaced by key-phrases, a drastic improvement was expected. This expectation led to attempts to select those combinations of words that together embodied a concept not present in the separate words. To do this automatically robust parsers were needed and the last few years have seen several systems that use a partial grammar with which to extract phrases from the document. We will shortly describe two of such systems: CLARIT and TINA.

3. 0. 1. CLARIT

To conclude this section about weighing keywords and keyphrases, we will describe the CLARIT system: an IR system that perhaps incorporates the most sophisticated handling of keywords and keyphrases so far. We have referred to the CLARIT system before, in chapter V, as an example of a system that uses syntactical parsing to extract interesting parts of the document, compares them with a reference set of certified terminology and so identifies important concepts in the texts, i.e. keywords and keyphrases.

The system adresses itself to the problem of identifying first-order concepts. i.e. the recognition of morphological, lexical and semantic variants of terms. The processing of a document by CLARIT, a process that ends in a list of index terms, consists of three stages: formatting, natural language parsing and filtering, of which especially the third stage interests us here.

The formatting as described by Evans [Evans, 1990] consists essentially of a normalization of the texts offered to the system and exhibits no special traits. The natural language parsing too is aimed at a robust extraction of NP's, identifying the constituents in their roles as heads and modifiers and providing '*information-theoretically useful*' parses, not necessarily syntactically accurate ones.

The following stage, filtering, starts with applying several measures of frequency, distinguishing 'minimal' documents (queries, phrases, single sentences) from short documents (abstracts) and long documents such as papers or book chapters. In setting values for words in documents the expected frequency of the word in a domain (as represented by a corpus) is taken into account, scoring words in short document according to the domain frequency and words in longer documents according to both the document frequency and the domain frequency.

In this way a list of candidate terms is generated. The next step consists of a matching procedure, which compares these candidate terms to a set of certified terms. This process classifies the candidate terms in three groups: *exact* matches, that are identical to terms in the certified list; *general* matches, that may be traced to constituents or sub-terms of the certified list and finally the *novel* terms, which consists of the general matches and the group of words that could not be matched at all.

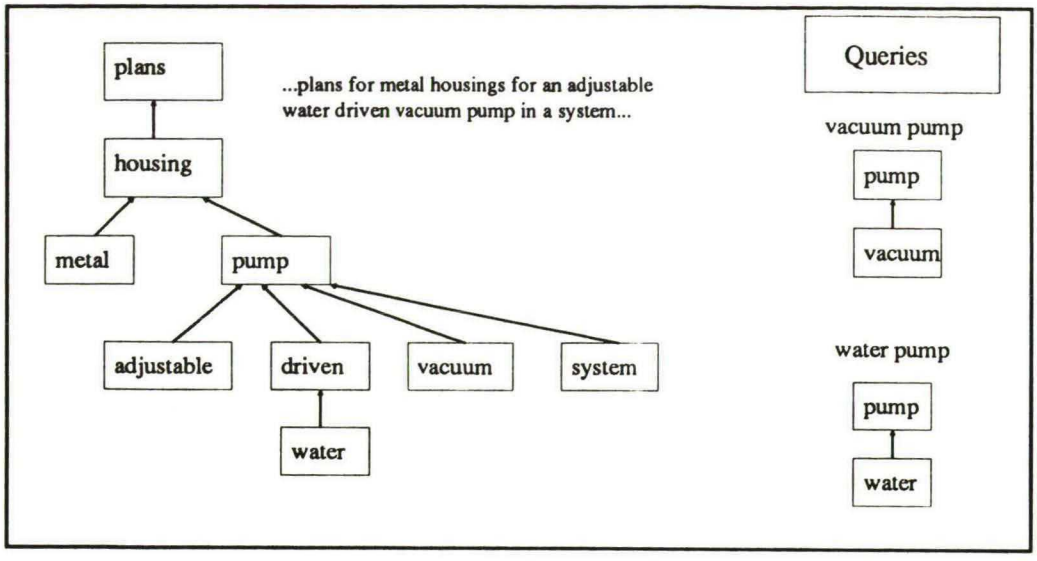
The matching itself takes the form of comparisons of all the permutations of adjacent sub-terms (windows: thus the term ABC consists of the windows (A,B,C,AB,BC,ABC) in the candidate phrase to all windows in the certified list. An exact match of course is when candidate term ABC finds a certified term ABC. A general match is when the candidate term is a window of the certified term. Novel terms are again compared to the certified list, to see if other combinations of windows match. Evans reports considerable success in creating keyword and keyphrase representations of documents in this way.

3. 0. 2. TINA.

A second system that uses parsing to arrive at a meaningfull representation of documents is TINA[Ruge/Schwarz/Warner, 1990]. It is similar to CLARIT in that TINA too uses robust text parsing to extract phrases from the text. Although this system too uses the head-modifier relation to collect related words, the parser seems a bit more sophisticated than the one used in CLARIT. The big difference though is that the result of the TINA processing is not subsequently compared with reference sets or 'windowed' into a thesaurus, but that the dependency structures were stored 'as is'. The dependency structure of a natural language question then is matched with the stored structures.

The result consists of a ranked set of document classes, in which not only the number of matched words are taken in account, but also the number of links. In figure VI. 5 we see at the left a part of a document about a water driven vacuum pump and at the right two questions for respectively a vacuum pump and a water pump. Although all three words from the question, water, vacuum and pump do all occurr in the document, the first query matches also the direct arc from vacuum to pump.

TINA builds and stores a representation, not only of the document in terms of tokens and statistical properties, but also of the semantic relations between the tokens. It therefore properly belongs in the next chapter. It is included here with CLARIT to mark the transition from document representations to document knowledge representations.



VI.5. Dependency tree and query.

3. 0. 3. The Semantic Enhancement Experiment.

A third and rather interesting experiment was recently published [Wendlandt/Driscoll, 1991] in literature. No name for the system was given; we will therefore refer to it as the semantic enhancement experiment (SEE), from the title of the article.

This system too crosses the border between document representations and document knowledge representations, but it too is mentioned here, because of the adherence to the probabilistic departure point, described in this chapter.

