[Commission of the European Communities, Joint Research Centre, Annual Report 1972]

AUTOMATIC THESAURUS CONSTRUCTION FOR INFORMATION RETRIEVAL

S. Perschke, H. Fangmeyer

Introduction

The Thesaurus, in information retrieval, represents one of the cardinal points of a system, since neither indexing quality nor retrieval strategy sophistication are able to remedy deficiencies of the Thesaurus used. In principle, the Thesaurus is a collection of concepts which are more or less important for the subject field of the document collection. Usually the mere list of the terms representing the concepts is enriched by two factors:

— lists of synonyms of which one only is considered to be valid (this list also revolves the problem of ambiguities of terms through replacement by univocal terms);

—classification schemes which may be both hierarchic and associative and may present different degrees of sophistication. Classification of terms, in information retrieval has the primary task of permitting search on a different level of specificity from that of the documents.

In the present large-scale information retrieval systems, the thesauri are compiled manually, frequently on the basis of pre-existing classification schemes, authority lists, subject headings etc. and are intended for manual indexing. As a consequence, there is a trend to limit the Thesaurus' size somewhat so as to facilitate the work of the indexers and of the documentalists who formulate the users' queries. The limitation of the number of terms admitted to the Thesaurus, in even, forces indexing to be on a rather general level and therefore the search efficiency is frequently unsatisfactory if the queries are very specific.

Furthermore, the Thesauri rely rather heavily on the indexer's associative power. The experience made at CETIS in automatic indexing with the EURATOM Thesaurus [1, 2, 3] showed an evident incoherence between the thesaurus conceived for normal indexing and the automatic indexing methods — in particular, the different degree of specificity and an unsatisfactory classification scheme. It was therefore felt to be worthwhile to try to give up dependence on manually compiled Thesauri in automatic documentation systems and to develop methods and the respective software packages for automatic Thesaurus construction. This decision appears to be even more important, now that documentation is rapidly expanding into new, and inter-disciplinary sectors which do not possess a very well-defined specific vocabulary, and manual Thesaurus construction is known to be an extremely time-consuming and costly work with a relatively small degree of efficiency.

The idea of automatic vocabulary definition for documentation purposes was probably one of the first solutions in automatic document processing with KWIC and KWOC indices and permuted KWOC indices [4,5]. These solutions are still largely used especially for SDI, news, bulletins etc.

For large-scale documentation systems with retrospective search capabilities, KWIC and KWOC turned out to be less satisfactory because amongst other draw-backs the size of lists increases very fast and the lack of any vocabulary control causes a high noise ratio[6]. As a special case of oriented KWIC/KWOC one can consider the so-called free-text search systems which, in principle, do not perform any analysis of the source documents, but try to match the words contained in the queries with those on the documents. There is at present a dispute going on between traditional systems with hand-made thesauri and manual indexing and free-text search systems.

The opposite solution, which is also being studied by CETIS, is the development of thesauri with automatic methods on the basis of the documents which constitute the information base. The first attempts were made in particular by G. Salton[7] who demonstrated through a series of experiments that automatic indexing with an automatically-built thesaurus — even with extremely rudimentary and brute-force methods — can be more efficient than the traditional methods (man-made thesauri — free-text search), if the search strategy is adequate (automatic query formulation, probabilistic search, relevance feed-back). The methods and the speculative back-ground of this position were examined critically at CETIS and found acceptable as a good point of departure. Two aspects of the CETIS approach constitute a substantial difference as compared with Salton's work:

— the use of more sophisticated linguistic methods

— the practicability of the software for very large data bases.

The package for automatic thesaurus construction developed at CETIS is part of the SLC-II system and is inserted into the framework of the FAIRS project[8]. The work involved is on the one hand methodological (linguistic and statistical) and on the other hand a software development which completes the SLC-II system.

The Automatic Thesaurus Construction Package

The CETIS project for automatic thesaurus construction does not start (as Salton did) from the statistical evaluation of graphemes (i.e. character strings representing the texts), but from lexemes (i.e. lexical units which, ideally should represent "concepts"). Therefore, before one can start with the procedure for thesaurus construction, one needs two kinds of linguistic preparatory work:

—a grammar of the character string representing the source text which permits recognition and normalisation of substrings which are elements of the source language (word items) and separation of them from other substrings which may be control characters, command codes for display function, type selection etc. (non-word items).

