PERSPECTIVE PAPER: COMPUTATIONAL LINGUISTICS

Naomi Sager
New York University

Prefatory Note

The observations regarding computational linguistics in this paper are limited to developments in the United States, and further, to work which employs some measure of linguistic (rather than purely statistical) analysis. The paper is not a survey, which would be redundant in view of the comprehensive reports on natural language processing provided by Walker (1973) and Damerau (1976), and in Sparck Jones and Kay (1973). Rather, it is organized around problems and solutions to problems, and it attempts to document the view that techniques of computational linguistics have reached the stage where applications in information science are now possible.*

Introduction

That computerized natural language processing might be a useful tool in information retrieval was an idea that came early in the history of computers, and at a time when information retrieval was still a largely unfamiliar term. In the 1950's, administrators in the Science Information Service of the U.S. National Science Foundation were already considering this possibility. At that time the impetus was theoretical. The enormous power of the computer as a symbol processor was beginning to be appreciated. At the same time, the discovery in linguistics of transformations appeared to open the way for linguistic normalization of texts by formal, in principle, computable, methods. It seemed reasonable to project that computer programs for language analysis could be used to extract the topics and relations of interest in scientific documents by direct processing of the texts themselves. The basis for this expectation clearly existed, although the prospect was not immediate.

The 20 years that have ensued have seen a tremendous growth in computer capabilities, including higher level languages which make complex tasks like natural language processing easier to formulate and systematize. But they have not seen the emergence of fast, practical language processors to do the work of information analysis and retrieval. The field of

----

*Throughout, where it seemed appropriate to include examples, I have drawn mainly on material from the Linguistic String Parser (LSP), developed by my own research group at NYU (for basic references, see Sager, 1973; Grishman, 1973; Grishman et al., 1973; other references are given below). This is not intended to slight the work of others but was the only convenient way such illustrations could be included. I have referenced a few sources which were outside the scope of the surveys mentioned above, but were relevant to the material presented in the paper.

computational linguistics has grown nonetheless. Systems and approaches abound, but on the whole these are no longer directed to text processing and no longer rely primarily on the methods of linguistic analysis. A small stream of research efforts, however, have persisted in the belief that there is an inherent, discoverable, and computable connection between language structure and information. The tools for extracting and exploiting this connection have been slow in developing for several reasons:

1. The natural language problem is more difficult, more complex, and requires more detailed preparatory work than was initially realized.

2. For many years the appropriate software was not available, and every linguistic computation had to be programmed in excruciating detail -- very time-consuming with results that were uninspectable and difficult to modify.

3. The needed store of detailed grammatical formulations and word classes were lacking over virtually all of English grammar and linguists were not interested in doing the type of work which would fill in the gaps.

4. Much time and effort was lost in trying short-cut solutions.

For these and other reasons, most workers abandoned the linguistic program, and turned to other areas, or restricted themselves to applications where the language processing could be limited to a few sentence types.

In Sparck Jones and Kay, a number of the substantive problems that discouraged computational linguists are noted, such as multiple parses and the seemingly inherent difficulty of mechanizing transformational decomposition. Considerable progress has been made in these and other problems so that today they are no longer obstacles to achieving useful applications of natural language processing, although as yet these applications are on a more limited scale than was envisioned at the start.

Sparck Jones and Kay also conclude that linguistics as such has had little influence on the information field. This is not surprising in the face of the tendency over the past two decades for linguists to be concerned primarily with theories of sentence generation. A generative grammar does not in itself provide for recognition, unless reversibility is included as a requirement of the system. On the other hand, information science is necessarily concerned with recognition. Language is the natural carrier, *encoder*, of information. If a theory of language is to shed light on information, it must deal, at least rudimentarily, with the problem of *decoding* the message. From this point of view, information science may have more to gain from certain developments in computational linguistics where the focus is on recognition, than from linguistics itself, at least within the theoretical climate of linguistics over the past 20 years.

Parsing

Despite differences in computational and linguistic frameworks it is generally agreed within computational linguistics (in practice if not in theory) that a syntactic analysis of a sentence is a necessary step in arriving at an explicit representation of its contents. While the gross structure of sentences in Indo-European languages is not difficult for us to state, its computation has not proved easy. For one reason, in the development of natural language parsers it was by and large assumed that syntactic structure meant phrase structure, and that phrase structure analysis could be performed by context free parsers, at least to the extent that the resulting parse trees could be used as inputs for further processing. This effort was doomed :o failure because in natural language, detailed constraints are involved at the various levels of description. Since the amount of detail that it is possible to express in context free rules is severely limited, this meant that the CF grammars that were used were necessarily inadequate. This problem is overcome by adding a component to the grammar which carries the detailed constraints, and applying these constraints to putative parse trees in the course of analysis. The parsing results then in fewer spurious analyses, and the parsing time stays within reasonable (experimental) bounds.