The system centers on the thematic roles that words in a text may occupy. Rather than using syntactical parsing to extract the exact thematic role of a word in a document, the thematic categories and related keywords are given a probability for keyword *k* triggering category *c*. For example, the DESTINATION (fig, VI.6) category or *thematic role* triggered by the word 'to' has a probability of

$$P_{DESTINATION,to} = 0.33$$

On this basis a number of probabilities are computed, analog to the frequency formulas that have been mentioned before, e.g. the inverse document frequency of category c_j for a set of N documents:

$$idf_j = \log\left(\frac{N}{edf_j}\right)$$

In this system the typical function words (*how, does, the, through, before*) are not, as in most other IR-systems, discarded in an early stage of processing, but also used to infer thematic information. The final computation of the similarity between query and document *i* thus has the form

Word	possible thematic role triggered
by	CONVEYANCE, INSTRUMENT, LOCATION
carry	LOCATION,none
in	DESTINATION, INSTRUMENT, LOCATION, MANNER, PURPOSE
into	LOCATION, DESTINATION
to	DESTINATION, LOCATION, PURPOSE

VI.6. Thematic categories

$$sim(Q,D_i) = \sum_{j=1}^{t+s} w_{qj} . d_{ij}$$

If we compare this formula to the related formula on page 76. We see that the category weight s is added to the term weights t of each document vector D_i and the query vector Q.

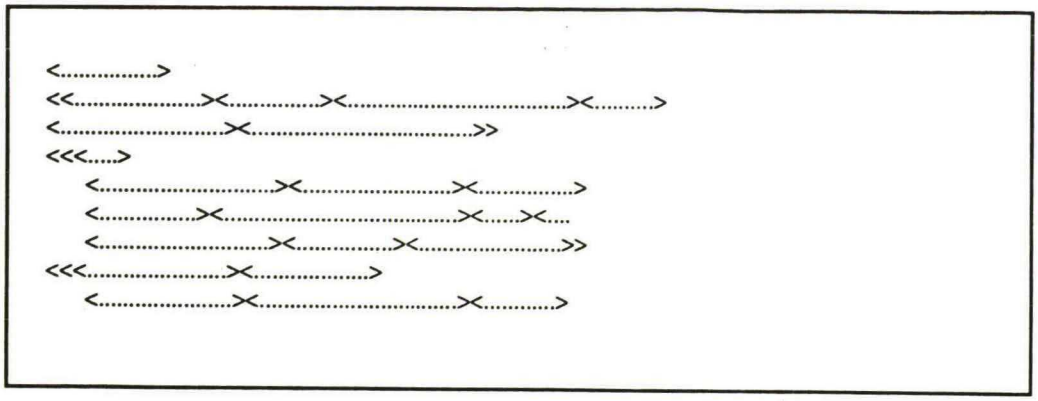
4. Representation by extracts.

Extracts and abstracts try to capture the essentials of the document in a form that is in itself 'readable'. This necessitates additional processing to ensure this readability. The easiest way to create readable docreps is the creation of *extracts*, which consist of sentences and/or phrases that are extracted from the original document according to some selection scheme. It is understood that these sentences and phrases keep their original sequence. To cite Earl: *"An automatic extract can be defined as a small number of sentences chosen from a text by a computer and presented in the order of their original occurence"* [Earl, 1970].

Closely related to the extract, but created by totally different procedures, is the *abstract*, which is a complete reformulation of the contents of the document. To do so a representation has to be created of the relevant contents of the document, that in its turn has to be formulated in natural (or at least understandable) language by a NL-generator. Because of the evident need for a knowledge representation when generating abstracts, we will not consider abstracts in this chapter.

The extraction of important sentences using one form of weighing or another generally produces texts which lack cohesion. The generating of a document-extract exists of the weighing of the sentences, using one of the methods or a combination of the methods mentioned above and discarding sentences that remain under a treshold. The list of sentences that is the result of this process generally has a lot of dangling anaphores, so the next step is the backward resolution of these anaphores and adding some sentences, that were initially thrown out, but which have the antecedent of an anaphore in a 'heavy' sentence in it [Bonzi/Liddy, 1989].

Blind application of this procedure might lead to a reconstruction of almost all of the original document: one might wonder if the antecedental sentences (sometimes called kataphores) would not have been selected by the weighing algorithm in any



VI.7. Docrep after total subtraction.

case and that forward resolution would not implicate the reconstruction of the original document.

There are three problems inherent in the extraction of meaningful sentences: the first is the problem of deciding what which properties of the document should be present in the docrep, secondly the devising of a weighting method that differentiates between sentences on this properties and last but not least the solving of the anaphores. The first problem is one of the fundamental problems of the Information Retrieval; the second and third are very much linguistic problems.

An extract is very much like the skeleton of the original document and should exhibit the same structure as the document. Nevertheless we will group it under the non-structured representations, because the structure is not made explicit. In contrast with the extract we have the abstract, which is a reformulation of structures, which themselves have been made explicit, or the table of contents (TOC) which also makes structures explicit.

5. Subtraction

If indexing concentrates on the words in the document and their meaning, subtraction moves the opposite way. In stead of making the info that adheres to the words explicit, it subtracts the word from the documents, in the process uncovering structures in the documents that do not relate to the semantics of the words. We discern two types of subtraction, the semantic subtraction and the total subtraction.

As we have said before, the residue remaining after subtraction does not carry much information in the sense of 'aboutness'. But the structures that emerge, may be stored separately thus facilitating access for other enrichment techniques that may need it.

5. 1. Semantic subtraction.

The first step towards the subtraction of words in the document would be the discarding of the semantic contents of the word, preserving its syntactic properties.The document then would consist of a list of representations of phrases.

5. 2. Total subtraction.

The ultimate subtraction would bring us to a document representation in which all words would be omitted or replaced by a single symbol.

VII. Document Knowledge representations.

Until now we have been talking about sets of keywords and keyphrases as document representations. It should be clear that there is no such a thing like an unique representation of the *meaning* of a document, but that (from an information retrieval point of view) meaning is always related to the information need of the user. Any representation of a document should make explicit the essentials of that document as seen from the standpoint of the predicted user.

So the expected questions, or other formulations of the user information need, are factors of foremost importance when choosing among the possible document representations. Analysis of possible questions may even discover the need for new document representations, hitherto unthought of.