—a source language dictionary which is articulated into two parts:

—on one hand the morphological search dictionary with the associated tables of paradigms, and on the other hand the "lemma" dictionary which must contain, as a minimal information, a character string which represents the lexeme, especially for intellectual control, but may contain any other type of data considered to be useful.

The details on coding and monitoring dictionaries and grammar can be found in the description of the SLC-II system at present in preparation at CETIS. Here, only an extremely sketchy outline of the potentiality of the morphological analysis algorithm of the SLC-II system is given, as the decisions taken may considerably influence the picture of the resulting thesaurus.

The dictionary search and morphological analysis module of the SLC-II system provides for the following functions:

—segmentation of compound words

—detachment of prefixes — which can be considered as a particular case of word segmentation. If this option is chosen, the "lemma dictionary" must contain an indication of whether the detached string represents a prefix, and its function, or whether it is an autonomous lexeme.

The decision on how far these options (beyond inflection) are activated depends on the objectives of the system - SLC-II, in principle, can reduce the chains given below to one lexeme only (ION) or consider, as an opposite extreme, each one of the items an autonomous lexeme.

ION-IONIC-IONIZE-IONIZATION-DEIONIZE-DEIONIZATION

Which extreme or intermediary solution is chosen depends upon the subject field and on the use of the system.

Another aspect of this model is the automatic detection of homographs, largely used on the present package.

The process of automatic thesaurus construction is subdivided into a series of cycles which are described in detail here below

1) Detection of words Not In Dictionary (NID), concordance of NID, updating of the dictionary, correction of errors in the source text.
2) Detection of homographs, concordance of homographs, intellectual resolution of homographs.
3) Frequency count of the lexemes and determination of the statistically significant words ("concepts").
4) Statistical definition of compound expressions.
5) Automatic classification of terms.

Handling of Words Not In Dictionary (NID)

The methods used for automatic thesaurus construction are principally statistical, and it was felt to be important to avoid errors due to the incompleteness of the source language dictionary and to errors in the source text.

The source text is processed by the SLC-II system batch-wise (app. 700,000 current words per cycle if the main storage size available is 300 K). In this cycle, the first two modules — source text analysis and dictionary search are executed. The frequency of each distinct grapheme is determined and recorded for subsequent use. Optionally, for text and control purposes, the system can produce a list of the graphemes with their frequency sorted either by frequency or alphabetically.

After dictionary search, the results are scanned for words not in dictionary and the context of each NID is either printed or displayed on a CRT.

This concordance must be scanned intellectually, and the analyst must decide whether a given NID is a new word or it is due to a spelling error in the source text.

In the former case, the source language dictionaries must be updated. In the latter case, a series of commands has been provided for, which permit the errors to be corrected (replace, insert, delete graphemes).

This facility, which is an integrated part of the SLC-II system can be used in all applications (indexing, translation, query formulation etc.) to correct errors. The approach is valid as long as a spelling error does

not produce a valid word. In this case the error would not be detected. The analyst's decisions are collected into a file of detector commands with the correct spellings, looked-up again in the dictionary and merged with the pre-existent results. The correction cycle can be iterated until a satisfactory degree of correctness is reached.

Resolution of Homographs

Homographs are automatically detected during dictionary search. Automatic resolution of homographs, in the light of the progress of computational linguistics, is possible to a large extent through syntactic and semantic analysis. However, satisfactory and efficient algorithms are not always available or easy to integrate into the system, and it is also extremely dubious whether the limited scope in this context would justify the expense of the development and the operation of such algorithms.

The solution chosen here, is an intellectual resolution of homographs. After dictionary search (and eventual NID resolution) the results are scanned for homographs. For each homograph the "lemma" entry is loaded (the lemma should be sufficiently explicit to permit distinction of meanings), and for each occurrence of the ambiguous word the context is printed or displayed on CRT.

The analyst scans all occurrences and decides with which of the "lemmas" it should be associated.