Other problems in parsing, however, are related to the inherent complexity of natural language, to the existence of ellipsis, to the rich network of conjunction and comparative constructions, and to the syntactic ambiguities which remain even when a strong grammar with many detailed grammatical constraints is used. The proper system framework is an important ingredient in the solution to some of these problems. It is a great help to work with a computational formalism that allows naturally for embedding, for applying constraints to parse trees, for dynamic generation of conjunctional constructions, for storing and passing along information conveniently, to mention just a few features which are now common in a number of systems. It should be kept in mind, however, that the use of a suitable formalism does not in itself provide solutions (a point which is sometimes overlooked by newcomers to the field). Where a creditable level of linguistic performance is required over a reasonably broad set of sentence types, there appears to be no substitute for an extensive grammar and word dictionary.

The problem of syntactic ambiguity was well illustrated by the passage quoted in Sparck Jones and Kay (1973, p. 86) from Robinson and Marks concerning a sentence for which they obtained 106 parses. The sentence was:

> *An exploratory investigation is undertaken to survey the extent to which the idea of undetermined coefficients can profitably be used in practice for root-extraction with polynomial equations.*

Robinson points out that "all sentences ending with two or more prepositional phrases or with a prepositional phrase following a noun object will receive multiple parsings; and if they may be attached to several verb forms, the multiplication of analyses is compounded further."

There is no general solution to this problem on the syntactic level, if one understands by *solution* the correct assigning of prepositional phrases (and other modifiers) to the element each modifies in the intended reading of the sentence. This would require applying selectional rules valid over the whole language, to use the distinction made in linguistics between grammatical rules and selection. Grammatical rules apply to all sentences of the language and deviations are recognizable as violations of grammaticality, whereas selection refers to the particular word-choice combinations that are made within grammatical categories to carry the specific content of the discourse. Certain word-choice combinations are frequent or accepted at a given time and in particular domains of discourse, but no hard and fast rules governing word-choice can be stated, as this would have the effect of legislating via the grammar what content can and cannot be said.

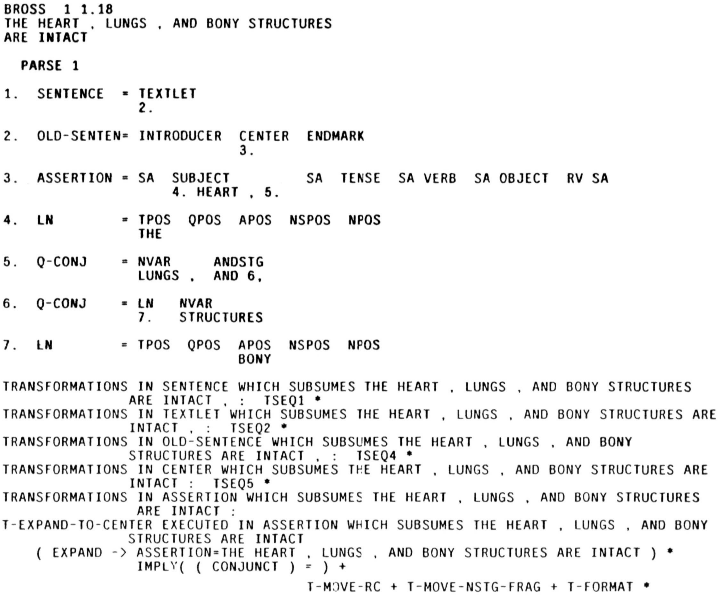
Since selectional constraints are in their nature subject-matter specific, they can only be applied when a particular subject matter, for which constraints are available, is being treated. Some workers have therefore limited their parsing to subject areas for which such rules can readily be stated. If we envision less restricted uses of natural language processing, some means which is independent of subject-matter constraints is needed to cope with -- if not solve -- the problem of multiple parses. For example, we can add to the parsing grammar the ability to distinguish multiple readings of an associative kind (e.g., the prepositional phrase problem), where the different readings do not affect the division of the sentence into its major syntactic components, from those which imply a radically different structural analysis of the sentence. We need not generate all the parse trees of the former type, provided subsequent stages of analysis can locate the implicated structure starting from any element in the sentence which might have a special relation to it. The Linguistic String Parser (LSP), for example, has such a pass-along device optionally available. The parser recognizes which structures are adjuncts (modifiers) and attaches a given adjunct, spanning a given sequence of sentence-words, in only one of the parse tree slots available for the given structure in otherwise identical parses. Later, when the transformational stage must examine some of these structures, chiefly prepositional phrases, to see if they are arguments of a nominalized verb--e.g., *of calcium* and *with other cations* as arguments of *exchange* in *The exchange of calcium with other cations*, a routine hunts for the prepositional arguments in adjunct slots adjacent to the nominalized verb. This device does not solve the ambiguity problem, but it enables the system to *get along with the job*, by passing information along to the point where it can be used without bogging down in excessive output at an earlier stage of analysis. Using this device, the LSP obtained one parse for the sentence discussed by Robinson, above.

Transformations

Parsing is rarely seen as an end in itself and often as the first step in a transformational decomposition. The great interest of transformations for linguistic computations, as Sparck Jones and Kay point out, is that they reduce the number of alternative grammatical forms for the same information. Thus, for example, two sentences which contain, respectively, *Ca^{++} exchanges with other cations in SR* and *the exchange of Ca^{++} with other cations in SR*, over these portions carry the same information, and it would be a definite gain to have only one representation for it. Furthermore, it is hoped that the structural representation of the sentence after a transformational decomposition is performed will be closer to an *ideal* representation of its contents than either the original word string or, in most cases, the syntactic parse.