1. Understanding a document.

Creating a representation of a document might be called 'understanding' the document. Turning this around, one might ask: what is *understanding a document*?

A person who understood the contents of a document should be able to do at least one of the following things:

1. Answer questions about the document.
2. Create a paraphrase of the document.
3. Create a summary of the document.

Early experiments were SAM (Script Applier Mechanism), FRUMP (Fast Reading Understanding and Memory Program) and BORIS.

SAM operated with a knowledge base of scripts, thus enabling the system to make inferences about events that were not explicitly mentioned in the input story.

FRUMP too had a knowledge base of scripts. The system tries to match incoming news stories (from a telex) with these scripts and thus to generate summaries of these stories, using templates for each group and filling them with appropriate values. Almost all later attempts to create DK representations depart from script-like structures as a semantic background against which to develop representations of the documents. But before embarking on the description of a few of these systems, we will have to look at another approach to the problem of meaning: the strengthening of the keyword and keyphrase systems by applying additional knowledge at query time.

1. 1. Thesaurus

A major problem in the questioning of *derived* inverted file systems is the number of synonyms, near synonyms and related terms, that in natural language may be

used to refer to concepts; a problem that perhaps does not loom so large when using *assigned* inverted files, but nevertheless is real even in those systems. When assigning keywords to documents the need was felt for standards in the use of those keywords and keyphrases (controlled dictionaries) and after that standards in the relations of those keywords to each other. Lists that made those relations explicit were called *thesauri*. Such thesauri (see fig. III. 4), that originated as a help for assigned indexing, now generally are accessible at query time as external knowledge structures, that suggest alternative keywords while searching in inverted files or even expand or restrict the set of keywords in the query automatically.

The thesaurus in IR is almost always domain-bound and compiled by experts in that domain, thus reflecting human knowledge [Lancaster, 1972]. A few pages back we have reported about attempts to create thesauri by computer and using statistical methods or cluster representations; also we have mentioned the manner, in which a first-order thesaurus is built by the CLARIT system.

1. 2. RESEARCHER.

A typical instance of a restricted domain as mentioned above would be a patent's office, where detailed and exact descriptions of physical objects are managed. In [Lebowitz, 1986] an experimental information system for such an environment is described. This system, RESEARCHER, fulfills the following tasks:

1. 2. 1. Building object representations.

Starting from a natural language text (patent abstracts) describing a physical object, it builds a representation of that individual object. Such an activity needs "*a level of understanding that many artificial intelligence text processing systems might achieve if applied to this domain*" (Lebowitz, op.cit. p.130). Implicitly he refers at earlier work of his own: the Integrated Partial Parser (IPP). This IPP is a typical top down, expectation driven parser, that tries to recognize parts of the sentence that fit in its frames. As RESEARCHER is not primarily interested in the documents, but in the objects, these representations of course are called object representations (although pointers to the original documents or patents are attached), limiting themselves to physical descriptions.

This IPP is a typical top down, expectation driven parser, that tries to recognize parts of the sentence that fit in its frames.

1. 2. 2. The RESEARCHER Document representations.

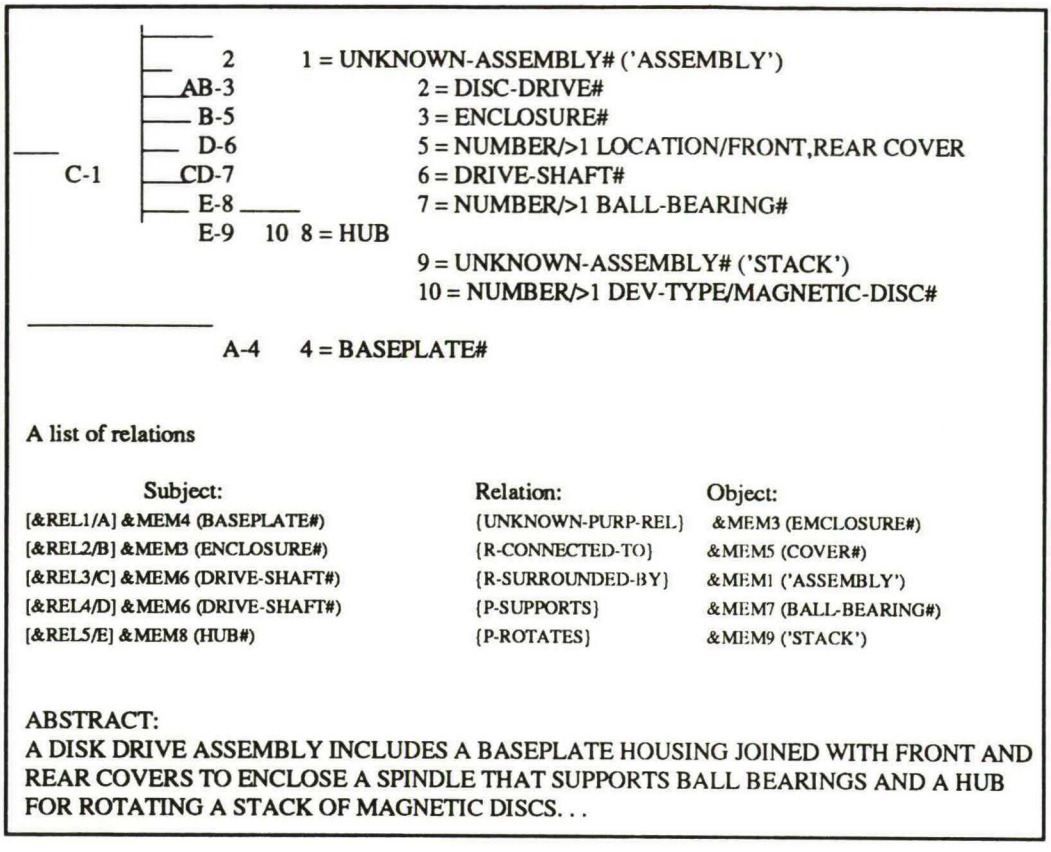
This representation includes three classes of information (see fig.VII, 2):

- a *parts hierarchy*, that recursively indicates the components of each part.
- *interpart relationships*: the physical and functional relations between the components.
- *properties* of the object.

The exact handling of this third class of information is still under consideration by Lebowitz and his collegues and is not present in the figure.

