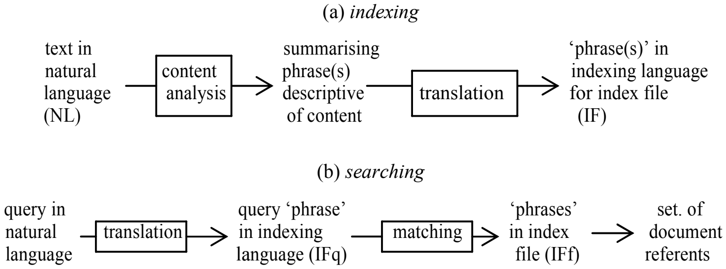
[From: *Linguistics* 61 (September 1970), pp. 29-64]

LINGUISTIC PROCESSES IN THE INDEXING AND RETRIEVAL OF DOCUMENTS

W. J. HUTCHINS

1 INTRODUCTION

For the storage and retrieval of documents in a library or other depository many kinds of clerical and intellectual operations are performed by librarians and by library users. Some of these are purely routine and are done almost 'mechanically' but others are highly complex and require intelligent application. The complex processes concerned with the indexing of documents for their subject content and the searching of index files from the subject approach may be represented as follows:



(a) *indexing*

text in natural language (NL)
content analysis
summarising phrase(s) descriptive of content
translation
'phrase(s)' in indexing language for index file (IF)

(b) *searching*

query in natural language
translation
query 'phrase' in indexing language (IFq)
matching
'phrases' in index file (IFf)
set. of document referents

Fig. 1

The indexing process is divided into two stages. Human indexers generally perform both stages simultaneously, but for the purposes of formalization it is useful to consider them separately.

(i) The text in natural language (NL) — which can range in size and complexity from a single paragraph to a multi-volumed treatise — is analysed and described in a set of phrases summarizing the content of the document.

(ii) These phrases are translated into 'phrases' (or formulae) of the indexing language (IL), which, depending on the kind of IL used, may be any combination of alphabetic, numeric or any other graphic symbols.

An ordered collection of such index phrases (IPs) for documents constitutes an index file for the document store (library).

The searching process is also represented in two stages.

(iii) The query expressed in NL by the inquirer (or index-user) is translated into a phrase of the IL with the same meaning. This may be done by the index-user himself or by a librarian.

(iv) The IL phrase is compared with entries in the index file, and if a match is made, the documents indicated are (hen consulted as of potential interest to the inquirer.

Except in the case of matching, all these processes involve the transformation or translation of one sequence of symbols in a code or language into a sequence of symbols in another code or language, i.e. they are linguistic activities. The matching process has been tackled relatively successfully in the automation of many systems of information retrieval (IK), but the linguistic processes often remain the province of human operators — indexers, cataloguers, readers' advisors, information scientists, etc. They are acknowledged to be not only the most complex but also the most expensive of all IR processes and consequently their mechanisation is seen to be most desirable.

An alternative method for retrieving information from a document store from the subject approach is the process of 'scanning' or (possibly) 'browsing'. Faced with a set of (unanalysed) documents the user scans the text of each for phrases or groups of words which in his opinion characterise his subject interest. These phrases are all equally valid expressions of his query: they may legitimately be held to be reformulations or translations of a statement of subject interest made by the user before beginning his search.

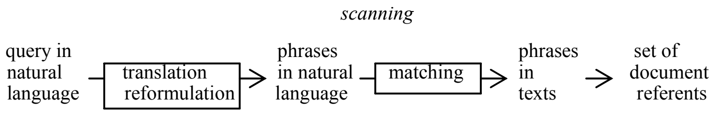

scanning
query in natural language
translation reformulation
phrases in natural language
matching
phrases in texts
set of document referents

Fig. 2

Formally the process may be divided into two stages:

(i) The query phrase is reformulated (in the same NL) or translated (into another NL) in alternative expressions which are all semantically equivalent.

(ii) These expressions are matched with phrases occurring in the unanalysed texts of documents.

Whereas the matching stage is relatively straightforward, as it is in searching index files *(*fig. 1 (b)iv), the first stage – translation/reformuiation – is a linguistic activity as complex as those in indexing outlined above. Apart from the similarities in the processes, our reason for treating this IR process along with indexing is that the appearance of the computer now makes its mechanisation feasible (Hutchins, 1967, and section 13 below).

In this paper we concentrate on the linguistic aspects, developing a formalization suitable as a framework for the design of automated systems. The discussion is restricted entirely to theoretical aspects and excludes all questions of economic feasibility. After a brief outline of linguistic structure (of both NL and IL) we describe a system of translation via a sememic code as an interlingua and then the use of the same code in the processes of content analysis, searching and 'scanning'.

2. THE STRUCTURE OF LANGUAGES

2. 1. *Physical form.* -- The external form of a language is the sequence of sounds or symbols by which communication is made from person to person. The graphic representations of NLs have many forms. They range from Egyptian hieroglyphs and Chinese ideograms through the syllabic alphabet of Arabic to the semi-phonetic alphabets of Western languages. With a relatively small number of different symbols (26 in the English alphabet) a very large variety of sequences can be represented. Thus, for example, with the letters *a, e, l* and *t* can be represented in English *tale, late, teal, ale, ate, tea, eat,* etc.

2.2 Similarly the graphic representation of an IL is also achieved with a relatively restricted number of symbols. Two main types of IL can be distinguished by the symbols they employ. There is the type found in alphabetical subject indexes which uses symbol-sequences (words) taken from NL, and there are the numerous examples of ILs which use symbol sequences not found in NL; the Dewey Decimal Classification uses the numeric symbols 0 to 9 and the decimal point, the Library of Congress Classification uses a combination of alphabetic and numeric symbols, and many faceted classification schemes use a combination of upper and lower case alphabetic symbols.

2.3 *Semantics.* — Of all the possible sequences only those unique symbol sequences with distinguishable meanings constitute the lexemes of a language (its vocabulary). Not all sequences in a NL or in an IL are meaningful, i.e. they are not all lexemes, in English the sequence *ltae* has no semantic value. Similarly, in Dewey the sequence *1.020* is not a recognised lexeme. It is only by convention between members of a language community that particular sequences have meaning and the status of lexemes and not others.

The physical form of any lexeme[1] has no direct relationship with its meaning. In English *oculist* and *eye-doctor* are considered synonymous, but this fact is not ascertainable from the graphic representations of the two lexemes. Clearly there are non-physical associations between lexemes, semantic values assigned to them by which they may be related and distinguished. English speakers recognise the presence of an element of 'maleness' in the lexemes *man, boy, son, ram, bull, heifer, king,* etc. and an element of 'humanness' in *man, boy, son, king* which is not present in *ram, bull* and *heifer.* The analysis of lexemes in terms of such elements (semons) can provide definitions of the meanings of lexemes (sememes): thus the sememe for *boy* might be the set of semons '(being), (human), (male), (young)' and the sememe for *girl* the set '(being), (human), (female), (young)'. We shall have more to say about semantic structure in sections 6 and 7.

2.4 Whereas English and other NLs do not reflect semantic relationships in the graphic representations of their lexemes, this is not true of many ILs. An IL can be so structured that lexemes having common semons have also common symbols. The creation of a parallelism between external form and internal semantic structure has its justification and value in the searching process (section 12.2).

To illustrate we give an excerpt from the Dewey Decimal Classification, which is not generally as well structured in this respect as more recent classifications;

820	English literature
821	English poetry
822	English drama
823	English prose
823.1	English medieval prose
830	German literature
831	German poetry
832	German drama

[1] Lexeme: in NL the form of a word with no indication of grammatical role, e.g. *houses* consists of the lexeme *house* and the 'plural' morpheme *-s*.

833	German prose
833.1	German medieval prose
....	
840	French literature
....	
843	French prose
843.1	French medieval prose

The semons assigned to the lexemes 823, 833, and 843 can be divided into two kinds: those which are also found in 820, 830, and 840 respectively and those which are not found there. These latter semons are held in common by 823, 833, and 843 and are represented in every case by the symbol ‘3’ in the third position. It may be asserted from the schema illustrated that this symbol represents a semon ‘(prose)’. Similarly of the symbol ‘1’ in the first position after the decimal point it may be asserted that it represents a semon ‘(medieval)’.

2.5 *Syntax.* — Lexemes are not used singly in language but in combinations to form phrases and sentences. The syntax of a language specifies what kind of combinations may occur and which lexemes may appear in given syntactic structures. In English the lexemes *the, boy, love* and *girl* may combine as *the boy loves the girl* and *the girl loves the boy,* but not as *boy the girl the loves, the loves girl boy the* or *girl boy the loves the.* Each lexeme has only a restricted number of grammatical roles (tagmemes) in sentences and not all sequences of tagmemes are acceptable. The underlying syntactic structure of *the boy loves the girl* can be represented as the tagmeme sequence ‘Art + Nsg + VtPr + Art + Nsg’ and that of *boy the girl the loves* as ‘Nsg + Art + Nsg + Art + VtPr’. Whereas the syntax of English admits the first as a well-formed tagmeme sequence, it does not admit the second. Syntax rules can be formulated to enable the derivation of all grammatical (i.e. well-formed) sentences and of no ungrammatical ones. With the rules of fig. 3(a) we can derive a syntactic tree, fig. 3(b), the terminal nodes of which form the tagmeme sequence underlying *the boy loves the girl.* On the other hand, the syntactic tree and the tagmeme sequence underlying *boy the girl the loves* cannot be derived by these rules. (This is, of course, a simplified example. The syntax rules of English are far more complex, and include rules of a different kind from those given in fig. 3(a) —as we shall see later in section 7.8.)