The main steps in computerized transformational analysis are first to obtain a parse or *surface structure* for the sentence, and then to apply the appropriate sequence of reverse transformations so as to arrive at the desired underlying representation or *deep structure*. This has to be repeated for each initial surface parse tree, and for all the alternative intermediate trees created when more than one reverse transformation applies to the tree resulting from the previous application of a reverse transformation. Clearly, the process explodes unless (a) the parser delivers a reasonably small number of alternative surface parse trees to begin with; and (b) the number of reverse transformations which can apply to any intermediate tree is severely constrained.

Early transformational programs foundered on these two counts. Parsing grammars, as we have seen, were not strong enough to deliver only correct parses, and the problem of ambiguity remained. With regard to the reverse transformations, most workers were using the transformational generative grammar of Chomsky, which does not provide an explicit statement of surface structure/deep structure relations, and they did not have available detailed statements of the parse tree conditions which permit a given transformation to



```
BROSS  1 1.18
THE HEART , LUNGS , AND BONY STRUCTURES
ARE INTACT

  PARSE 1

1.  SENTENCE   = TEXTLET
                 2.

2.  OLD-SENTEN= INTRODUCER  CENTER  ENDMARK
                            3.

3.  ASSERTION = SA  SUBJECT         SA  TENSE  SA VERB  SA OBJECT  RV SA
                    4. HEART , 5.

4.  LN         = TPOS  QPOS  APOS  NSPOS  NPOS
                 THE

5.  Q-CONJ     = NVAR     ANDSTG
                 LUNGS ,  AND 6,

6.  Q-CONJ     = LN   NVAR
                 7.   STRUCTURES

7.  LN         = TPOS  QPOS  APOS  NSPOS  NPOS
                             BONY

TRANSFORMATIONS IN SENTENCE WHICH SUBSUMES THE HEART , LUNGS , AND BONY STRUCTURES
                ARE INTACT , :  TSEQ1 *
TRANSFORMATIONS IN TEXTLET WHICH SUBSUMES THE HEART , LUNGS , AND BONY STRUCTURES ARE
                INTACT , :  TSEQ2 *
TRANSFORMATIONS IN OLD-SENTENCE WHICH SUBSUMES THE HEART , LUNGS , AND BONY
                STRUCTURES ARE INTACT , :  TSEQ4 *
TRANSFORMATIONS IN CENTER WHICH SUBSUMES THE HEART , LUNGS , AND BONY STRUCTURES ARE
                INTACT :  TSEQ5 *
TRANSFORMATIONS IN ASSERTION WHICH SUBSUMES THE HEART , LUNGS , AND BONY STRUCTURES
                 ARE INTACT :
T-EXPAND-TO-CENTER EXECUTED IN ASSERTION WHICH SUBSUMES THE HEART , LUNGS , AND BONY
                STRUCTURES ARE INTACT
    ( EXPAND -> ASSERTION=THE HEART , LUNGS , AND BONY STRUCTURES ARE INTACT ) *
                 IMPLY( ( CONJUNCT ) = ) +
                                      T-MOVE-RC + T-MOVE-NSTG-FRAG + T-FORMAT *
```

Fig. 1 -- LSP analysis showing surface parsing and initial transformations

apply. The result was that after valiant efforts some transformational projects were abandoned; others limited their goals and attempted to supply the needed surface grammar and transformational pinpointing themselves. Later arrivals to computational linguistics sought alternative grammatical frameworks or turned away from grammatical analysis to semantics altogether.

Conditions (a) and (b) are somewhat easier to meet using the original formulation and subsequent development of transformations by Harris. This theory provides an explicit statement of the changes in the sentence produced by each transformation. These changes have their counterpart in the surface parse tree.