1. 2. 3. Storing the generalizations.

The second task of the system consists of comparing the representations of different objects and abstracting out similarities, thus creating *generalization trees* and adding this information to long-term memory. The contents of this memory



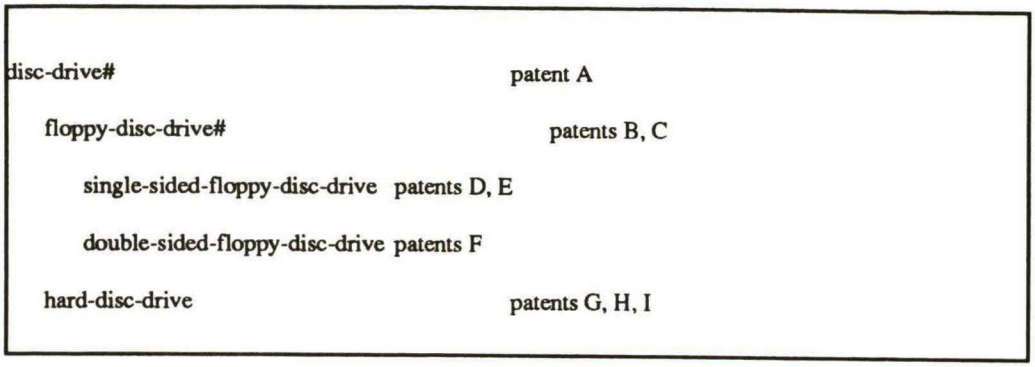
VII-1. A typical RESEARCHER representation.

itself are organized as hierarchies of hierarchies, so that every generalization is subordinated to another yet more general concept (see fig.VII.3). Every node in the figure is a complete hierarchy description and information in the gemneralizations can be inherited by lower levels. In RESEARCHER - at least in a fully developed system - this hierarchy is automatically created, not provided to the system in advance.

For every new object to be processed by RESEARCHER, the tree is searched for the generalized concept most similar to it, using either best a match or a treshold match algorithm. According to Lebowitz only experimentation will in the long run decide which algorithm gives best results. When it has found a best match, RESEARCHER will factor out the similarities between the object under consideration and the existing generalization and, if need be, create a new generalization node. The current implementation is reported to work quite well on modest-sized examples, both in the patent's domain and in a domain of hierarchical descriptions of corporate organizations.

1. 2. 4. Text processing using memory.

When the system has access to a collection of such texts, its representations and generalizations, it seems natural to use this information for the processing of new texts. RESEARCHER tries to solve ambiguities by comparing the object under



VII. 2. A typical generalization based memory.

consideration with the objects already stored in the generalization hierarchy. If several examples seem to match, the system selects the most general and tries to solve the ambiguity by comparing the two objects.

1. 2. 5. Question answering.

The search for possible examples that answer a given question in such an object-hierarchy is relatively simple, both when searching for a main concept (e.g. the disc-drive) and subsidiary parts (head-assemblies, discs). An interesting part of the RESEARCHER-system is the idea that different users may get answers, taylored to their level of expertise (expert or naive). It was found that e.g. encyclopaedia's that are aimed at adults and relative experts tend to describe the part structure of an object, while a childrens encyclopaedia would describe the same object in a process-oriented manner.

1. 3. SCISOR

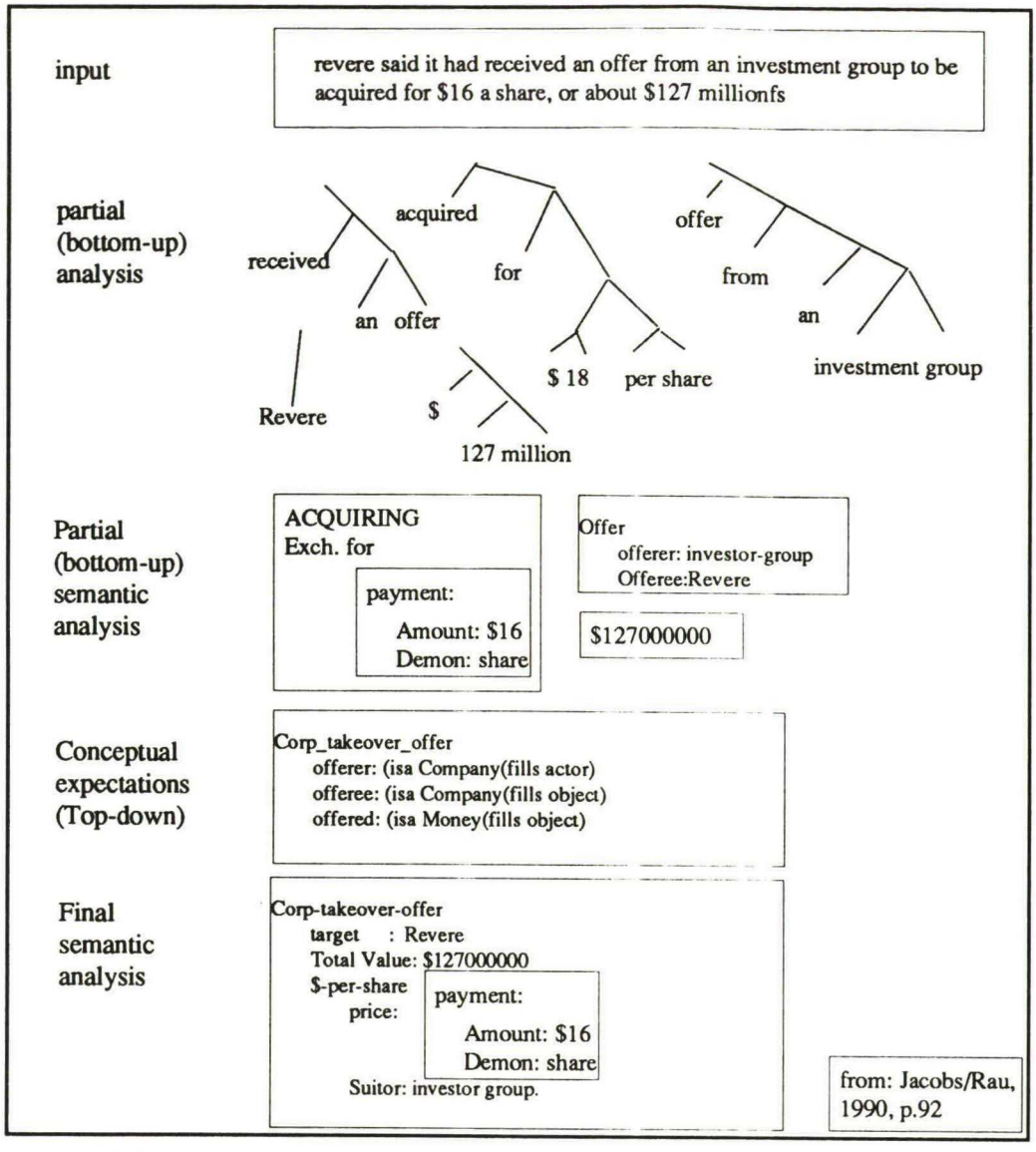
A similar approach is exhibited by the somewhat younger SCISOR-system. Developed at General Electrics it is an experimental system, that detects and stores information about financial transactions, such as mergers, takeovers etc. in an input stream of financial news (The Dow Jones). Subsequently it answers questions about this domain.