The decisions are collected into a command file and the system scans the source text image against this file and applies the decisions. As a result, the homographs are handled as if they had been distinct graphemes, and the frequencies are adjusted correspondingly. After all these operations have been performed, the source text preparation can be considered to be concluded. All words occurring in the text are known univocally, and an evaluation of their usage frequency can begin.

Cumulative Frequency Dictionary Construction

The purpose of this operation is to distinguish the significant words from the non-significant ones in the collection of source documents. The frequency of usage of words has been examined in various contexts (most frequently in terms of graphemes and not of lexemes) and it is known that in each language - if one takes a sufficiently large data collection as basis — the frequency distribution is very much the same: there are a few words with a very high frequency, a certain number of words with a medium frequency and very many words with a low frequency.

The concept of significance, from the documentary point of view can be interpreted in two ways

— words are significant if they define important concepts in the subject field - here, the traditional linguistic interpretation is applied which subdivides the vocabulary of a language into "non-significant" words (mots vides) which are also called "function words" such as articles, prepositions, conjunctions etc., and "significant" words which are the rest of the vocabulary.

— words are significant, if they are useful for the distinction of one document from another. This is a documentary interpretation and makes the significance a function of the frequency of usage. At the limit, if some word appears in all documents of the collection, it is of no further use for retrieval purposes and hence non-significant.

The procedure developed takes into consideration both definitions. One can introduce a "stop-word" list which includes all function words and words which are considered to be non-significant even if their frequency is relatively low. Furthermore, after the frequency count executed on a representative sample of the data base, one can introduce a cut-off function which takes into account the dimension of the sample and the final dimension of the data base projected (relative frequency). In principle, all words below this threshold are significant. Special attention is to be given to the low-frequency words, since statistically, they must be interpreted as casual events. This casuality can be interpreted due to the usage of the word, in this case it should be replaced by some more common synonym. On the other hand the concept defined by the term may be unusual, and in this case the term is valid. Furthermore, these words are not very convenient for automatic classification procedures. After the frequency count, and on eventual merging with a master dictionary, a list by decreasing frequency can be issued, which is to be scanned for possible further intellectual interventions.

Automatic Recognition of Compound Expressions

The reason for this phase is the generally accepted fact that in natural language many concepts, are defined not by a single word, but by some expression (two or more words with some syntactic link).

Frequently, the single words used in an expression, are of little significance, and only their co-occurrence conducts to some precise concepts (as, for instance, the single terms "house" and "common" combined into "House of Commons". The automatic definition of expressions is based on the statistical evaluation of the co-occurrence of words in logical text units (e.g. sentences, clauses). All linguistically significant words are examined, and the low-frequency words are excluded since they do not provide any statistically valid evaluation.

In the algorithms developed, the sequence of the words in the texts is ignored (a.b = b.a), which permits the evaluation of expressions of any reasonable length. The algorithms are designed so as to consider each

batch of text (app. 100,000 current words) as independent events and the results can be a-posteriori.

The procedure of detecting expressions is combinatorial, but a few precautions have been taken to avoid processing excessively large matrices:

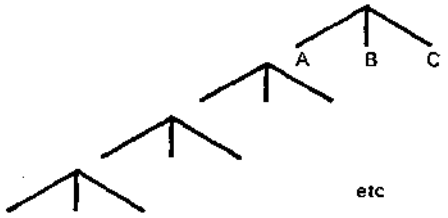
— low-frequency words are excluded
— words are combined only with higher-frequency words.

By this means, words which occur frequently can only be combined with a few other words, while words which can be combined with many others, have only a few occurrences. This way, the size of the matrices can be kept quite constant. Each matrix itself is a word-sentence incidence matrix for each root word, organised for the rows, by the decreasing frequency of the words, and for the columns, by their occurrence in the text. The decision on the acceptance of an expression depends upon

—the dimension of the sample
—the length of the expression
— the frequency of co-occurrence
— the frequency of the single elements.

With these data, a decision table is built.