Given, then, that reasonable parses can be obtained by enriching the parsing grammar, and that the ambiguity problem can be contained by such devices as the pass-along mechanism described above, it is a straightforward operation to test the parse tree for traces of a particular transformation in order to determine that the corresponding reverse transformation applies. In point of fact, using this approach, it is possible to build most of the information needed to select the correct reverse transformation into the node labels and node attributes of the surface parse tree itself. This makes many reverse transformations exceedingly simple, e.g., relative clauses, it-permutations, object sentence-nominalizations, and even conjunctional expansions, as has been the LSP experience. (But the code for strong nominalizations, as in the *calcium* example above, is extensive.)

```
TRANSFORMATIONS IN LNR WHICH SUBSUMES THE HEART , LUNGS , AND BONY STRUCTURES :

T-EXPAND-TO-CENTER EXECUTED IN LNR WHICH SUBSUMES THE HEART , LUNGS , AND BONY
              STRUCTURES
    ( EXPAND -> LNR=THE HEART ) + IMPLY( ( CONJUNCT  -> LNR=THE LUNGS , ) + ITER(
               ( IMMEDIATE-NODE -> NSTG=THE HEART , THE LUNGS , AND BONY STRUCTURES
                )  + ( EXPAND -> NSTG=THE HEART )  + IS( ( CENTER ) ) - (
          IMMEDIATE-NODE -> SUBJECT=THE HEART , THE LUNGS , AND BONY STRUCTURES )
           + ( EXPAND -> SUBJECT=THE HEART )  + IS( ( CENTER ) )  - (
          IMMEDIATE-NODE -> ASSERTION=THE HEART, THE LUNGS  , AND BONY STRUCTURES ARE
           INTACT )  + ( EXPAND -> ASSERTION=THE HEART ARE INTACT )  + IS ( (
          CENTER )  )  - ( IMMEDIATE-NODE -> CENTER=THE HEART ARE INTACT , THE LUNGS
          , ARE INTACT AND BONY STRUCTURES ARE INTACT )  + ( EXPAND -> CENTER=THE
          HEART ARE INTACT )  + IS( ( CENTER ) ) + ) + ) +
1.  SENTENCE  =  TEXTLET
                 2.

2.  OLD-SENTEN=  INTRODUCER  CENTER  COMMASTG  ENDMARK
                               3.          4

3.  ASSERTION =  SA  SUBJECT  SA  TENSE  SA  VERB  SA  OBJECT  RV  SA
                     5. HEART                ARE       INTACT

4.  Q-CONJ    =  CENTER  ANDSTG
                 6.      AND 7.

5.  LN        =  TPOS  QPOS  APOS  NSPOS  NPOS
                 THE

6.  ASSERTION =  SA  SUBJECT    SA  TENSE  SA  VERB  SA  OBJECT  RV  SA
                     8. LUNGS                  ARE       INTACT

7.  Q-CONJ    =  CENTER
                 9.

8.  LN        =  TPOS  QPOS  APOS  NSPOS  NPOS
                 THE

9.  ASSERTION =  SA  SUBJECT         SA  TENSE  SA  VERB  SA  OBJECT  RV  SA
                     10. STRUCTURES                 ARE       INTACT

10. LN        =  TPOS  QPOS  APOS  NSPOS  NPOS
                             BONY
```

Fig. 2 -- LSP analysis showing conjunction transformation and expanded sentence tree

As an example of a transformation in operation, I have chosen an output which shows the operation of the conjunction-expansion transformation (more exactly: reverse transformation) on a simple sentence from a corpus of medical narrative which has been computer-analyzed by the LSP. Figure 1 shows the surface parse (string analysis) which is passed to the transformational component, and the trace of the first few steps of the transformational decomposition. Figure 2 (a later portion of the same trace) shows the conjunction-expansion transformation being executed and the resulting expanded sentence-tree, which now contains three similar ASSERTIONS each with one noun as subject, in place of one ASSERTION with three conjoined nouns as subject. Conjunction expansion is a crucial ingredient of text regularization, though the decision as to which types of conjuncts should be expanded and up to what level is sometimes application-specific. The code for the expansion transformation shown in Figure 2 is written at an extremely high level of generality (partly indicated by the brevity of the trace) and operates effectively on a wide range of conjunctional constructions, virtually all those encountered in most scientific writing. A similarly general comparative transformation has been written in the LSP framework and one also exists in the IBM REQUEST System (Petrick, 1976; Plath, 1976). Both groups have a number of other transformations in stock or in process.

One should not conclude from the above that all the problems in automatic transformational analysis have been solved. Transformational decomposition on a broad scale over arbitrary texts is still in the future. What has been achieved in the work illustrated above and by others (notably Petrick, Plath, and their colleagues at IBM working on the REQUEST System) is a major step toward that goal. Also, the techniques that have been developed and the set of transformations which have been programmed so far do a useful amount of sentence regularization in themselves. Furthermore, because of their general linguistic basis, they should be applicable in widely different subject matter areas. Since the writing and debugging of transformations is a lengthy process, it is fortunate that a number of applications are possible without a complete transformational analysis (see the section on Applications below). It is also possible to proceed with the expansion of the transformational component and with applications while problems on the discourse level (e.g., reference resolution and intersentence connectives) are being studied.

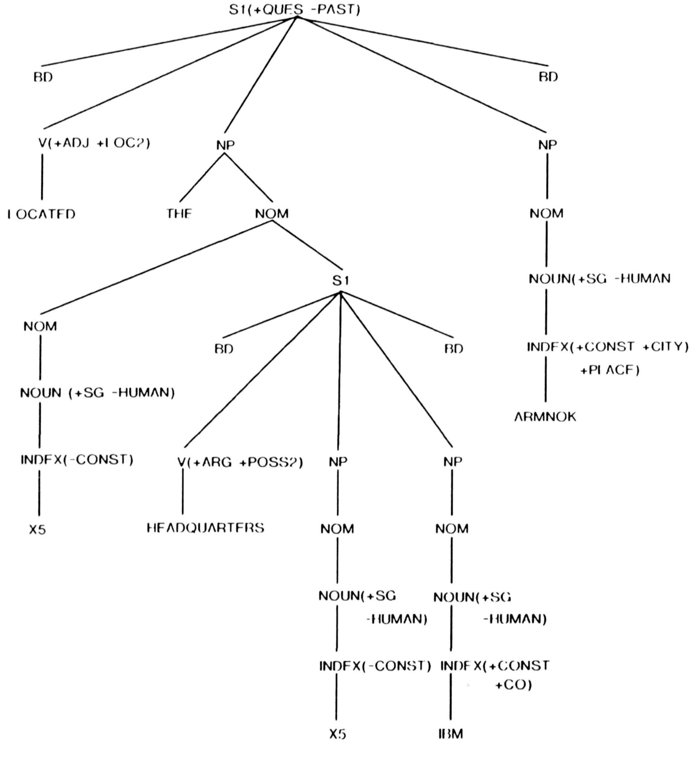


Fig. 3 -- Underlying Structure of "Is IBM's headquarters in Armonk?"