SCISOR (System for Conceptual Information Summarisation, Organisation and Retrieval) essentially does three things:

1. 3. 1. Selecting the stories that fit the domain.

The system analyses the input stream to decide whether the incoming stories are about its domain of corporate mergers and takeovers. To achieve this goal it first does lexical analysis on each story and separates it in differentiated structures: header, byline and dateline designations. It then passes the story through a number of sieves, each trying to decide if the story is definitely about the merger/take-over domain, definitely not about this domain or if there still is doubt left. In the latter case it is passed to the next sieve.

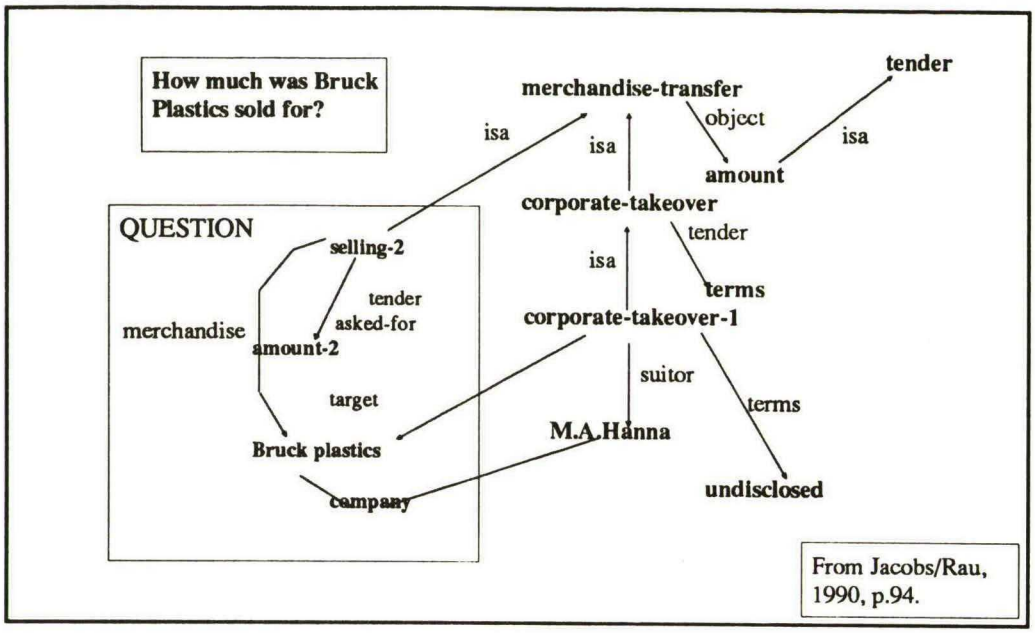
The sieves start with rather coarse filtering on headlines and keywords, becoming more sophisticated and thus more expensive later on. This arrangement ensures that the expensive techniques only have to be called in on a subset of the documents.



VII.3. Integration of bottom-up and top-down analysis.

The modular architecture of this arrangement also makes it easy to plug new algorithms in or out, making comparisions between them relatively easy.

If we look at the results of this layer of sieves in terms of recall and precision, we come to the rather appaling conclusion that in such a controlled environment, where the incoming documents already belong to a rather narrow domain (financial news) only 90% combined recall and precision is attained in the selection of documents about mergers and acquisitions [Jacobs/Rau, 1990, p.91].



VII.4. Answer retrieval in SCISOR

1. 3. 2. Creation of a conceptual representation.

A natural language analysis is done on the stories thus selected, which exists of an integration of both bottom-up linguistic parsing and top-down conceptual analysis. The bottom-up parsing identifies linguistic structures and tries to map these in a conceptual framework; top-down analysis tries to fit partial information from the text in conceptual expectations (see fig. VII.4.).

1. 3. 3. Storage and retrieval of the representation.