This table is also used to decide whether the word can be used in an expression at all. It is evident that the minimum co-occurrence decreases with the increase of the expression length. The operation of testing the consistency of the expression is a combinatorial recursive push-down operation (interpreting all expressions as a tree):



At each step, the row representing a previous intermediate result and one row of the matrix are intersected, and the number of incidences counted. If the number of incidences is over the threshold for a given expression length, the expression is accepted. Then one tests whether the number of incidences is sufficiently high to form a larger expression. If not, the path on the tree is abandoned, and the next one is examined through a pop-up operation.

The expressions thus formed, are linked with the "lemma" entries and the sentences on which they occur and displayed for intellectual control. This intellectual control has two functions:

— to decide whether the expression is to be accepted or not
- to define a "lemma" which is mnemonically adequate to describe the concept.

After selection of the expressions, the thesaurus is expanded by the addition of the new expressions.

Automatic Classification of Descriptors

General Considerations

The thesaurus vocabulary is constituted on the one hand by the significant words encountered in the source documents, and on the other hand by the compound expressions. In the procedures described here-above, homographs are resolved intellectually. In principle, it is sufficient for automatic indexing (which takes into account all criteria with which the thesaurus has been built), which must be exactly at the level of specificity of the document.

The necessity of "classifying" the thesaurus terms derives from the fact that indexing is not an end in itself, but is only meaningful as a tool for retrieval. As, frequently, retrieval queries do not ask only for situations precisely as described in the documents, but may also ask for situations of which the ones described by the documents are special cases, or are merely similar, the level of specificity in indexing may be the same as for the documents, but, during search, one must be able to scan the documents at a different level of specificity. For this purpose, a system of paradigmatic relations between descriptions must be created. Hand-made thesauri rely upon an empirically constituted classification scheme (such as that of UDC), and normally distinguish two kinds of relations:

—hierarchic relations (general-specific terms)
— associative relations (related terms).

These relations, in general, are not qualified further.

The purpose of the package for automatic classification is to resolve the following problems:

— difference of the level of specificity between documents and queries (hierarchic relations).
—search for more or less similar situations (associative relations)
—resolution of synonyms (in principle, synonyms are a special case of associative relations with a very high degree of association)
— automatic resolution of homographs.

Definition of the Properties of the Terms

In any sort of classification, it is necessary to identify some properties of the objects, which are then evaluated statistically. In the specific case of classifying terms, it would be the best solution to give an exact semantic definition of the meaning of each term and to consider these definitions to be the properties of the terms. Since, unfortunately such definitions (and even methods to produce them) are not yet available, one is forced to look for some surrogate which, theoretically, is less satisfactory, but appears to be practicable.

The most evident property of a term is its use in some document, together with other terms. This property is the point of departure in all known attempts to classify terms automatically (term-document-matrix). In each document, a given term appears together with other terms which can be conventionally called the context of a term. If one cumulates all single contexts of a term, one obtains an "ideal" context representative for a given sample (term-term-matrix).

The terms, their frequency or weight in the context are considered to be the properties according to which the classification is performed.

All classification attempts are based on the two following hypotheses which appear to be quite plausible, but have no exact proof:

— the degree of the similarity of two terms is the same as the degree of similarity of their contexts
— the degree of specificity of a term is inversely proportional to the frequency of its use.

These hypotheses have not been critically evaluated at CETIS: they were accepted for their plausibility, as they are the only ones available, which permit practical work. Only when the software package which is being implemented for this purpose has become operational, and has been applied for' very large data collections, shall we be in the position of issuing some assessment on their practical usefulness for information retrieval.

Construction of the Context Vectors (CV)

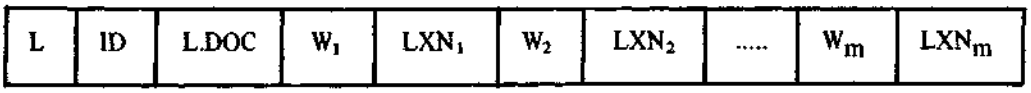
The terms being classified are no longer the words of the natural language of the source documents, but have become the elements of the thesaurus (single terms and compound expressions). This presupposes the availability of indexed documents. The SLC-II system works with an inverted-direct file organization for search and an alphanumeric document file for visualization.