2.6 The syntax of an IL specifies how IL lexemes may combine to form

(a)

S → NP + VP
VP → V + NP
NP → Art + Nsg
V → VtPr

(b)

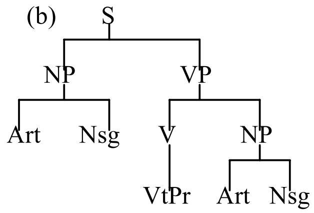


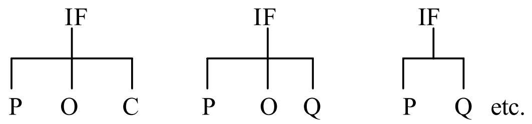
Fig. 3

index phrases (IFs) expressing the 'content' of documents. In some ILs entries in the index file may consist of single lexemes only; the language may be said to have 'no syntax'. Co-ordinate indexes are an example: any syntactic relationships between lexemes are made by index-users and never by an indexer.

In other ILs the sequence of lexemes in an IF is strictly determined. In a faceted classification, the classificationist determines that lexemes with a common element (semon) belong to a particular facet. By fixing the order in which facets may occur, he thereby determines what order a particular set of lexemes must have to be a well-formed IF. In a classification of veterinary science, Vickery has the facets P 'substance, organism', O 'part, organ, structure', C 'constituent' and Q 'property and measure'. These facets may combine only in the order POCQ, e.g. Cow (P): Intestine (O): Osmotic pressure (Q). We can formulate this as a syntax rule:

IF → P + O + C + Q (where any position may be null)

By this rule can be derived the following trees:



To the terminal nodes of such well-formed IFs may be attached only lexemes with the appropriate semons (which thus act here as tagmemes), e.g. only lexemes having a semon *p* common to the facet 'P' may occur at the node P.

In a similar way IL syntaxes may be described which specify, for example, that lexemes may appear only in the order: Thing (T), Part (Pt), Material (M), Action (A), Property (Pr). Coates (1960: 57) provides examples of permitted sequences:

Thing, Action, Part, Material, Action, Property
Thing, Material, Part, Material, Action, Property
Thing, Part, Part, Material, Action, Property
Material, Thing, Part, Material, Action, Property
Tiling, Thing, Part, Material, Action
Action, Thing, Part, Material, Action, Property
Thing, Thing. Part, Material, Action, Property
Thing, Action, Part, Material, Action

The syntax rules might be formulated as:

$$\mathrm{IF} \rightarrow \mathrm{Th} + \mathrm{Pt} + \mathrm{M} + \mathrm{Act} + \mathrm{Pr} \quad \text{(where any position may be null)}$$

$$\mathrm{Th} \rightarrow \mathrm{T} + \left\{\begin{matrix}\mathrm{Act}\\ \mathrm{M}\\ \mathrm{Pt}\end{matrix}\right\}; \textit{ or } \mathrm{T}_1 + \mathrm{T}_2; \textit{ or} \left\{\begin{matrix}\mathrm{M}\\ \mathrm{Act}\end{matrix}\right\} + \mathrm{T}; \textit{ or } \mathrm{T}$$

$$\mathrm{Act} \rightarrow \mathrm{A}_1 + \mathrm{A}_2; \textit{ or } \mathrm{A}$$

Since IL syntax is not the same as that of NL, when IL lexemes are taken from NL words, those features of NL words such as adjectival endings which indicate grammatical functions are dropped. Thus, an IL drops the suffixes *-ic, -ical, -ize* (as in *dramatic, dramatical, dramatize)* and adopts the NL form with little or no syntactic indication *(drama).*

3. MEANING

The meaning of a phrase or sentence as a whole, whether in NL or in IL, is ascertained from the meanings of its constituent lexemes and of the syntactic structure relating them. The importance of syntax for meaning is well illustrated by the familiar *man bites dog* — *dog bites man.* Both phrases have the same constituent lexemes but the syntactic relationships differ formally and semantically.

Some syntactic structures have the same semantic value as others. A well known example is the passive equivalent of an active sentence, e.g. *the girl is loved by the boy* and *the boy loves the girl.* Such syntactic equivalencies are, however, valid only if certain rules are strictly maintained, e.g. that the ‘logical subject’ remains the same. Thus, *the girl is loved by the boy* is equivalent to *the boy loves the girl* but not to *the girl loves the boy.*

The semantic abstract for any phrase or sentence — its sememic formula (SF) — must, therefore, specify the semons attached to its lexemes, the relationships between them, and the syntactic relationships between the lexemes.

Different phrases may have the same SF in the following instances:

(a) If they have the same lexemes but in different syntactic structures, when these structures are semantically equivalent (e.g. our example of active and passive forms)

(b) If they have lexemes of different graphic form but with the same semons (i.e. synonyms) and at the same time they have the same syntactic structure, e.g. *oculists sometimes marry shorthand-typists* and *eye-doctors sometimes marry stenographers*

(c) If both (a) and (b) occur simultaneously; thus, two phrases can differ in both their constituent lexemes and in their syntactic structures and yet still have the same SF (i.e. one is a paraphrase of the other).

While in NL the presence of synonyms and paraphrases is a well known feature and generally welcome for stylistic and aesthetic reasons, ILs are usually designed on the principle that no concept may be expressible in more than one way. From this it follows that in an 'efficient' IL any particular SF may have only one IF. However it may no necessarily be true that every IF expresses only one SF.

4. TRANSLATION

The first requirement for any translator is that he be able to interpret the language from which he is translating (the source language). To interpret any phrase or sentence he must discover its meaning (i.e. its SF) and this, as we have seen, requires knowledge of the meaning of its lexemes and knowledge of the syntax relating them.

The second requirement is that he be able to express the meaning (the SF) in the language into which he is translating (the target language). This also demands knowledge of the semantic structure and of the syntax of the language.

For the translation from a NL into an IL (stage ii of fig. 1) indexers have the same requirements, namely; the ability to interpret the NL (which may in practice be any existing or extinct NL and not only English) and the ability to use the IL, i.e. knowledge of the semantics and the syntax of both NL and IL.

Whereas indexers are usually provided with dictionaries and grammars of an IL (e.g. classification schedules, subject heading lists, thesauri, etc.) to help them, index-users do not have such aids for NL -> IL translation (stage iii of fig. 1). When an IL uses lexemes not found in NL (e.g. a classification scheme), the difficulty is recognised and index-users are supplied with a means of access through the other basic type of IL (alphabetical indexes, which use NL lexemes). It is generally assumed that this kind of IL is easily understood, even though the syntax is quite different and the principles of cross-reference are not always obvious to users. Human error in the use and interpretation of ILs is recognised as a major factor in the inefficiency of present IR systems. Potentially the mechanisation of NL → IL translation can bring immense improvements.

5. MECHANICAL TRANSLATION

5.1 There are basically two methods of automatic translation: binary translation and translation via an interlingua or intermediary language. Binary translation consists of devising an algorithm to transform directly phrases of the source language into phrases of the target language. Translation via an interlingua consists of devising an algorithm to transform phrases of the source language into phrases of the interlingua and an algorithm to transform phrases of the interlingua into phrases of the target language (Andreev, 1967).

If we have Q languages, then the number of algorithms required for binary translation from each language to each other (except itself) is Q (Q — 1). When translating via an interlingua only 2Q algorithms are needed: from each language to the interlingua and vice versa.

The argument for translation via an interlingua is persuasive — but what form will the interlingua take and how will the algorithms be constructed? It is clear that the interlingua must be some kind of language capable of expressing the sememic formulae (SFs) of every NL in the system in such a way that transformations can be made automatically from the graphic representation of an SF in any language to the graphic representation of the same SF in another language. It must be a sememic code or language (SL) whose phrases (SFs) are abstractions from the semantic and syntactic structures of each language. Thus, if NL_x has a phrase P_x which is equivalent to the phrase P_y of NL_y, then the interlingua (= SL) must have a phrase SF_k for which algorithms can be written for the transformations $P_x \rightarrow SF_k$ and $SF_k \rightarrow P_y$, and vice versa.

5.2 Similarly for NL → IL translation. Algorithms could be constructed for translating from each NL into each IL; in some cases the algorithm might be quite straightforward, e.g. translation from English into an IL using English words: at Sheffield Lynch (1967) has devised a fairly simple algorithm for the construction of index entries from title-like prepositional phrases. Unfortunately there are many NLs and there are many ILs more complex than this. The arguments in favour of translation via an interlingua (SL) between NLs and ILs are as powerful as they are for NL → NL translation. The special arguments in the case of NL → IL translation are:

(a) Since there is rarely any need to translate from an IL into a NL or another IL, the number of algorithms for binary translation would be $\frac{Q(Q—1)}{2}$ but for translation via an interlingua only Q. On the rare occasions when IL → IL translation is required — e.g. when an 1R system is changing its IL, or when two IR systems with different ILs are to be made compatible — then an additional IL → SL algorithm is needed.

(b) Since in any efficient IL there is for every SF only one corresponding IF (section 3), the SL → IL algorithms will be far less complex than any SL → NL algorithm.