Naomi Sager

Underlying Representations

As noted earlier, the main motivation for using transformations in computational linguistics was the belief that such an analysis would strip the sentence of its grammatical redundancies and provide a structure which was in closer harmony with the content of the sentence. Now that a considerable amount of work has been done, we can ask to what extent this is the case and whether (or what kind of) further analysis is needed. The answer to this question clearly depends on the particular version of *deep structure* that is being considered and also on the particular view that is held of sentence content. In practice, an overriding consideration is what one intends to do with the analysis.

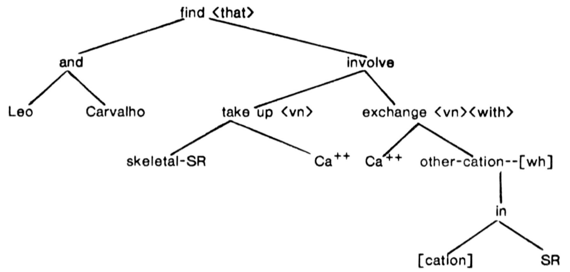
With regard to the different versions of deep structure, I intend here to bypass what most linguists (and many others) would consider the critical issue: the *choice* of the proper underlying structure. The reason for this, crudely stated, is that I do not believe this is an area in which we have much freedom of choice. Rather, we are given language data and we try to develop methods for treating these data; the structures we arrive at are the ones which result from applying the methods, and which can be shown to exist in a broad coverage of the language data. This is, of course, a simplification of the problem since we have some choice over the methods of analysis and these are intertwined with the theory of language structure which is used. Nevertheless, I believe this a relevant computational view, at least where applications are intended. The diverse structures which have been proposed for representing the underlying content of sentences become appropriate for consideration in an applicational context to the extent that they are accompanied by methods (including here linguistic content) which can be shown to apply to a reasonable corpus of non-contrived sentences. On this criterion, many of the interesting semantic theories that have been proposed are not at a stage where they can be considered as models for applications.

With regard to the two major transformational theories, transformational generative grammar, as incorporated into the IBM REQUEST system (a system for answering queries from a structured data base of business statistics), produces deep structures of the type illustrated in Figure 3 (from Plath, 1976, p. 329*). The underlying structures are trees consisting of one or more nested propositions (S1) each marked off by a pair of boundary symbols (BD). Each proposition consists of a predicate (V) followed by its associated argument (NP's), which always occur in a fixed order in the structure. Plath points out that such surface structure elements as auxiliaries, prepositions, inflectional endings, and punctuations -- all major sources of syntactic variation -- have been eliminated in favor of binary syntactic features (e.g., (-PAST), (+POSS2), (+SG), and (+QUES)) occurring at standard positions in the tree. He goes on to point out that a key property of these underlying structures is their close resemblance to expressions in the predicate calculus, e.g., as illustrated in Figure 3, The representation of surface proper nouns (*IBM, Armonk*) as logical constants (INDEX (+CONST)) and the treatment of surface common nouns such as *headquarters* as variables (INDEX (-CONST)) in propositional functions. Following the transformational stage, the REQUEST system translates structures of this type into an underlying representation in *logical form*, which is useful in the question-answering operation proper. Woods' LSNLIS system for answering questions from a structured data base concerning lunar rocks also mapped into an underlying representation based on the predicate calculus (Woods et al., 1972).

----

*Copyright 1976 by the International Business Machines Corporation; used by permission.

Carvalho and Leo found that the Ca^{++} uptake of skeletal SR involves the exchange of CA^{++} with other cations in SR. (LE711 13C.5.7)



Linearization of example:

```
(find <that>
 (and (Leo) (Carvalho))
 (involve
   (take up <vn>
      (SR-(skeletal)) (Ca$^{++}$))
   (exchange <vn> <with>
      (Ca$^{++}$)(cation-(other) -- ([wh](in([cation])(SR))))))).
```

Fig. 4 -- Underlying representation for LSP analysis

In this connection it is worth noting that currently interest is growing in the data base management field in alternatives to the predicate calculus as a representation for structured data bases in certain applications. In particular, the relational model of data introduced by Codd (1970) has stimulated the development of at least one data base query language (SQUARE), which the authors suggest "mimics how people use relations or tables to obtain information" and does not require the mathematical machinery of the predicate calculus in order to express simple references to tables (see Boyce et al., 1975a,b). Natural language processing front ends have been suggested for these systems. These developments are of interest both to computational linguistics and information science.