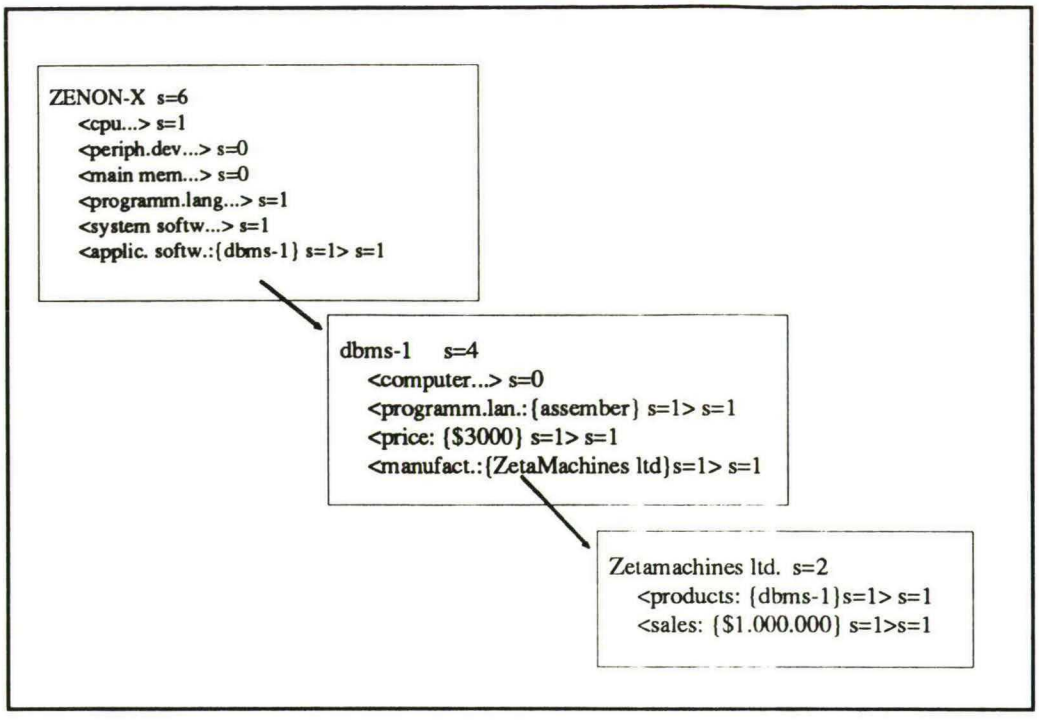
The conceptual representation of the story that is created in this way, is stored and retrieved into and out of a knowledge base. SCISOR stores the conceptual representation of the story as a network of unique instances,i.e. individual members of conceptual categories. The answering of questions becomes the reporting on slots.

SCISOR is fundamentally different from RESEARCHER in the fact tat SCISOR has a rich hand-coded knowledge base as the backbone of the system, while RESEARCHER emphatically tries to construct (or augment) such a knowledge-base from the documents it reads.

1. 4. The German TOPIC.

Although the German TOPIC has the same name as the commercial descendant of RUBRIC, mentioned in the last chapter, it is an totally different system and belongs to those systems that create a real Document Knowledge Representation in the same vein as RESEARCHER and SCISOR.

Similar to RESEARCHER, concepts in the document are translated to a hierarchical knowlegde representation of frames. The extraction of knowledge is driven by script-like structures and controlled by socalled *word experts*, that apply



VII.5. Significant degrees of slot occupancy and nesting

grammaticalconstraints to the matching of text-items to frames and connected structures. But TOPIC also counts the references to these structures, not as in orthodox statistical systems do by occurrences of word tokens, but by actual references to the frames, slots and slotfillers that constitute the knowledge representation of a TOPIC database.

The combination of this hierarchical knowledge structure and the activation weights assigned to the various structures and substructures becomes a powerul tool for text summarization and the determination of dominant concepts in the text. First those concepts that are significantly important or dominant are decided upon, which step is followed by the recombination of those dominant concepts to form a condensation or summarization of the original text.

1. 4. 1. Identification of dominant frames.

A major measure for identifying an important concept in a text is the activation weight attached to the relevant structures. Since in TOPIC these weights are adjusted both by the explicit and inplicit occurrence (e.g. by resolving of anaphora) of the concept in the text, they are better indicators of the importance of the concept than plain word occurrence (for the relative importance of anaphora in the weighting of keywords see [Bonzi/Liddy, 1989]).

A slotfiller is considered as a dominant, if any of these conditions is true:

1. The particular slotfiller has a significantly higher activation weight than the average slotfiller.
2. A slot is taken as dominant if significant more slotfillers are assigned to it than to other slots. Measures will have to be taken to account for structural biases inherent to a concept, e.g. the CPU-slot in a computer-frame will only have one possible slotfiller, but the peripheral devices for that same kind of computer may have any number of potential fillers.
3. A more advanced criterion of concept dominance is the *slot occupancy* and the *depth of nesting* of the slot fillers. See for instance fig. VII.6, where frames as slotfillers are nested. Accordingly a slot is considered dominant if a frame is assigned to it such that the majority of its slots have been filled too (i.e. a significant degree of occupancy), or if a slotfiller exists that is elaborated in more detail.

A further measure of importance was investigated by Hahn and Reimer: the role of connectivity patterns

4. A number of active frames with a common superordinate frame may constitute a cluster of frames. This superordinate frame is called the cluster frame, but it does not have to be active or even be mentioned explicitly in the text. Cluster frames are detected by recursively searching downwards from the most general concepts as long as no significant loss of active concepts occurs (according to an empirically chosen treshold or when the summed activation weight of the frames drops below a certain level).

1. 4. 2. Topic descriptions.

The dominance measures result in a collection of formally unconnected concepts, which may be represented as linear graphs. Complete descriptions of topics ares arrived at by checking for overlapping nodes of the same type, but occurring in different descriptions, adding links where possible. The result is a text graph, which allows flexible, content-oriented access to full-text information.

TOPIC as yet has no full-fledged natural language generator. Emphasis currently is on an interactive graphical retrieval interface.

2. Connectionism.

A survey of information retrieval would not be complete if no mention was made of the attempts to use the connectionist approach to the problems of aboutness and documentmatching. In an experiment by Belew [Belew, 1989] it is shown how a connectionist approach is taken to information retrieval, that is, to a specific aspect of the discipline.

In this system, AIR, ways are explored to improve the performance of retrieval by changing its document representation, using relevance feedback from the users of the system. It operates on a database of bibliographic citations; each document is represented by its title, its author(s) and a number of keywords or descriptors. In the experiment described here, the keywords are taken from the title.

To start with, a representation of the information in the database is built by creating nodes for all documents. These nodes are connected with the nodes for the authors (one for every author) and the nodes for the keywords (one for every keyword). The links are weighted according to an inverse frequency weighting

scheme. The sum of all the weights departing from a node is forced to be constant.

If a query is put to the system, "activity" is placed on all nodes that correspond to that query and the answer of the system is ranked according to the activation of the nodes and presented to the user. Now the user may indicate which nodes he considers relevant and which are not. The system creates a new query based on this feedback, strengthening or weakening the links, and so is effectively trained by the user to recognize associations that are useful for IR.

If a query contains a new term, i.e. one for which no node exists, the query is first handled without that term and subsequently (after the user's response) a new node is created for that term and connected to the network.