The two types of algorithm required for NL → IL translation are then: (i) an algorithm to transform phrases in NL into SFs: an algorithm of analysis, and (ii) an algorithm to transform SFs into IFs: an algorithm of synthesis.

5.3 Before proceeding to the description of these algorithms, we must establish the general structure of the SL (using English as an example) but after having first outlined the theoretical bases for our assumption that NLs do in fact have a semantic structure.

6. SEMANTIC STRUCTURE[2]

6.1 Despite the wide variety of sentence forms, grammarians have largely discovered the rules of syntax by which they may all be derived and have systematized them in, for example, transformational grammars. And, despite the wide differences in pronunciation from one speaker to another, phonologists have systematized the phonetics of languages. Linguists believe that the semantic component of language also has a structure which can be systematized.

Hjelmslev summarized the principles of the analysis and formalization of any process (including linguistic) as: “A priori it would seem to be a generally valid thesis that for any process there is a corresponding system, by which the process can be analysed and described by means of a limited number of premisses. It must be assumed that any process

[2] The approach here owes much to the work of, among others, Antal (1963). Hjelmslev (1961), Katz (1966), Lamb (1964), and Quine (1960).

can be analysed into a limited number of elements recurring in various combinations.” The purpose of semantic analysis is then to find those basic elements — of a limited number --- which underlie the immensely rich variety of ‘meanings’: to find the constants in the semantic flux of language,

6..2 A basic fact of language is that linguistic signs (words, lexemes) are arbitrary. The relationship between a symbol-sequence (a lexeme) and the object (percept) or concept (abstraction from a class of percepts or other concepts) to which it refers (its referent) is established solely by a convention among the language community in which it is used. Lexemes may have one unique referent (singular terms) or more than one referent (general terms). The ‘extension’ of a lexeme is the set of all its potential referents. The application of a lexeme in reference is determined by its sememe — defined as a set of semons. Semons are those characteristics (attributes and properties) of a lexeme’s referents which fulfil the following conditions: (i) they must be common to all the referents, (ii) they must also be present in other sememes (e.g. our example in section 2.3 of the ‘maleness’ semon in *boy, man,* etc.), and (iii) they must be sufficient to distinguish the usage of the lexeme from the usage of all other (non-synonymous) lexemes. From the last two conditions it follows that semons are relatively few in number although capable of forming a large number of different sememes — in the same sense that the symbols of a language’s alphabet are few and yet capable of forming a large number of different and distinguishable lexemes (cf. section 2.1). It follows also that not all the common characteristics of a lexeme’s referents are semons. As we shall see, any characteristics of referents may appear as connotations of the lexeme in certain personal or social usages.

The distinction between conceptual characteristics and semons must be kept clear — the examples of semons (and of sememes) in this paper are purely illustrative and not to be taken to assert any direct parallelism between them and conceptual universals.

6.3 A percept is considered an appropriate referent for a given lexeme if it has the necessary characteristics to be the semons and to form the sememe of that lexeme. In order to be understood, therefore, a speaker (writer) must know how to select a lexeme which is appropriate to the percept he wishes to express; and in order to understand a lexeme and to ascertain its specific referent a hearer (reader) must know its sememe and what characteristics in referents the sememe demands. Communication clearly requires from both speaker and hearer prior knowledge of the sememes of the lexemes they are using and of their potential referents.

It is sometimes asserted that some lexemes are 'meaningless' out of context. But from what we have just said it is clear that context cannot give meaning to lexemes — that is done by sememes. What context does is to narrow down (for the hearer) the potential referents of a lexeme to one specific referent. For example, the lexeme *book* may denote any one of millions of possible referents; if *red* is added to it in a phrase *(red book)* the possibilities are more limited; if *big* is added there is a further limitation; and finally if *my* is added *(my big red book)* the referent of *book* is specified to one particular object. Abstract words, such as *democracy, freedom, virtue,* etc., are no different in this respect from any other general terms. It is always possible to specify exactly the referent intended, e.g. *democracy as practised in Athens in the age of Pericles, democracy as understood by Lenin,* etc. We may recall Wittgenstein's summary of the message of his *Tractatus* (1929): "Was sich überhaupt sagen lässt, lässt sich klar sagen." What does distinguish abstract general terms from concrete general terms is the difficulty speakers of the language have in enumerating all their referents and consequently in conceiving and defining their ideal exemplar (abstraction or concept). But despite these difficulties speakers do know, to a considerable degree, exactly how abstract terms are to be used, i.e. it is not their sememes which are vague but their referential extension.

With homonyms the situation is rather different. A homonym is a lexeme to which more than one sememe can be assigned, since its referents fall easily into distinct non-overlapping groups, in each of which a set of common characteristics (and of semons) can be distinguished. In an act of communication a speaker knows which referent and which sememe he intends when using a particular homonymous lexeme, but the hearer may not be able to ascertain the exact referent without 'contextual clues' i.e. the presence of other lexemes enabling him to exclude (as impossible or less probable) certain of the homonym's sememes. The uncertainty of the hearer lies not, as with abstract terms, in the vagueness of the extension but in the choice of conflicting sememes.

6.4 When an individual speaker diverges from the commonly accepted sememe of a particular lexeme, he assigns to it personal connotations. At a trivial level, for example, he might consider that a semon '(four legs)' is a component of the sememe for *chair,* whereas for the majority of the language community this semon is absent. Consequently the extension of the referents of *chair* is smaller for him than for others, and in the face of certain percepts (e.g. three-legged chairs) he will find himself in disagreement with them.

It is recognised that there are many connotations held by large subgroups of the language community in common. When a subgroup shares common interests — intellectual, professional or social — its specialist vocabulary (or specialist usage of common vocabulary) is called variously jargon, slang, journalese, etc. While personal connotations can be ignored for the purposes of semantic analysis, specialist usage cannot because it is the usage adopted by a community of speakers rather than that adopted by one individual. Therefore, just as homonyms have more than one sememe, other lexemes may also be assigned additional sememes to take account of their specialist connotations.

Personal connotations in language are reflections of the ways in which men differ in their conceptualisation of the world around them. Psychologists have long recognised that connotations are important indicators of a person's attitude to his environment. Literary critics are also well aware of the importance of an author’s word usage for the illumination of his ‘philosophy of life’.

6.5 Just as any characteristics of a lexeme’s referents may be connotations of that lexeme, any characteristics of percepts may contribute to the formation of a concept. We may broadly define a concept as an abstraction from a class of percepts (or other concepts) based on the characteristics perceived in those percepts. Since there is no limit to the characteristics which might be observed in percepts already ‘known’ and since new percepts are constantly being ‘discovered’, the formation of concepts is a continuing and recurrent process. In its wake it brings new conceptual patternings and typologies to supersede old ones which no longer adequately reflect the new points of view. Thus, the creation of new conceptual classifications is an inherent feature of any advancing science. However, the fact that all new concepts can be communicated from one person to another supports our contention that the basic semantic elements of the vocabulary in which the concepts are expressed are relatively constant within a particular language community. By combining and relating these basic elements in many different patterns NL is able to express the infinite variety of percepts and concepts.

7. THE STRUCTURE OF THE SEMEM1C LANGUAGE

We will briefly describe some of the methods by which semons can be discovered and how they might be incorporated in a SL which may serve as an interlingua in algorithms for translation and also in other IR processes.

7.1 A number of linguists, including Hjelmslev (1961), have suggested that methods used to find phonological and syntactic elements can be adapted for semantic analysis. A common and fruitful method is that of commutation: two units which are able to replace each other with no change in the response of speakers of the language are held to be variants of the same element. Thus the different sounds produced for the first letters in *kit, cat* and *cut* are held to be variants of the phoneme /k/, On the semantic plane, two units (i.e. lexemes) may be held to have the same semantic value (i.e. the same sememe or SF) if they are mutually replaceable.

7.2 *Synonyms.* — K. Sparck Jones (1965) has described a method for finding groups (or 'rows') of synonyms from definitions in dictionaries. (The use of general dictionaries to establish communal language usage is obviously valid.) When a dictionary does not define a word directly, e.g. by describing its referents, but instead indirectly by listing its synonyms, these synonyms and the word itself form a 'row' of lexemes all having the same sememe. Many words belong to more than one row. For example, the word *activity* occurs in the following:

action activity briskness liveliness animation
activity animation movement
activity motion movement
activity briskness liveliness quickness speed
etc.

Obviously the meanings of these rows are quite close and we could try to form larger groups of synonyms, i.e. draw the distinction between, lexemes less precisely by assigning the same sememe. To do this Sparck Jones experimented with the statistical technique of 'clumps' with a certain degree of success. It must, however, be kept in mind that many lexemes will appear in different rows because they are homonyms.

7.3 James H. White (1964) has attempted to establish the semons of English prepositions by examining the contents in which substitution by other prepositions or phrases is possible: e.g. *of* may be replaced by *made of (in strips of wood),* by *from* or *taken from (few of them, one of our biggest organizations),* by *in (heroes of Homer's Iliad, natives of Africa),* by *about (story of the Trojan war),* etc. Because they occur in so many idiomatic phrases, prepositions are likely to prove extremely difficult to analyse.

7.4 Antonyms can be treated in much the same way as synonyms, since an antonym when negated may serve as a synonym, However care must again be exercised over homonymous lexemes; *soft* can be the antonym of *hard* as well as the antonym of *loud,* but *hard* and *loud* are not synonymous.