The inverted file contains for each descriptor the pointers to the entries in the direct file addresses and its weight in each of the documents

L : length of the entry
LXN : identification code of the term
W : weight of the term in the document
TRK : Track number
P : Position in the track

Ideally each entry of the inverted file represents a row of the term-document-matrix.

The direct file contains at the position indicated the inverted file records as follows:

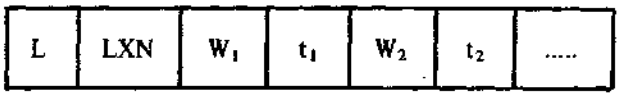


L —length
ID — internal document identification code
L.DOC — pointer to the alphanumeric document entry (track-position)
W — LXN — same as inverted file.

The CV construction algorithm is to be applied each time a batch of new documents has been indexed, since the inverted and direct master files, with large collections, may grow very large and require excessive computer time. Procedures are provided for, to cumulate CV files over the life time of an IR system.

In the construction of the CV, algorithms developed in the framework of the SLC system are largely used: the entries of the inverted file are scanned one by one, the corresponding direct file entries are loaded, and the LXN are cumulated either by just counting the frequency or by some formula which normalizes the cumulation of the weighting factors. The terms in the CV are ordered by decreasing weight in order to keep the vectors of a manageable length, one can cut-off the low-order section of the vectors

according to some threshold value which depends mainly on the sample dimension and the cumulative weight of the term examined. The structure of a context vector is very similar to that of the inverted file:



L – vector length
LXN – term identification code
W – weight of co-occurring term
t – LXN of co-occurring term

Clustering Algorithm

Methods of automatic classification normally imply combinatorial algorithms which inevitably lead to a number of operations in the order of magnitude of n^2 (where n is the number of elements to be classified). This magnitude becomes very soon impracticable with the growth of n (classification of 10,000 terms would imply the order of 100 million operations). Therefore, it becomes necessary to look out for some classification algorithm which implies a possibly linear incrementation of operations— even at the cost of some loss in precision. An automatic clustering algorithm which works within the limits of a linear incrementation function of the order n was developed in the environment of the SMART project known as a single pass algorithm, and was implemented in this phase of the package [9,10,11)].

The algorithm developed at CETIS is a slight modification of the single-pass algorithm and operates recursively on a top-to-bottom basis.

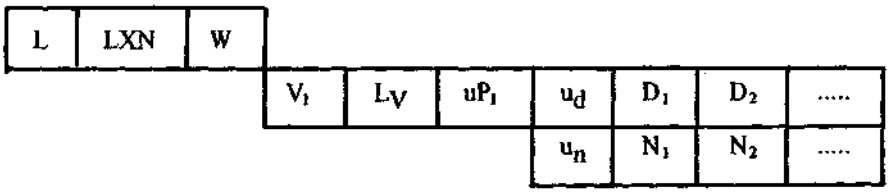
The algorithm defines the centroids of each cluster in the following way (see Figs. 1 and 2):

1. One takes the first CV and considers it as a potential Centroid Context Vector (CCV_1)
2. One takes the subsequent CV and computes the similarity function to the CCV_1
3. If the similarity is over a cut-off value, the CV is considered to be part of the class represented by the CCV_1 and a new CCV is computed which is the "center of gravity" between the two
4. If the cut-off limit is not reached, the current CV is declared a new CCV, and for the subsequent CV, all existent CCV are scanned. Optionally, one can admit a certain degree of overlap, if an item can be associated to more than one centroid with respect to the cut-off value.
5. At the end, since the CCVs are modified during processing i.e. the "centers of gravity" "move", the following operations are performed
   5.1 .The mutual similarity between all CCVs is computed and CCVs which are too close to each other are merged.
   5.2.The CCVs are re-processed, and the similarity to the CCVs established is re-computed.

The process is recursive and assigned to those class(es) to which it is most similar top-to-bottom, in the sense that each class can be split up into subclasses and terminates when either the number of elements - which enter into a class becomes too small, or an existent class cannot be broken up because all CVs are too close to each other.