The net result of all this is that the network will evolve towards a consensus of users about what keywords and documents belong together. This 'democratic' view of the aboutness of documents contrasts with the omniscient notion of aboutness, that is present in almost all other IR-systems. That is: the relevance of a document with respect to a query in orthodox IR systems tends to be absolute, as if determined by a omniscient indexer. In a system.

1. Bibliography

ANSI:" American national standard for writing abstracts." ANSI Z39.14- 1979

Attig, J.C.:" The concept of a MARC format" Information technology and libraries 2 (March 1983): p.7-17

Baars, C.G.H.; Schotel, H. :" Natuurlijke taal en databases" A.I.T. 1988

Bar-Hillel, Y.:" The mechanization of literature searching." in: National Physical Laboratory: Proceedings of a symposium on the mechanization of thought processes 2, 1959

Bates, M.A.:" Information search tactics." Journal of the American society for Information Science 1979, No 4. 204-214

Baxendale,P.b. :" Man-made index for technical literature - an experiment. " I.B.M. jopurnal of research and development,2 pp.354-361; 1958

Benschop, C.A.; Heer, T. de:" Voortzetting van het informatiesporen onderzoek" IWIS/TNO 1980

Blair, D.C. :" Searching biases in large interactive document retrieval systems." Journal of American Soc. for Information science, 31:4. p.271-277, 1980

Blair, D.C.:" Language and representation in information retrieval" Elsevier, Amsterdam 1990

Blair, D.C.; Maron, M.E. :" An evaluation of retrieval effectiveness for a full-text document retrieval system" Communic. of the ACM V28:3 pp.289-299, 1985

Borko, H.; Bernick, M.:" Automatic document classification." Journal of the association for computing machinery, 1963, p.151-162.

Brauen, T.:" Document vector modification" in: Salton(1971) :509:

Burkowski, F.J. :" The use of retrieval filters to localize information in a Hierarchically tagged text-dominated database" RIAO91, p.264 - 284

Burnard, L. :" The text encoding initiative" Oxford University computing service g.j.

Codd, E.F.:" Normalized data base structure; a brief tutorial" Proceedings of ACM SIGFIDET workshop on data description access and control 1971 pp. 1-16

Chudacek, J.:" Least effort text-retrieval definitiestudierapport." IWIS/TNO 1983

Chudacek, J.:" Statistische en organisatorische eigenschappen van trigrammen in natuurlijke talen." IWIS-TNO Den Haag 1983

Cleverdon, A.W. :" Optimizing convenient on-line access to bibliographic databases" Inf. Serv. Use 4 pp.37-47, (1984)

Cleverdon, C.W.; Keen, E.M. :" Aslib-Cranfield research project" Cranfield institute of technology, Cranfield, England 1966

Conklin, J.:" Hypertext: an introduction and survey." Computer, September 1987, p.17

Daelemans, W.:" Studies in Language technology: an object-oriented computermodel of morphological aspects of dutch"

Date, C.J. :" An introduction in Data-Base Systems" Addison-Wesley, 3th ed. 1981

Davies, R.:" Classification and ratiocination: a perennial quest." Davies '86 :745:

Davies, R.:" Outlines of the emerging paradigm in cataloging" Information Processing and Management 23(2),p.89-98, 1987

Davies, R.D.:" Intelligent information systems; progress and prospects." Horwood ltd. Chicester 1986

DeJong, G. :" An overview of the FRUMP system"

Dijk, A.; Swede, V. van; Visser, J.S. :" Taalkundige informaticsystemen ontwikkeld met GRAMMARS" Pandata, Rijswijk, 1989

Earl, L. :" Experiments in automatic extracting and indexing" Information storage and retrieval 6 pp.313-334. 1970

Edmundson, H. P. :" New methods in automatic abstracting" Journal of the ACM 16, pp.264-285; 1969

Enser, P.G.B.:" Experimenting with the automatic classification of books" Aslib 1985 :650: p.66-83

Evans, D.; Ginter-Webster, K.; Hart, M.; e.a.:" Automatic indexing using selective NLP and first-order thesauri" Manuscript 1990 (?)

Evans, D; e.a. :" Computational-Linguistic approaches to Retrieval and Indexing of Text: The CLARIT project." Carnegie Mellon University 1989

Evans, David :" Natural Language in Document Retrieval Systems (Abstract) " ITK Colloquium Series March 26, 1991

Fagan, J.L.:" The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval" JASIS 40(2):115-132, 1989

Faloutsos, C.; s. Christodoulakis:" Signature files: an access method for documents and its analytical performance evaluation" Transact. on Office Autom. systems, Vol.2 No.4, Oct.1984, pp 267-288

Faloutsos,C. :" Access Methods for Text" ACM Computing Surveys (New York, NY) 17 (1986.03) nr.1 p.49- 74 (100 refs.)

Files, J.R.; Huskey, H.D.:" An information retrieval system based on superimposed coding" Proceedings Fall Joint Computer Conference; reston Va. 1969

Foskett, A.C.:" The subject approach to information" London 1969, 4th edition 1982

Gordon, M.; Kochen, M.:" Recall-Precision trade-off: a derivation" Journal of the American society for information science 40(3):145-151, 1989

Graesser, A.C.; Black, J.B. (ed):" The psychology of questions" Lawrence Erlbaum 1985

Grosz, B.; C.L. Sidner :" Attention, intentions and the structure of discourse" Computational linguistics vol 12. Jul-sept 1986

Guthe, C.E.:" The management of small history museums" Nashville 1964

Hahn, U.:" Topic parsing: accounting for text macro structures in fulltext analysis-" Information Processing and Management vol.26, p.135-170, 1990

Hahn, U.; Reimer, U.:" Knowledge based text analysis in office environments: The text condensation system TOPIC" Lamersdorf (ed): IFIP conference proceedings of 1987, North Holland 1988 :687:

Harrison, M.C.:" Implementation of te substring test by hashing." Commun. of the ACM 14. Ppp. 777-779, 1971

Holstege, M.; Inn, Y.; Tokuda, L. :" Visual parsing: an aid to text understanding" RIAO91, p.175-193. 1991

Jolley, J.L.:" Information Handling: Einfuehrung in die Praxis der Datenverarbeitung" Fischer Taschenbuch Verlag 1968