7.5 *Paraphrases.* — In many cases lexemes do not have synonyms but may nevertheless be replaced by a sequence of lexemes with the same meaning (paraphrases: section 3). Dictionaries give many examples of definitions in the form of paraphrases. If every paraphrase were to be reformulated in a phrase using a restricted set of lexemes (i.e. by 'paraphrasing' each paraphrase) and the process were repeated successively, the vocabulary used in definitions would be reduced eventually to those lexemes which cannot be substituted by any other lexemes and which consequently could be regarded provisionally as having sememes with only one semon. We must, however, be careful once more that no lexeme is used homonymously in the definitions. Such would be the danger if we adopted the vocabulary of Basic English; but, it must be acknowledged, the definitions in the Basic English Dictionary do illustrate the practicability of the method.

7.6 Kinship terminology provides an example of a set of lexemes having an apparently well-ordered network of semantic relationships: *uncle* is replaceable *by father's brother* or *mother's brother, aunt* by *father's sister* or *mother's sister, sister* by *female sibling, father* by *male parent,* etc., etc. Clearly it is possible to express all kinship relationships with a very small number (perhaps four) of basic lexemes.

Of course, dictionary definitions which are phrases consist of more than just a string of lexemes, they also have syntax. Thus a sememic dictionary would define *historian* not as a simple set of sememes '(man), (write), (history)' but would also indicate the relationship between them. The semantic analysis of syntactic relationships will be considered below.

7.7 *Classification schemes and semantic analysis.* — As we have already mentioned (section 2.4), classification schemes order lexemes according to their semons: contiguous lexemes have many semons in common, and lexemes more distantly related have few common semons. The NL translations provided in the schedules are considered, by the compiler at least, to be equivalent in meaning to the IL lexemes. An examination of these NL translations and the relationships between them in the classification may help us discover the semons of NL lexemes. For example a hypothetical scheme may have the following IL lexemes and NL translations:

Ab	Sheep
Aba	Ram
Abb	Ewe
Abc	Lamb

Assuming it is well structured, we can infer from this classification that the semons of 'Ab' form a subset of the semons of 'Aba', and also of 'Abb' and of 'Abc'. We may be able to infer also that the same semantic relationship holds between their NL translations, i.e. that the semons of 'Sheep' are a subset of the semons of 'Ram', 'Ewe' and 'Lamb'.

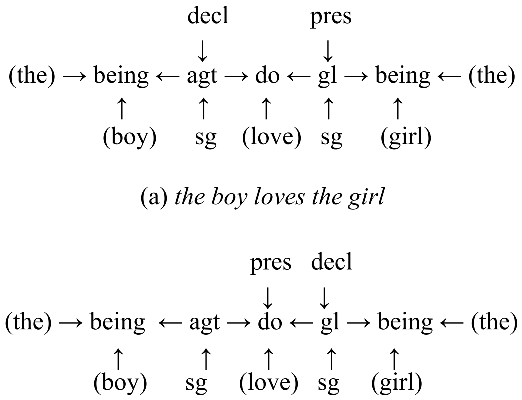
Such inferences find support in statements by classificationists recognizing the importance of a subject's terminology for the establishment of the bases of its classification: e.g. Vickery (1966: 38) "Facet analysis is one form of ... conceptual analysis. It begins by collecting terminology. The collected terms form the raw material of analysis.... The resulting faceted schedule is a conceptual scheme, a structure in which terminologically expressed concepts have been organized." However, this quotation also illustrates the major difficulty in using the results (or the methodology) of non-linguistic analysis: we cannot always be sure that the semons discovered are not connotations, i.e. that they are truly communal. There are, as we have said, many ways of conceptualizing percepts and there are many possible conceptual classifications. Library classifications are frequently biased to a particular point of view.

7.8 *Syntax.* — Analogous to the synonym relation between lexemes there are, as we mentioned in section 3, certain phrase-structures which are semantically equivalent and hence commutative. We take again our active-passive transformation: *the boy loves the girl* and *the girl is loved by the boy.* In transformational grammar the passive construction can be derived from the active by a 'transformational rule' applied to the whole terminal string of the active form. A transformational rule such as:

$$X_1 — X_2 — X_3 \rightarrow X_3 + \text{be} + \text{en} — X_2 — \text{by} + X_1$$

when applied to the tagmeme string for the *boy loves the girl,* Art + Nsg_1 + VtPr_2 + Art + Nsg_3, forms the string Art + Nsg_3 + be + en + VtP_2 + by + Art + Nsg_1 which underlies th*e girl is loved by the boy.* With such rules transformational grammar tackles the problem of semantically equivalent phrase-structures on the purely syntactic level.

7.9 Lamb (1964) has a different approach. He constructs graphs to illustrate both semantic and syntactic relationships. The synonymity of two phrases is apparent from the equivalence of their sememic graphs. Our two appear as:


decl
pres
(the) → being ← agt → do ← gl → being ← (the)
(boy)
sg
(love)
sg
(girl)
pres
decl
(the) → being ← agt → do ← gl → being ← (the)
(boy)
sg
(love)
sg
(girl)

(a) *the boy loves the girl*

(b) *the girl is loved by the boy*

Fig. 4

The graphs are exactly the same for both except for the position of 'decl', since it is this that determines whether the agent (agt) of the verb (do) or the goal (gl) is to be the grammatical subject. With small emendations the graph can represent many other NL sentences. For example if in (a) the 'sg' is changed to 'pl' it represents *the boys love the girl.* If in (b) the 'pres' is changed to 'past' it gives *the girl was loved by the boy,* and if changed to 'pot' it gives *the girl may be loved by the boy.*

In certain formal respects Lamb's graphs resemble the 'criterion trees' of the SMART system (Harvard Univ., 1964): (i) 'criterion trees' consist of nodes representing 'concepts' (i.e. sememes) and edges representing 'dependencies' (i.e. syntactic relationships), but (ii) SMART does not reduce synonymous ‘criterion trees’ to one single form. Instead, for the purposes of phrase-matching procedures, their semantic equivalence is simply recorded.

7.10 *Logical syntax,* — The formalization of syntax has been a subject of intense study by logicians. In predication a genera! term is joined to a singular term (cf. section 6.3); the variety of ways this can be achieved in NL is irrelevant to logicians: “*Fa* [can be] understood as representing not only ‘*a* is an F’ (where ‘F’ represents a substantive) but also ‘*a* is F’ (where ‘F’ represents an adjective) and ‘*a F*s’ (where ‘F’ represents an intransitive verb)” (Quine, 1960: 96), and “parallel to the form of predication *Fa* for absolute terms there is, for relative terms, the form of predication *Fab:* ‘*a* is *F to b*’, or ‘*a F*s *b*’” (p.106). For logic the reduction of syntactic structures to a restricted set of canonical forms facilitates the examination of the conceptual processes of deduction and inference (prepositional calculus). The meaning of the variables (‘a’ and ‘F’ in the above) is irrelevant, only their relationship is significant. In this, logical syntax has obviously different aims from a formalization of NL syntax. In addition, logical syntax has been concerned almost exclusively with simple forms of declarative sentences, and these form but a small subset of all NL sentences. Nevertheless, logic can and will contribute much in syntax formalization and in the methodology of language analysis.

8. THE FORM OF SFs

For the purpose of demonstrating how sememic formulae (SFs) may be used in IR processes, we adopt a rather simplified version of Lamb’s graphs. A SF graph consists of nodes representing sememes (in round brackets) and of labelled edges representing syntactic relationships. The SF graph for *the boy loves the girl* (cf. Lamb’s version, fig. 4(a)) appears as:

$$\begin{tikzcd}
\text{(the)} \arrow[r] & \text{(boy)} \arrow[r, Rightarrow, "\text{agt}"] & \text{(love)} \arrow[r, Rightarrow, "\text{gl}"] & \text{(girl)} & \text{(the)} \arrow[l] \\
 & \text{sg} \arrow[u] & \text{pres} \arrow[u] & \text{sg} \arrow[u] &
\end{tikzcd}$$

Fig. 5

In a SF graph a sememe may be associated with any tagmeme, and any sememe may be dominated by ‘decl’: thus, (i) if ‘decl’ dominates ‘(boy)’ and both ‘(boy)’ and ‘(girl)’ are nouns ‘(love)’ a verb in ‘past’ we have *the boy loved the girl; (*ii) if ‘decl’ dominates ‘(girl)’ and the tagmemes remain as in (i) we have *the girl was loved by the boy,* and (iii) if ‘decl’ dominates ‘(love)’ and it is a noun we have *the love of the boy for the girl* or *the boy's love for the girl,* etc.

In section 7.6 we quoted an example where the lexeme *historian* was assigned the sememes ‘(man), (write), (history)’. We can now supply the edges and form a SF graph: (man) $\xrightarrow{agt}$ (write) $\xrightarrow{gl}$ (history). If this SF is now put in place of ‘(boy)’ in the SF of fig. 5, the following SF network is formed:

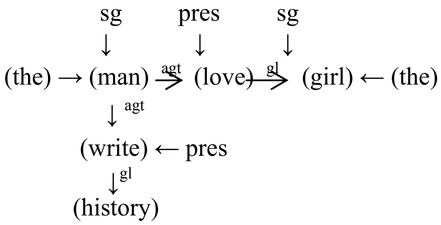


Fig. 6

Depending on where the dominating ‘decl’ is placed, on the grouping of sememes into lexemes, on the tagmemes selected for them, and on which syntactic structure is chosen, the following phrases may represent this SF in NL: *the historian loves the girl, the writer of history loves the girl, the girl is loved by the historian, the man writes history and loves the girl, the man is a historian and loves the girl, the man who writes history loves the girl, the man who loves the girl writes history, the man who loves the girl is a historian; the lover of the girl writes history, (he girl's lover is a historian, the historian is the girl's lover,* etc. In the last examples, analogous to the representation of (man) $\xrightarrow{agt}$ (write) $\xrightarrow{gl}$ (history) as *historian,* (man) $\xrightarrow{agt}$ (love) is represented by *lover.*

Many more NL sentences are possible with a change of any ‘sg’ to ‘pl’ and of the tenses of the verbs. Graphs more complex than this are needed for many NL sentences, but the example does show that a basically simple pattern is capable of expressing the SFs of a wide variety of NL sentences.