The main problem in applying this method is the fact that the results obtained depend on the sequence in which the CVs are processed. One should be more or less certain that this sequence, with respect to the contents of each CV, is random. The primary sequence of the CVs is the same as in the thesaurus and in the inverted file, i.e. by increasing LXN. As the LXN is normally chosen independently of the meaning of the words (e.g. according to the alphabetic sequence of the graphemes) it is very likely that the LXN result will be random with respect to the properties, i.e. the contents of the CV.

The centroids of the clusters at each level become terms of the thesaurus just like all other terms, but they are exclusively used in retrieval. Such a term is structured as follows:



L —entry length
LXN —identification code for centroid
W — the centroid weight
V_1 — identification code of variable field
L_v —length of variable field
uP —Link to upper-level centroid
u_d —number of links to lower-level centroids or terms
D_1 — etc. - LXN of lower-level centroids or terms
u_n — number of centroids at the same level + similarity ordered by similarity
N_1 — etc. — LXN of some-level centroids or terms

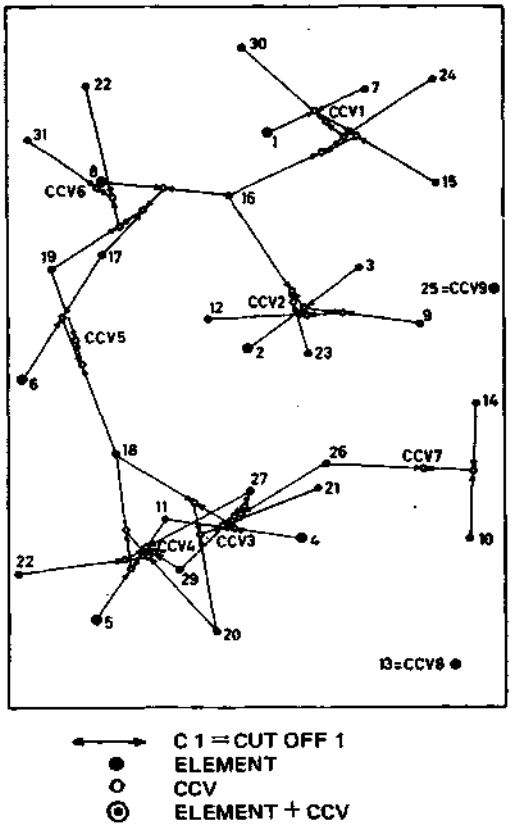


Fig. 1 : Steps 1 - 4

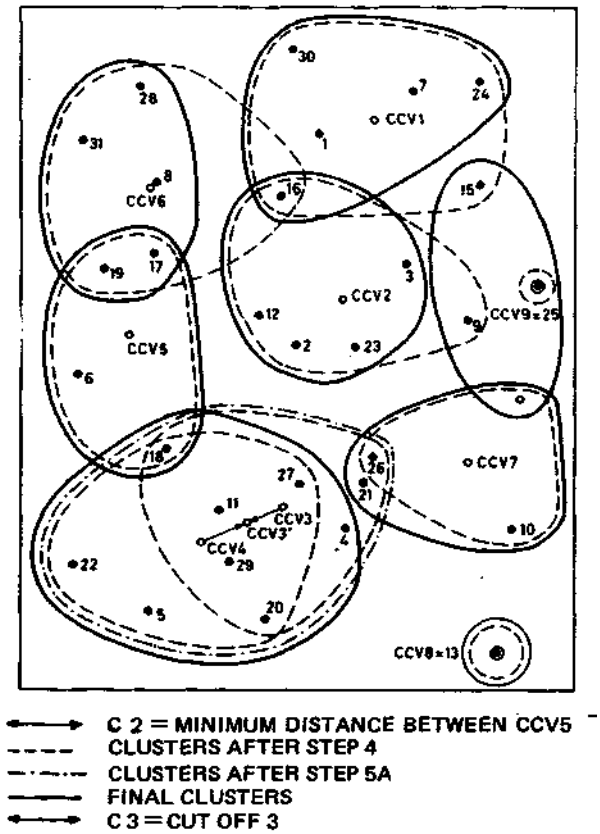


Fig. 2 : Steps 5.1 + 5.2

Handling of High-frequency Significant Terms and Homographs

Retrieval strategies based on an inverted-file structure can be seriously disturbed, if some terms are used too frequently, as the inverted file vector becomes too long and one runs into main-storage and time difficulties.