The use of Harris' transformational theory (Harris, 1968), as adapted for text processing by the LSP, results in underlying representations of the type illustrated in Figure 4 (from Hirschman *et al.*, 1975, pp. 42-43).* In Figure 4, the form is a dependency tree where each node corresponds to a word or base-form of a word; if the word is a base form of a verb, or is by another criterion an *operator* in the sense of Harris, its arguments are shown as daughter nodes. For some purposes a linearization of the tree is desired; this also is shown.

This type of analysis has several interesting properties from an informational point of view. First of all, it suggests that information, at least scientific information as it is carried by language, can be resolved in a hierarchy of predications. In the example sentence, what is predicated about Ca^{++} and other cations is that they exchange with each other, and what is predicated about skeletal SR (sarcoplasmic reticulum) and Ca^{++} is that the one takes up the other. This is from reading the bottom-most operators with their operands. Next, about the taking up and the exchange, it is predicated that the one involves the other. On the highest level of the hierarchy it is predicated that there is a relation between the human investigators Leo and Carvalho and the hierarchy of relations dominated by *involve*, and that this relation is *find.*

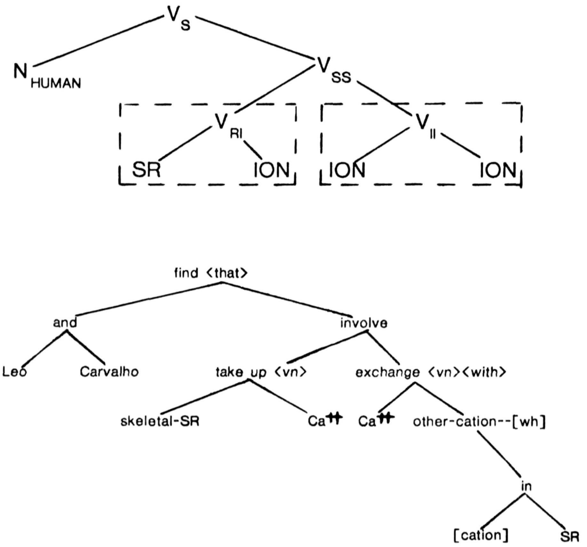
When this type of analysis is made over several hundred science sentences it can further be seen that certain classes of words characteristically are found at the bottom of the hierarchy and others near the top (when both classes are present in the sentence). The concrete words referring to objects in the science are at the bottom, and the intellectual verbs, like *find, hypothesize, discover,* which characteristically take human subjects, appear at the top; thus, the analysis effectively separates the *object language* of the science from the *meta language* -- that is, it separates the words referring to the objects and relations in the science proper from statements the human investigator makes about those objects and relations. On levels in between, one finds definite classes, like quantity operators and causality operators, each in characteristic positions *vis a vis* other operators. These classes and their positions in the hierarchy appear to be common in much of scientific writing. All of this is highly suggestive of a structure to the information which is being transmitted -- a structure not identical with that of the sentences, but certainly in close relation to the linguistic frame.

Can this *deep structure* itself serve as the ultimate representation of sentences? For certain statistical processes, such as clustering to obtain semantic classes, this representation yields good results without further transforming. Thus, a clustering program to be described later successfully obtained a promising set of semantic classes for a subfield of pharmacology when it was applied to a sample corpus of linearized transformative trees of the type shown in Figure 4.

For other purposes, such as question answering, fact retrieval and data processing, this representation, while undoubtedly capturing the overall fabric of the relations involved, lacks the detailed classification of operators and objects which would permit specific pinpointing of content or the retrieval of particular subsets across the hierarchies. It also lacks the simplicity and the logical structure which is sought in most data base applications. One way of arriving at structures that reflect specific textual content is to make use of the restrictions in language usage which are characteristic of the texts in a particular subject matter: that is, to exploit the fact that on a particular topic, only certain words in certain combinations actually appear. Distributional linguistic techniques applied to transformationally analyzed texts in a given subfield yield patterns of word-class cooccurrence. These patterns can be seen as selectional specializations of the transformational dependency trees. Instead of seeing the words at the nodes as members of the large grammatical classes *noun, verb,* etc., we can view them as members of the subfield classes (e.g., in the pharmacology data: the drug class, the ion class, etc.). We find that only certain combinations of word-class nodes occur. Each pattern can be represented by a transformational tree whose nodes are relabeled in accord with the subfield word classes which cooccur in that particular configuration. The interesting point is that a small set of patterns (relabeled trees) for a body of texts reflects specifically what is being talked about most in the texts. The relabeled trees are thus a type of format for the information in a great many sentences and sentence-portions dealing with the same subject matter.

----

*Reproduced with permission from Information Processing and Management, copyright 1975, Pergamon Press, Ltd. The transformational decomposition operates on the string parse tree and converts it to a tree which is composed of simple ASSERTIONS connected via nodes specifying the transformations which operated (Hobbs et al., 1976). A simple program converts outputs of this type to the type shown in Figure 4.