9. ANALYSIS: TRANSLATION FROM NL PHRASES INTO SFs

Having outlined the sememic code for NL and illustrated SF graphs we are now able to attempt to formulate the NL → SL algorithm.

9.1 *Syntactic analysis and problems of syntactic ambiguity.* — Workers in machine translation have developed a number of methods for the automatic syntactic analysis of NL sentences. (For an introduction to the various methods see Hays, 1967: chapters 6-8.) Very briefly, syntactic analysts operates as follows: A dictionary provides for each word (either for every form of the word or for its stem and suffixes separately) both its set of semons (sememes) and its tagmeme(s). Thus, in *the boy loves the girl, boy* may be assigned the tagmeme Nsg, *loves* the tagmemes VtPr and Npl, *the* the tagmeme Art and *girl* the tagmeme Nsg. The syntactic analyser examines whether these tagmemes form a tagmeme-sequence permitted by the syntax rules. Our phrase has two alternative sequences depending on whether *loves* is assigned VtPr or Npl: Art + Nsg + VtPr + Art + Nsg and Art + Nsg + Npl + Art + Nsg; but only one of these, the first, is acceptable by the rules of English syntax (as given in section 2.5 and fig. 2).

9.2 Systems differ chiefly in the ways in which syntactic rules are incorporated in the syntactic analyser. The differences are attributable to the linguistic models selected for the interpretation of English syntax. However, it has been shown (Gross, 1964) that the models used are all 'context-free grammars'. By their very nature context-free grammars are incapable of deciding which of alternative analyses of a sentence may be correct in a given context. For example, the sentence *they are flying planes* may be analysed (correctly in English syntax) in two different ways:

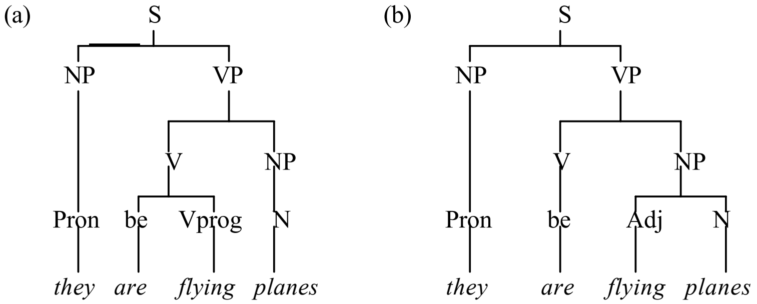


Fig. 7

A grammar which is 'context-free' as opposed to 'context-sensitive' cannot decide between them. Similarly, it cannot decide on the correct analysis of:

(i) *John is easy to please*

and

(ii) *John is eager to please*

Both sentences will be given the two alternative analyses: (a) where *John* is the object of *to please,* and (b) where *John* is the subject of *to please.* A human speaker intuitively selects (a) for (i) and (b) for (ii), but the machine cannot — in the present state of automatic syntactic analysis. However, workers in machine translation are now fully aware of the need for context-sensitivity in their systems and some progress is being made in this direction, e.g. by introducing rules from transformational grammar (cf. section 7.8 and also Hays, 1967: ch. 8).

9.3 *Semantic ambiguity.* — Similar difficulties over ambiguity are encountered on the semantic plane. A context-free analysis cannot discover which of the three possible interpretations of *bank* is intended in *The bank was the scene of the crime* — a human can do so only by reference to earlier or later sentences. There are, however, methods of resolving some homonyms mechanically.

Firstly, certain sememes may be excluded by syntactic analysis. For example the lexeme *fast* can represent three sememes: in one case it can function only as a noun or verb, e.g. *the holy man's fast, the holy men are fasting,* and in the others only as an adjective or adverb, e.g. *the fast train, to hold fast.* Secondly, within a given phrase certain sememes are excluded by their incompatibility with the sememes or semons of other lexemes in the phrase. For example, *soft* can represent two sememes, one with a semon '(audible)' which occurs in phrases such as *soft voice* the other with a semon '(tactile)' occurring in phrases such as *soft touch.* In interpreting the noun phrase *soft caress* the semon '(audible)' is found to be incompatible with the sememe for *caress.* The semon '(tactile)', on the other hand, is not incompatible. Thus, in this context, the sememe for *soft* with the semon '(tactile)' is selected. Thirdly, within a given phrase certain sememes are more probable than others. For example, in *The bank raised its rate of interest* the meaning of *bank* as 'a steep slope' or 'rising ground bordering water' is much less likely than its meaning as 'an establishment for monetary exchange'. Assuming the semon '(money)' is present in some sememes for *bank, interest* and *raise,* homonymity can be resolved by comparing the sememes of each lexeme and selecting those sememes with this common semon.

9.4 *Transformation into SFs.* — After the analysis of a phrase in terms of its sememes, the tagmemes attached to them and the syntactic structure joining all together, it is now to be transformed into a SF graph, (Hays [1967; 166-168] may again be consulted for a brief theoretical discussion of the automatic conversion of syntactic trees into sememic graphs). Our example *the boy loves the girl* is taken once more. The analysis of its phrase-structure as Art + Nsg + VtPr + Art + Nsg and its sememes as '(boy), (girl), (love), (the)' gives the string:

Art(the)Nsg(boy)VtPr(love)Art(the)Nsg(girl)

To transform this string into a SF, the algorithm must be able to perform the following:

(i) Recognise that the tagmeme-sequence Nsg...VtPr...N... indicates that the first N is the 'subject' of Vt and that therefore the sememes to which they are attached may be linked by the SF edge $\xrightarrow{agt}$ as: (boy)$\xrightarrow{gl}$ (love)

(ii) Recognise that in the same tagmeme-sequence the second N is the object' of Vt and that the sememes to which they are attached may be linked by the SF edge $\xrightarrow{gl}$ as: (love) $\xrightarrow{gl}$(girl)

(iii) Recognise that Art(the) is linked directly to the next following N as: (the) → (boy) and (the) → (girl).

With the appropriate attachment of '(sg)' semons we now have the SF graph illustrated in fig. 5.

9.5 Our second example is closer to the kinds of phrase encountered in indexing, and it is given in a slightly more rigorous formalization.

(i) *Destruction of timber by beavers* is analysed as:

Nvt(destroy)Pposs(of)Nin(timber)Pagt(by)Nan(beaver)

(ii) The algorithm identifies that Nvt has an essentially verbal function; since a Vt must have two connected edges $\xrightarrow{agt}$and $\xrightarrow{gl}$, so too must a Nvt.

(iii) The node at the other end of an 'agt' edge must be a N; in the case of a Nvt this N may occur as (a) a Ngen preceding Nvt or (b) PagtN succeeding Nvt. The sequence Pagt(by)Nan(beaver) fulfils (b) and so the 'agt' edge can be completed: (beaver)$\xrightarrow{agt}$ (destroy)

(iv) The node at the other end of a 'gl' edge must be a N; in the case of a Nvt this N may occur as (a) a Ngen preceding Nvt or (b) PpossN succeeding Nvt. The sequence Pposs(of)Nin(timber) fulfils (b) and so the 'gl' edge can be completed: (destroy) $\xrightarrow{gl}$ (timber)

(v) All sememes and tagmemes having been accounted for the whole SF is:

$$\begin{tikzcd}
\text{(beaver)} \arrow[r, "agt"] & \text{(destroy)} \arrow[r, "gl"] & \text{(timber)} \\
\text{pl} \arrow[u]
\end{tikzcd}$$

9.6 The practical feasibility of an algorithm for transforming NL into such a SL is demonstrated in part by procedures in the SMART system

(Harvard Univ., 1964) for the syntactic analysis of NL phrases (by the Multiple-Path Syntactic Analyzer developed at Harvard) and their conversion into 'criterion trees' — which, as we remarked in 7.9 above, have some similarities with SF graphs.

10. SYNTHESIS: TRANSLATION FROM SFs TO IFs

As we saw earlier, compilers of ILs attempt to simplify the variety of NL and to restrict the expression of a particular 'concept' to only one possible IF. Thus, whereas in NL there may be many different expressions for a single SF (cf. the example in section 8), there ought in an IL to be never more than one IF for any given SF. Those ILs which do admit synonymity, e.g. ILs basing the selection of the lexemes for the IF of a document on the vocabulary of the document itself without reference to the vocabulary of other documents, are considered to be inherently inefficient languages for IR systems.

While for one SF there is only one IF, the opposite is not true. There may well be more than one SF represented by a single IF. ILs with this feature lack expressiveness in comparison with the SL and consequently with NL. Librarians and IR workers are well aware of the drawbacks of such ILs.