According to the working hypothesis described above, high-frequency terms are general terms. In order to restrict the length of the vectors in the inverted file, it was decided to handle these general terms as if they were ambiguous, i.e. to specify the meaning through context analysis.

One uses the same algorithms as for term classification and classifies the documents (direct file entries) of a given general term.

The centroid of each cluster is declared an artificial term, the inverted file entry relative to the high-frequency term is deleted and replaced by the (shorter) ones relative to each centroid. After this operation, in indexing, the general term is no longer used, but the context of the document being examined and the centroid representations are matched, and the closest centroid is assigned as descriptor.

Term clustering can also be used for the automatic resolution of homographs, once the size of the document collection has sufficiently grown for statistical decisions. For this purpose, the context vectors of the terms must be available during indexing. The ambiguous term can be interpreted as a general term which must be replaced by one or more specific terms. The criteria for decision is given through the similarity function between the context vectors and the document.

Conclusions

Statistical evaluation of co-occurrence and association factors has been one of the principal research objects in automatic documentation at CETIS before 1968[12], but practical applications for this work could not be gained due to the lack of efficient software and, consequently, of statistically significant document collections and vocabularies.

The implementation of SLC-II made it possible to design a package which will permit application of these methods in a real life environment with large data bases (10^5 - 10^6 documents). The analysis and design of this package started in early 1972, was concluded in the summer, and implementation started in September 1972, partly under contract with the software company Italsiel, Rome. The implementation effort is estimated to be two man-years, and it is expected that the package win become operational in autumn 1973.

The availability of an efficient software package for automatic thesaurus construction represents a further progress in the realization of FAIRS, i.e. a fully automatic information retrieval system. It is evident that a good deal of documentary and statistical research is necessary before an optimal application of this package can be obtained, but past experience has shown that this kind of empirical and heuristic research is not meaningful if it cannot be based on statistically significant samples.

References

1) H. Fangmeyer, G. Lustig: The EURATOM automatic indexing project. Paper presented at the IFIP Congress, Edinburgh (1968).
2) H. Fangmeyer, G. Lustig: Experiments with the CETIS automatic indexing system in handling of Nuclear Information. International Atomic Energy Agency, Vienna (1970).
3) S. Perschke: A generalized information retrieval system and the associated software (Paper presented at the Seminar on "Sistemi di Reperimento e Selezione Automatica dell'Informazione") April 1971 (in press)
4) H.P. Luhn: Keyword in context index for technical literature (KWIC - Index) presented at: American Chemical Society Division of Chemical Literature at Atlantic City, N.J. 14, Sept. 1959
5) M. Fischer: The KWIC index concept: a retrospective view. *Am. Doc.* 17,57-70 (1966)
6) M.E. Stevens: Automatic indexing: a state of-the-Art. Report National Bureau of Standards Monograph 91
7) G. Salton: Automatic information organization and retrieval. McGraw-Hill Book Company, 1968
8) S. Perschke, H. Fangmeyer: Methodology and software development for application in information science.Commission of the European Communities, Joint Research Center, Ispra Establishment, Italy. Annual Report 1971, EUR 4842.e
9) S. Rieber, V.P. Marathe: The single pass clustering method. Scientific Report No. IRS-16. Department of Computer Science, Cornell University, Sept 1969
10) D.R. Hill: A vector clustering technique in: Mechanized information storage, retrieval and dissemination proceedings of the F.I.D/I.F.I.P. Joint Conference, Rome, June 14-17, 1967. North Holland Publishing Company-Amsterdam 1968.
11) K. Hamilton, J. Leslie: Thesaurus Class Modification based on user feedback. Scientific Report No. ISR-21. Department of Computer Science, Cornell University, Dec. 1972.
12) G. Lustig: A new class of association factors. In: Mechanized Information Storage and Retrieval and Dissemination. Proceedings of the F.I.D./I.F.I.P. Joint Conference, Rome June 14-17 1967. North Holland Publishing Company-Amsterdam 1968.