Vs : verb class with sentential object, human subject (e. g. *find, think, assume*)

V_{ss} : verb class with sentential subject and object, often causal in nature (e. g. *influence, affect*)

S_e : elementary sentence format, specific to the subfield; here involving:

V_{RI} : verb class connecting nouns of the SR class with those of the ION class (e. g. *take up, contain*)

V_{II} : verb class connecting nouns of the ION class to nouns of the ION class (e. g. *exchange, compete*)

Fig. 5 -- Subfield format in tree form

To illustrate, a sentence format of the type found to be characteristic of the pharmacology subfield referred to above is shown in Figure 5, along with a repeat of the sentence of Figure 4, which is an instance of this format. The lower level subformats S_e cover the occurrences of elementary (object language) sentences. In the pharmacology material, two of these were of the types SR V_{RI} ION, ION V_{II} ION, appearing in Figure 5 under V_{SS}. Of course, there are fairly numerous bottom level subformats for a subfield, and they appear

N_0 Q	V_{SS} . . . V_{SS}	V_Q	N_1 Q V Q (P)N_2 PN_3	D_S	CONJ

N_0 Q	V_{SS} . . . V_{SS}	V_Q	N_1 Q V Q (P)N_2 PN_3	D_S	CONJ
DRUG ACTION	DRUG ACTION	QUANTITATIVE CHANGE	ELEMENTARY SENTENCE OF CELL PHYSIOLOGY, BIOCHEMISTRY,...	EXPERIMENTAL AND CLINICAL CONDITIONS	SENTENCE CONNECTIVE

Fig. 6 -- Partial text format for pharmacological subfield

in different combinations under V_{SS} (causal) verbs. This reflects the fact that scientific reporting is concerned with establishing causal connections between events. A feature not illustrated in Figure 5 is the fact that, in this science at least, quantitative operators play a significant role. These characteristically occur between the elementary sentence level and the causal (V_{SS}) level. In contrast, time operators did not play a role in this material.

A summary of the features of the pharmacology subfield formats for the factual portions of sentences (i.e., cut off at V_S) obtained for the large bulk of the sentences in a 300 page corpus is shown in Figure 6 (from Sager, 1975, p. 15).* Here, to illustrate the repetition of format occurrences in texts, each subformat S_e with its immediate operators up to the introduction of a drug word, if present, is shown *flattened*, filling one line. Connectives between format lines may be conjunctions, but alternatively V_{SS} verbs, as in Figure 5. In a text occurrence, the innermost section (between double bars) contains a verb with its concrete noun subject and object (s), i.e., a bottom level operator in the transformational tree, along with its arguments (e.g., *Ca^{++} exchanges with other cations*); and the successive boxes to the left represent the hierarchy of verbal operators on it.

Linguistically-based subfield formats are one answer to the question of underlying representation. While they are based on selectional constraints that operate in particular science subfields, they have certain features which may be common to many science fields. Just as one does not arrive at a universal grammar by studying all languages at once, it may be that in attempting to represent the semantic content of texts, we will be obliged to start with the regularities that obtain in demarcated domains of discourse. At the same time, obtaining the special forms which can be established for one subfield at a time is certainly not the only way to utilize the informational connections which emerge from transformational analysis. We may also consider each triple consisting of an operator with its two arguments (the most frequent case) as a unit. The operator, usually a verb, can be viewed as filling the role of F in the functional notation F (X)=Y.**

----

*Reproduced with and by permission of the Journal of the American Society for Information Science, Volume 26, No. 1, Jan.-Feb. 1975. Copyright 1975 by the American Society for Information Science, 1155 16th Street N.W., Washington, D. C. 20036.

**For the moment, in this adaptation of transformations, we do not count among the F operators the connectives concerned with adjunction and conjunction because the deformed sentences under such connectives can be expanded to full subtrees, and the connection established by *and*; *wh-* is too general for pratical use.

X and Y may be concrete nouns, or themselves *F*s with their own arguments. The words which fill an *F* role on similar argument types can be grouped together and assigned a semantic label. Irwin Bross and associates at Roswell Park Memorial Hospital used such a final representation for linguistically analyzed surgical reports (Bross and Stermole, 1973). This may also be a useful form for retrieval, or for arriving at coordinated index terms from deep structures, a possibility raised by Sparck Jones and Kay to which we will return in the next section. It should also be noted that the F(X) = Y notation has something in common with the relational model of data, mentioned above.

Applications

Computerized language processing at its most successful will not replace existing methods of information retrieval, but may supply tools for improving present services and, perhaps more significantly, open the way to new services which are only conceivable in the wake of advanced linguistic techniques.

Generating Word Lists. Under the rubric of tools would be new approaches to thesaurus generation based on linguistic analysis of texts. The experimental results obtained in generating word sets from transformationally analyzed pharmacology sentences are quite encouraging in this respect. This experiment, referred to earlier, demonstrated that a statistical treatment of linguistic tokens could produce semantically sharp and informationally relevant word classes when the clustering took into account the grammatical relations of words in the sentences. In particular, the sentences in the experiment were all from texts treating the mechanisms of action of digitalis; *Methods* sections were excluded but the sentences were not otherwise especially selected. Two words were considered *similar* if they occurred in the same argument position under the same operator (e.g., as subject of a certain verb) or as operator on the same operand (e.g., as verb with the same noun object) in the transformational decomposition of a sentence. Similarity coefficients SC were computed for all possible pairs of words, and clusters were built up one word at a time; a word was added to a cluster if the average of its SC with each other word in the cluster exceeded a threshold value. Clusters with 2/3 overlap were merged to produce the final word classes. These classes turned out to be semantically coherent, and the set of word classes as a whole included the main types recognized by a cardiologist for the same material.