10.1 The task of the SL → IL algorithm is in essence: to select the lexemes of the IL expressing the semons of the SF and, if more than one IL lexeme is needed in the IF, to ensure that the syntactic relationship made between lexemes conforms to the syntax rules of the IL (e.g. as given in section 2.6).

To illustrate we take the example of the preceding section, *Destruction of timber by beavers,* in its SF; (beaver)$\xrightarrow{agt}$ (destroy) $\xrightarrow{gl}$(timber).

10.2 *Semantics.* — There is considerable variety among ILs in their ability to express particular SF sememes,

(a) In some ILs there may be lexemes mirroring this group of sememes exactly. An IL using lexemes from NL may translate them back into the 'original' NL lexemes, as: 'Destruction', 'Timber', 'Beavers'.

(b) Some ILs may express two sememes by one IL lexeme, e.g. 'Xab' may express '(destroy)$\xrightarrow{gl}$(timber)'. Cf. the Dewey translation of '(history) ← (U.S.A.)' as '973'.

(c) Some ILs may be able to translate the SF only by 'overlapping' lexemes. For example, an IL may have one lexeme for 'Destruction of timber by animals' and another for 'Beavers'. Their conjunction in an IF results in the duplication of the sememe '(animal)'.

(d) Some ILs may have no syntactic facility for adding the specific 'Beavers' to the lexeme for 'Destruction of timber by animals' in IFs (even though, perhaps, 'Beaver' may legitimately occur alone or in other IFs).

(e) Another IL may be even less expressive than this since it may have a lexeme for 'Destruction by beavers' but none for 'Destruction of timber'. Thus no IF can be formed to specify (timber)'.

10.3 *Syntax.* — According to the IL there are basically three ways of expressing SF edges.

(a) They may be all translated as the same symbol: one undifferentiated link, e.g. 'Destruction: timber: beavers'. The order of lexemes is important for the interpretation of such an IF by index-users, since the order conveys a certain amount of implicit syntactic information. Lack of knowledge of the IL syntax is a common source of misinterpretation, as Vickery showed in the example 'Destruction, bacteria, dyestuffs' which may be understood as either 'Destruction of bacteria by dyestuffs' or 'Destruction of dyestuffs by bacteria.'

In section 2.6 one system of IL syntax was described which categorized lexemes as 'Thing, Part, Material, Action, Property'. There is clearly no correlation between SF edges and such categories. They can be derived only from constituent semons of SF sememes, e.g. in the following way: if a semon of '(destroy)' is considered to belong to the category 'Action', then the IL translation for '(destroy)' will be ordered by the syntax rules according to this category in IFs.

As we also saw, faceted classifications also categorize lexemes and order them in IFs by this categorization. Thus, in a hypothetical scheme '(beaver)' may be expressed as 'Vz', '(destroy)' as 'G' and '(timber)' as 'Af', giving for the whole SF: 'AfGVz' — the facets being ordered alphabetically.

(b) In other ILs SF edges may not be translated at all and lexemes are entered separately in the index file. When this is done (e.g. in co-ordinate indexes) syntactic relationships between lexemes derived from one SF are left for the index-user to make during his search of the file. Naturally enough he may provide the wrong syntax (i.e. translating the SF edges incorrectly) or he may link lexemes from different SFs (i.e. descriptions of different documents).

(c) In some ILs SF edges may be roughly translated as logical relationships, such as ‘roles’, ‘interfixes’ etc. If, for example, an IL has the role indicators A ‘subject of process’, B ‘process’ and C ‘agent of process’, then for our SF they would be attached to the lexemes for ‘(timber)’, ‘(destroy)’ and ‘(beaver)’ respectively. A relatively simple algorithm could be devised to attach such ‘roles’ to lexemes: role C would be attached to a sememe with an ‘agt’ edge leading from it, role A to a sememe with a ‘gl’ edge leading to it, and role B to sememe with both an ‘agt’ edge and a ‘gl’ edge.

Interfixes indicate a direct relationship between two lexemes, e.g. Beavers (1), Destruction (1) (2), Timber (2). Lexemes with interfixes may now be entered separately in the file. Interfixes could be derived from SF edges by a simple algorithm selecting a unique interfix symbol for each edge and assigning it to the sememes at both its ends.

II. CONTENT ANALYSIS

The human indexer does not find brief descriptions for documents ready-made either in IL or in NL. He arrives at descriptions either (a) by expressing in his own terms (in NL, or, if an experienced indexer, directly in IL) the content he has ‘extracted’, by some means or other (largely obscure), from the meanings of each word, phrase, sentence, paragraph, etc. of the documents or (b) by selecting a NL phrase or individual NL words from the document which he considers, by some criterion or other, to be an adequate and fairly complete summary of its content: if a phrase is selected it is frequently the document’s title (i.e. the author’s own summary of the content).

11.1 The use of the SL in content analysis is roughly analogous to the first procedure. However, before describing a possible system we must be clear about what kind of content analysis is required and is attainable by any mechanized process.

An index-user approaches an index file in the hope of finding references to documents of interest to him. After reading documents denoted by a particular IF of the file he may find that the ‘message’ of some of them is known already, i.e. they convey little ‘information’ to him. Others, however, may have high ‘information content’ if they convey facts previously unknown to him and important for his work. Another index-user may select the same set of documents by the same IF and yet find that in his case quite different subsets have high and low ‘information content’.

11.2 Just as the semantic value of words (their sememes/semons) is objectively determinable and also the meanings of phrases and sentences, it follows that the semantic content of a sequence of sentences, i.e. a document, ought to be also objectively determinate. But the 'information content' of a document for a particular reader at a particular time cannot be — it is a subjective matter which no indexer can presume to prejudge, Of course, indexers can and do bias their interpretations of content in favour of the known interests of a particular group of people. What they do in effect is to ignore that part of a document's 'semantic content' which is common knowledge in the group and to describe only that part with potential 'information' for members of the group. In an automated system bias could be introduced easily — e.g. by assigning greater weights to some elements than others and by excluding other elements entirely — but only if the basic system is able to analyse first the objective 'semantic content'.

11.3 In fig. 6 (section 8) we gave the SF for *the historian loves the girl.* The graph consists of two SFs: (man) $\xrightarrow{agt}$(write)$\xrightarrow{gl}$(history) and (man)$\xrightarrow{agt}$(love)$\xrightarrow{gl}$(girl) with a common sememe '(man)' joining the two into one SF. Both SFs may represent NL phrases which can stand independently in a text: *The man writes history... The man loves the girl.* Let us suppose that both these sentences are encountered in a certain document. The algorithm described in section 9 transforms first one into its SF and then the other into its SF. Since both SFs have a common sememe they are made one SF by another algorithm — the algorithm for forming SF networks. Later in the document may occur the sentences: *The historian works at the university, The girl breeds horses, The man is American.* Since the SFs for these sentences have sememes in common with the SF already formed they can be incorporated in the same way into the network to form:

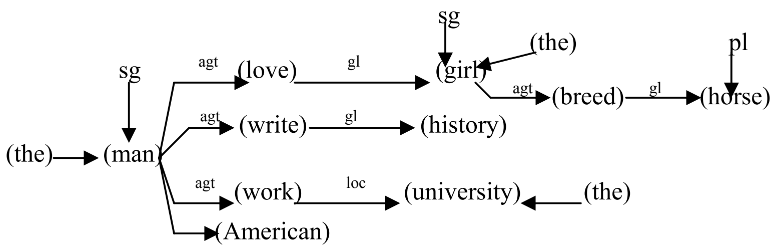


Fig. 8.

This process involving new sentences and their SFs could continue and form a more and more complex network. However, not all sentences would add new sememes and edges, i.e. they would add nothing to the 'semantic content'. For example, the SF for *The girl who breeds horses is loved by the American historian* is already included in the above SF network. It is well known that NL texts have a great deal of semantic redundancy so the SF network for a document, though complex, ought to be more compact than the full NL test. No part of the message would be omitted, but equally it would include no repetition.

Apart from the general problems of syntactic and semantic analysis (cf. sections 9.2 and 9.3) and, of course, problems of computer technology, the major difficulties in this system of content analysis lie in our almost complete lack of knowledge about how sentences are related to each other in continuous discourse. It has been assumed, for example, that in our document every time *the man* occurs the same referent is intended. In some cases this may not be true; but how is a mechanical device to 'know' this?

11.4 Most present systems of automatic content analysis involve the selection of significant words from documents by statistical methods (Stevens [1965] gives a comprehensive survey and critique). The words selected are used as the 1L lexemes in a description of the document. The systems are analogous to the second method of human content analysis.

The basic difficulties found in this approach stem from the obvious crudity of attempting to extract the 'content' of a text from the frequency and rarity of the words occurring in it, i.e. to proceed as if words were (he basic units of meaning and without making an analysis of NL vocabulary. The statistical approach would be on a firmer basis by using semons, NL texts have much semantic redundancy which would be manifest in the repetition of semons, Thus, the relative frequency of particular semons would indicate the document's main 'semantic content'. There would still be problems of statistical significance and the dangers of unintended bias but it is conceivable that an analysis of semon occurrence would be a far better indicator of content than an analysis of word occurrence.

11.5 In practice, no doubt, any automated SL-based system would probably adopt a combination of both approaches. Whether any system will ever be entirely satisfactory even in the extraction of 'semantic content' is highly debatable. For this reason, it is worthwhile to consider the alternative approach to 1R which mechanises the ‘scanning’ or ‘browsing’ process and by-passes content analysis (fig. *2,* and section 13).