Kieras, D.E.:" Thematic processes in the comprehension of technical prose" Britton&Black, p.89-108 :891:

Lancaster, B.C. :" The measurement and evaluation of library services" Washington 1977

Lancaster, F. W.:"Vocabulary Control for Information Retrieval", Washington D.C. 1976

Lebowitz, M.:" An experiment in intelligent information systems: RESEARCHER" Davies :745:

Lee Pao, M.; Worthen, D.:" Retrieval effectiveness by semantic and citation searching" JASIS 40(4):226-235, 1989

Levinson, S. :" Pragmatics." Cambridge university press 1983, 1984

Liddy, E.:" Structure of information in full text abstracts." RIAO88

Liddy, E.:" Sublanguage grammar in natural language processing" RIAO91, 1991, p.707-717

Loucopoulos, P.; Layzell, P.J.:" Improving information system development and evolution using a rule-based paradigm." Software engineering journal 1989, p.259-276

Luhn, H.P.:" The automatic creation of literature abstracts." I.B.M. Journal of research and Development 2(2), 159-165. 1958

MacLeod, I. A.:" Storage and retrieval of structured documents" Information processing and management, vol 26.2,pp 197-208, 1990

MacLeod, I.A.; Reuber, A.R. :" The array model: a conceptual modeling approach to document retrieval" JASIS 38(2=3):162-170, 1987

Mann, W.; S. Thompson :" Rhetorical structure theory: a theory of text organisation.reprinted from 'the structure of discourse'. " Information Sciences institute. USC 1987

Maron, M.E.:" Automatic indexing: an experimental inquiry" in: Journal of the association for computing machinery, 1961, p.404-417

Martin, T. H.:" A feature analysis of interactive retrieval systems" Stanford university California 1974

Mc Cune, B.P.; Tong, R.; Dean, J.;e.a.:" RUBRIC, a system for rule-based information retrieval" IEEE transactions on software engineering. 1985, p.939-944

Olle, W.T.:" The Codasyl Approach to Data Base Management" John Wiley & Sons Chichester 1980.

Oswald, V.A.:" Automatic indexing and abstracting of the contents of documents." Los Angeles, Planning Research Corporation, 1959

Paice, C.D.:" Constructing literature abstracts by computer: techniques and prospects." Information processing and management 26, 1990.

Paice, C.:"A thesaural model of information retrieval".Information processing and management, vol27, no.5. p.433-447, 1991

Paijmans, J.J.:" Free text data bases" Proceedings RIAO88, Mass. 1988.

Rau, L.F.:" Conceptual information extraction and retrieval" RIAO88 Cambridge mass. M.I.T. 1988

Rau, L.F.; Jacobs, P.S. :" Natural lanuage techniques for intelligent information retrieval" SIGIR(1988):642:

Rau,L.F.; P.S. Jacobs and U. Zernik. :" Information Extraction and Text Summarization Using Linguistics Knowledge Acquisition " Information Processing and Management (Oxford) 25 (1989) nr. 4 p.419-428 (20 refs.)

Reynolds, D.:" Library automation: issues and applications." Bowker, New York, 1985

Rijsbergen, C. J. van:" A non-classical logic for information retrieval" The computer journal 29, pp.481-485. 1986

Rijsbergen, C.J. van:" Information Retrieval" Butterworths, sec. edition 1979

Rouse, W.B.; Rouse,S.H.:"Human information seeking and design of information systems", in:Information processing and management, vol.20; p.129-138, 1984

Ruge, G.; Schwarz, C.; Warner, A.:"Effectiveness and efficiency in natural language processingfor large amounts of text." in: Journal for the american society for information science 42 (6): p.450-456, 1991

SARACEVIC, T.; P. Kantor; A.Y. Chamis :" A Study of Information Seeking and Retrieving; Part 1, 2, and 3" Journal of the ASIS (New York, NY) 39 (1988.05) nr.3 p.161-216 (with refs.)

Sager, N.:" Natural language information processing; a computational grammar of english and its applications." Addison Wesley 1981

Salton, G. :" Another Look at Automatic Text-Retrieval Systems" Communications of the ACM (New York, NY) 29 (1986.07) nr.7 p.648-656 (21 refs.)

Salton, G. ;M.J. McGill :" Introduction to Modern Information Retrieval" New York [etc.] : McGraw-Hill, 1983. - 448 pp.

Sandore, B.:" Online searching: what measure satisfaction" Aslib 1990 (?)

Schank, R.;Abelson, R.:" Scripts, plans, goals and understanding" Hillsdale, New York 1977

Small, G.W.; Weldon, L.J. :" Human factor studies of database query languages." Human Factors, 25, 253-263, 1983

Smith, P.;M. Barnes:" Files and databases." Addison Wesley 1987

Sparck Jones, K. :" Information Retrieval experiments" Butterworths, London 1981

Sparck-Jones, K.; Jackson, D.M.:" Current approaches to classification and clump-finding at the Cambridge Language Research Unit." Computer Journal 10, 1967, p.29-37

Sperberg-McQueen, C.M.; Burnard, L. (ed):" Guidelines for the encoding and interchange of machine readable texts" Chicago & Oxford 1990

Tague, J.M. :" The pragmatics of Information Retrieval experiments" in: Sparck-Jones(1981) :567:

Teskey, F.N.:" User models and world models for data, information and knowledge" Information processing and management, Vol. 25, no 1, 1989,pp. 7-14

Verharen, E. :" Hypercard en databases: een vooronderzoek naar de mogelijkheden van Hypercard." ITK memo, 1989

Waal, ? van de:" Some principles of a general iconographical classification." In: 'Actes du XVIIme congres internationale d'histoire d'art. Den haag 1955, pp.601-606

Waltz, D.:" Applications of the Connectionist machine." Thinking Machines Corporation 1986, Technical report 86-12

Wendlandt, E.; Driscoll, J.R.:"Enhancing text retrieval automatically", in: Karigiannis, D. (ed):"Database and expert systems applications, proceedings of the conference in Berlin, 1991" Springer Verlag, Wien, 1991, p. 118-123.

Winston, P.H.:" Artificial Intelligence" 2-th edit. Addison Wesley 1984

Index

ITK: P.O. BOX 90153 5000 LE TILBURG THE NETHERLANDS