12. SEARCHING INDEX FILES

After a query has been translated from NL into a phrase in the language of the index file being consulted, an index-user must find an IF in the file (IFf) to match the IF of his query (IFq).

If he is lucky he will find an exact match, i.e. IFq = IFf, and he can now go to the documents indicated by the IFf.

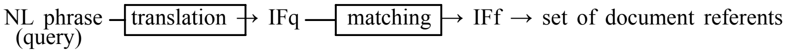


Fig. 9

If he fails he has two courses open to him:

12,1 (i) He can reformulate his query in NL so that it now consists of more than one phrase. For any document to be relevant each phrase must be an apt description of part of its content. He then translates each NL phrase into an IFq and again searches for matches with IFf's. If the matching is successful he can now find the documents he wants by examining those indicated by each IFf and finding their common members (i.e. *by* co-ordination or logical product).

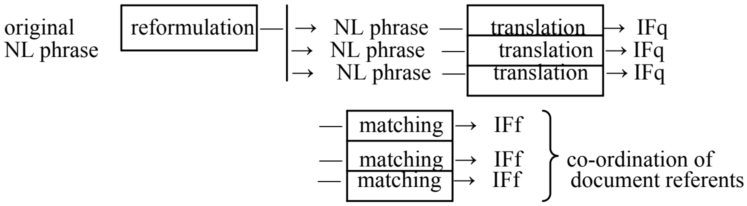


Fig. 10

Other logical operations than co-ordination can be performed in searching and there are many papers in IR discussing their application and efficiency. But since these are operations not upon IFs but upon their referents (sets of documents) and consequently not involving semantic relationships, they lie outside our concern here with the linguistic problems of searching. Similarly we also ignore other related aspects, such as recall and relevance ratios.

12.2 (ii) If the reason for failure is the absence of any referents for his IFq or the apparent inability of the 1L to express his query, then he must reformulate the query so that it is either more specific or more general than the original. In the selection of his new IFq the index-user may be guided by the 1L itself, i.e. by its in-built classificatory structure: the juxtaposition of semantically related lexemes in a classification scheme, or the cross-references in an alphabetical subject index. If he is not so guided (or does not choose to be) he must rely on his linguistic intuition to find for him a near-paraphrase in NL.

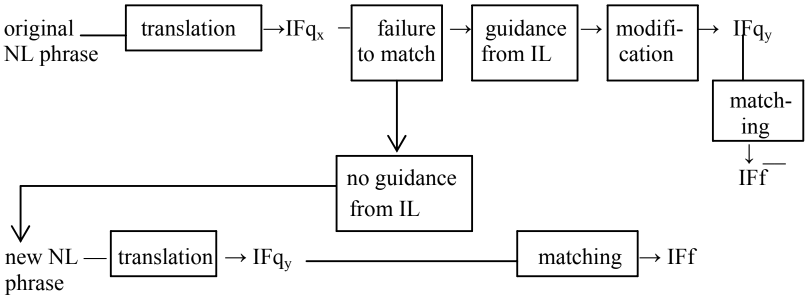


Fig. 11

For the purposes of searching index files there need be, in theory, no parallelism between the classificatory structure of ILs and the semantic structure of NL. The role of the classification is to help index-users to find the IF they want. There is no need for them to know the semantic structure of an IL (in fact most do not know it) since they are, or should be, guided from one IF to another by the IL itself and they can always tell whether they are approaching the IF for their query by examining the documents indicated. In practice, however, the less index-users agree with the basic premisses of the classification (the conceptualisation of the world underlying it) the more likely they are to be confused and unable to find the desired IF.

12.3 With this brief outline of human and semi-automatic searching systems we are now in a position to describe how an SL-based system might work.

We have already described in sections 9 and 10 how a query in N L can be translated into an IF. An exact match of the IFq with an IFf would obviously cause no difficulty, but if the result of the NL → IL translation is a set of IL lexemes unrelated syntactically (cf. section 10..3 (b)) the documents denoted must be co-ordinated — as in human or semi-automatic systems (fig. 10). As in all co-ordinating systems the risk is always present that the syntax provided by the index-user produces ‘false drops’. In contrast with existing systems there should never be any errors in translating a NL query into an IF. It may be that the IF is also a translation for other NL phrases which the index-user might not consider synonymous. But if the SL → IL translation algorithm is correct, then the IF must be too.

12.4 If there is no IFf for a given IFq the system must be able to modify the IFq as in existing systems. Since SFs are formed from a limited set of semons it would be possible to modify any SF from which an IFq has been derived by simply adding or subtracting semons. If the index-user wished to widen the scope of his search he could remove from the SF of his query one or more semons, or redefine the query in NL and create a new SF. The IF for this SF would be more generic than the old. By a similar procedure the IF could be modified to make the search more specific.

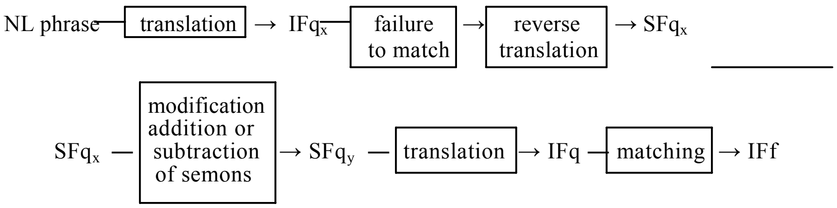


Fig. 12

An advantage of this method over present systems of generic and specific modification is that it would be entirely under the control of the index-user (through his use of NL) and not constrained by the structure of the IL.

13. SEARCHING UNANALYSED TEXTS

13.1 In another paper (Hutchins, 1967) the possibility was discussed of an IR system which matched queries in NL with phrases in documents which had not undergone previous content analysis and description as IFs and which, therefore, retained the NL forms used by their authors.

The suggested system (Quantras) transformed a NL query into sets of semantically equivalent NL phrases (i.e. paraphrases of the query, in the sense of section 3), which were then compared and matched with the NL, phrases of texts (fig. 13). Matches were assumed to indicate documents of potential interest to the inquirer. The major advantages of such a system would be that, since no content analysis is involved, no unintended bias would be introduced in IFs, no part of the message would be 'lost' to any future inquiry and the many complex technical problems of content analysis (cf. section 11) would be by-passed.

13.2 As described, the process of transformation could be performed, at the present state of linguistics and computer technology, by a program for syntactic analysis, a pool of (semantically equivalent) phrase structures and a dictionary of synonyms and near-synonyms. The system could also optionally include a classification of vocabulary for generic and specific searches.

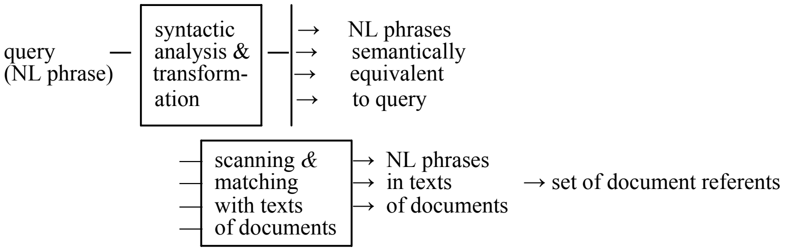


Fig. 13

13.3 This transformation process is equivalent to an algorithm for NL → NL translation when both NLs are the same (e.g. English). As we made clear in section 5 such a binary translation algorithm could well be fairly straightforward. In addition reasons were given in the paper for believing that further simplifications would be feasible: (i) the algorithm would need to deal only with the semantics and syntax of query phrases and not with all possible sentences; (ii) syntactic analysis would require no context-sensitivity (cf. section 9.2); and (iii) semantic analysis need not resolve ambiguities (cf. section 9.3). Because query phrases are almost exclusively 'context-free' it would be senseless to burden the algorithm of analysis with such unnecessary complexities. It would be up to the inquirer to formulate his query as unambiguously as he could.

13.4 As soon as other NLs in document texts must be dealt with the transformation process becomes more complex. Rather than compile algorithms for each NL it is more economic, for the reasons given in section 5, to employ an interlingua (the SL) and to design algorithms for SL → NL translation. The system would then be in three stages (fig. 14): (i) the query would be analysed as a SF — by the NL → SL algorithm described in section 9, but with the simplifications given in the previous paragraph; (ii) the SF would be transformed into all the NL phrases which may express it — by a SL → NL algorithm to be outlined below; and (iii) these phrases would be matched against the texts of documents — by the simple matching strategy of fig. *9,* with modifications performed as in section 12.4.

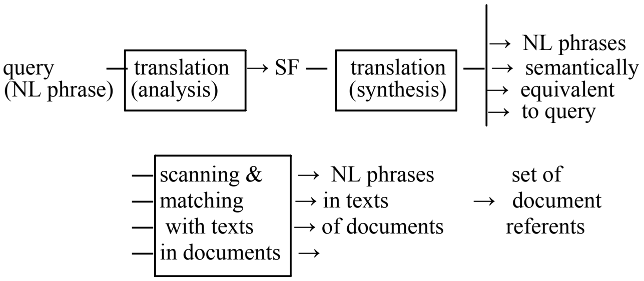


Fig. 14

13.5 The most complex component would be, of course, the SL → NL translation algorithm. However, some simplification is possible because, as explained in the paper (Hutchins, 1967), transformations resulting in 'meaningless' NL phrases need not be excluded since document texts do not presumably contain such phrases.

We will now attempt to outline briefly how a SL → NL algorithm might function. To illustrate we choose again the SF for 'Destruction of timber by beavers', i.e. (beaver) $\xrightarrow{agt}$ (destroy) $\xrightarrow{gl}$(timber), [In the following the subscripts 1, 2, 3 refer respectively to the nodes '(beaver)', '(destroy)' and '(timber)'.]

Taking each SF edge in turn the algorithm provides all the syntactic structures which may express it: (i) the 'agt' edge may be expressed by the tagmeme-sequences $N_1 + V_2$, $Ngen_1 + N_2$, and $N_2 + by + N_1$; (ii) the 'gl' edge may be expressed by the tagmeme-sequences $V_2 + N_3$, $Ngen_3 + N_2$, and $N_2 + of + N_3$. Joining together the tagmeme-sequences derived from SF edges, strings are formed for the whole SF which are permitted by the syntax rules of English (cf. section 2.5). All possible matchings are tested, some are not successful.

$N_1 + V_2$	and $V_2 + N_3$	→	$N_1 + V_2 +$ N3
$N_1 + V_2$	and $Ngen_3 + N_2$		
$N_1 + V_2$	and $N_2 +$ by $+ N_3$		
$Ngen_1 + N_2$	and $V_2 + N_3$		
$Ngen_1 + N_2$	and $Ngen_3 + N_2$	→	$\begin{cases} Ngen_1 + Ngen_3 + N_2 \\ Ngen_3 + Ngen_1 + N_2 \end{cases}$
$Ngen_1 + N_2$	and $N_2 +$ of $+ N_3$	→	Ngen1 $+ N_2 +$ of $+ N_3$
$N_2 +$ by $+ N_1$	and $V_2 + N_3$		
$N_2 +$ by $+ N_1$	and $Ngen_3 + N_2$	→	$Ngen_3 + N_2 +$ by $+ N_1$
$N_2 +$ by $+ N_1$	and $N_2 +$ of. $+ N_3$	→	$\begin{cases} N_2 + \text{by} + N_1 + \text{of} + N_3 \\ N_2 + \text{by} + N_3 + \text{of} + N_1 \end{cases}$

Attaching sememes to these structures and finding the correct lexeme for the sememe/tagmeme units produces:
N(beaver) + V(destroy) + N(timber) → beavers destroy timber ...
Ngen(beaver) + N(destroy) + of + N(timber) → beavers' destruction of timber
Ngen(timber) + N(destroy) + by + N(beaver) → timber destruction by beavers
etc.

14. CONCLUSION: THE PRESENT SITUATION

14.1 The chief aim of this paper has been to provide the basis for a formalization of the linguistic processes involved in the selection of documents for their content and to suggest means by which they may be mechanized. Because the goal has been to provide a formalization for complete IR operations it has necessarily been probably too complex for systems devoted to the automation of particular subsystems. Thus, for example, the NL → IL algorithm can be simplified considerably if the only NL involved is English and the IL is based on English lexemes (cf. section 5.2). However, it is believed that a formalization on the lines suggested here can form the framework for systems which will be capable of expansion, e.g. from a manual system to an automated one, from a system restricted to one subject field to a general system, from a system dealing only with English to one dealing with many NLs, etc., etc.

14.2 With this in mind we can see the value of improving ILs (whether used in manual or in automated systems) towards a closer proximity with the structure of NL, so that they may express any concept which can be now, or may be in the future, expressed in NL. The ideal IL would presumably be the SL itself, for the following reasons:

(i) Its semantic structure would be based on actual language usage and not upon the conceptual patternings of the designers of the IL. In section 6 we pointed out that the possible conceptual classifications of the world are infinite but that the semantic structure of NL is relatively constant.

(ii) Its syntactic structure would retain as much of the variety of NL syntax as necessary. It would not, unlike many existing ILs, abandon NL syntax altogether or set up a system of logical syntax unrelated to language habits (cf. sections 2.6 and 7.10).

(iii) It would be as flexible and expressive as the best ILs, because, being based on NL structure, it would be able to express any concept which can be expressed in NL.

(iv) If incorporated in an automated system, no SL → IL algorithm would, of course, be required, Thus, the translation processes would be greatly simplified.

(v) Being based on language usage, it would be more easily adaptable to changes in semantic structure: in many fields of learning and research terminology is frequently redefined and new terms are created.

If the system were automated, modification of the semantic structure could well be achieved by internal procedures. For example, a new lexeme could be added to the store and its semon structure obtained from an analysis of a NL definition provided for it (i.e. its SF could be obtained by the NL → SL algorithm). Similarly new sememes for lexemes already in the store (e.g. new specialist usages) could also be introduced in this way.

(vi) Since the SL would also serve as an interlingua in NL → NL translation, the indexing and searching of documents in non-English NLs would be no slower and no less consistent than for English documents.

14.3 The development of ILs in this direction is already evident. Some recent classification schemes incorporate features close to NL structure. Syntol (Gardin, 1965) is an example of a non-automated classification scheme and SMART (Harvard Univ., 1964) one intended for a particular automated IR system.

Syntol incorporates logico-linguistic features which have much format similarity (but no parallelism) with the structure of SL as outlined here. It has two basic components: a paradigmatic organization by which descriptors (IL lexemes) are related in a hierarchy (i.e. a conceptual classification), and a syntagmatic organization by which descriptors are related in document descriptions (IFs). The 'syntagmas' are formats for expressing logical syntactic relationships which in other ILs may be indicated by roles, finks, facets, etc. (cf. section 2.6), i.e. they are not comparable with NL syntax or SF edges.

SMART also has close formal similarities with SL structure. As we saw (section 7.9) its 'criterion trees' strongly resemble Lamb's sememic graphs — on which our SFs are based. Relationships between 'concepts' (IL lexemes) are made, however, by a conventional tree hierarchy (i.e. a conceptual rather than a semantic classification).

14.4 As the analysis and formalization of NL by linguists advances we can expect the results to be incorporated more and more in classification schemes and in the structure of automated IR systems. The formalization of NL syntax has already had widespread influence. The as yet poor development of semantic analysis will mean that probably for some time to come IR workers will still depend on (largely intuitive) conceptual classifications of vocabulary. (We must not forget in this respect the contribution work in documentary classification can make to semantic analysis — section 7.7.)

However, as this paper has indicated, without the analysis and formalization of NL semantic structure no automated IR system can hope to tackle the major operations of indexing or searching, let alone emulate the efficiency achieved in these activities by experienced indexers and librarians. In future IR research much higher priority must surely be placed on the solution of semantic problems than has been done in the past.

REFERENCES

Andreev, N. D.
1967 "The Intermediary Language as the Focal Point of Machine Translation", *Machine Translation,* ed. A. D. Booth (Amsterdam, North-Holland), 1-28.

Antal, Laszlo
1963 *Questions of Meaning* (= *Janua Linguarum,* Series Minor 27) (The Hague. Mouton).

Coates, Eric
1960 *Subject Catalogues: Headings and Structure* (London, Library Association).

Gardin, J. C.
1965 *Syntol* (= *Rutgers Series on* Systems *for the Intellectual Organization of Information,* 2) (New Brunswick, Rutgers Graduate School of Library Service).

Gross, Maurice
1964 "On the Equivalence of Models of Language Used in the Fields of Mechanical Translation and Information Retrieval". *Information Storage and Retrieval,* 2. 1, 43-57.

Harvard University. Computation Laboratory
1964 *Information Storage and Retrieval* (= *Scientific Reports,* ISR-7 & ISR-8) (Cambridge, Mass.).

Hays, David G.
1967 *Introduction to Computational Linguistics* (London, MacDonald) (also: New York, American Elsevier, 1967).

Hjelmslev, L.
1961 *Prolegomena to a Theory of Language,* transl. by Francis J. Whitfield, rev. ed. (Madison, Univ. of Wisconsin Pr.).

Hutchins, W. J.
1967 "Automatic Document Selection Without Indexing", *Journal of Documentation, 23,* 4, 273-290.

Katz, Jerrold J.
1966 *Philosophy of Language* (New York, Harper *&* Row).

Lamb, S, M.
1964 "The Sememic Approach to Structural Semantics". *American Anthropologist,* 66, 3, 2, 57-78.

Lynch, M, F., and J.E. Armitage
1967 "Articulation in the Generation of Subject Indexes *by* Computer", 153rd *National Meeting of the American Chemical Society, Chemical Literature Division* (Miami Beach, Fla.); also: *Journal of Chemical Documentation,* 7, 170-I78,

Quine, W, van O.
1960 *Word and Object* (Cambridge, M.I.T. Pr.).

Sparck Jones, K.
1965 "Experiments in Semantic Classification", *Mechanical Translation,* 8, 3/4, 97-112.

Stevens, M. E.
1965 *Automatic Indexing: a State-of-the-Art Report (= Monograph,* 91*)* (Washingion, National Bureau of Standards).

Vickery, B. C.
1966 *Faceted Classification Schemes (= Rutgers Series of Systems for the Intellectual Organization of Information,* 5) (New Brunswick, Rutgers Graduate School of Library Service),

White, James H.
1964 "The Methodology of Sememic Analysis with Special Application to the English Preposition", *Mechanical Translation,* 8, 1, 15-31.

Wittgenstein, Ludwig
1929 *Tractatus Logico-Philosophicus* (London, Routledge).

University of Sheffield