To check the coverage obtained by the computerized procedure, all the words in the corpus were manually classified and the resulting classes were compared. Detailed tables of this work have been published, of which one figure is reproduced here to illustrate the type of results obtained. Figure 7 (from Hirschman *et. al.*, 1975, p. 50*) shows the computer class of drug words (cardiac glycosides CG) compared with the manually obtained class. The computer class contains all the high frequency words which appeared in the manual class and only one other word. The low frequency terms which were missed might be captured in a larger corpus. (It was surprising how much was obtained with such a small set of sentences, ca. 400.). Or, once the important classes were established, they could be supplemented by manually examining word lists or consulting an expert.

The algorithm clearly works well in particular subject matter areas where not only do the texts have overlapping vocabularies, but usage has become routinized. Whether similar results can be obtained over broader subject matter areas remains to be investigated. It

----

*Reproduced with permission from Information Processing and Management, Copyright 1975, Pergamon Press, Ltd.

should also be noted that the input trees in this experiment were not obtained by computerized decomposition of sentences, though only standard English transformations which were in the process of being programmed were used. The use of such clustering algorithms on an extensive scale would of course depend on obtaining the input from computer sources, and one should not minimize the problems of real text processing. The very complexity of most science sentences even if they are not ambiguous is forbidding. In addition, the problem of ambiguity is real. In applying the clustering program to computer outputs of text decompositions, it would be necessary to select an initial set of unambiguous outputs (selected by hand or by skipping over sentences with multiple outputs) and to use them in a bootstrap operation to obtain others. Presumably, the feeding back of selectional constraints based on results from the initial set of outputs would aid in disambiguating a number of the remaining outputs. However, this step may not be crucial. The amount of repetition of significant operator-argument pairs is so great

CG CLASS		No. OCCURRENCES	
COMPUTER	MANUAL	IN PAIRS†	
CG	CG	156	$\frac{395}{442}$=89%
digitalis	digitalis	118	
ouabain	ouabain	70	
drug	drug	15	
agent	agent	8	
strophanthidin	strophanthidin	5	
strophanthidin 3 bromoacetate	strophanthidin 3 bromoacetate	4	
strophanthin	strophanthin	4	
cardiotonic glycoside	cardiotonic glycoside	3	
compound	compound	7	
inhibitor	inhibitor	5	
*erythrophleum alkaloid		*6	
	glycoside	11	
	digoxin	7	
	acetyl strophanthidin	7	
	cardioactive glycoside	6	
	digitalis glycoside	6	
	digitoxigenin	3	
	strophanthoside	2	
	cardiac glycoside	2	
	digitoxin	1	
	digitalis compound	1	
	strophanthin K	1	
		442	

**Erythrophleum alkaloid* does not belong in the GC class; it is a drug whose affect is compared to that of the cardiac glycosides.

Agent, *drug*, and *compaound* are classifiers for words in the GC class, as well as of the more general DRUG class. *Inhibitor* is also a classifier, which classifies according to function.

†An occurrence of a word either as the operator or operand in a pair. Pair occurrences are more numerous than text occurrences for several reasons. Recoverably zeroed material is reconstructed and contributes to pair formation. Also each operator can appear in a pair as the operand, as well as with each one of its arguments. (Thus, a two-argument verb can appear in three pairs.) For concrete nouns, however, this does not occur, and the pair occurrences correlate more closely with the number of actual occurrences in the text.

Fig. 7 -- Grammatically-based automatic word class formation

that even if only a small percentage of the text sentences can be used for clustering, the results are still likely to be creditable.

Generating Text Descriptors. In the class of tools, or adjuncts to retrieval, is the interesting possibility raised by Sparck Jones and Kay concerning the automatic generation of structured index descriptions. In the concluding paragraph of the chapter on Syntax in Information Retrieval (1973, p. 119) they write:

> *. . . it is intriguing to imagine a system in which document descriptions using links and roles would be obtained automatically from deep structures produced by an advanced transformational analysis technique. This would be a difficult enterprise, but there is no doubt that it would blend theories and methods of interest in both linguistics and information science and the results could hardly fail to be instructive.*

To put this hypothesis to an immediate informal test, a small experiment was run on the pharmacology corpus for which we had both the clustered word classes and the linearized transformational trees. All triples consisting of an operator and its arguments were generated. Every word which was a member of one of the three main noun classes DRUG, ENZYME, ION* was replaced by the name of the class to which it belonged. Triple occurrences were then counted and all those with a frequency of 5 or more were printed.

The printed list was then culled for all triples whose arguments were nouns. Note: nominalized verbs like *uptake*, etc. appear as verbs, not nouns, in the decompositions.) The results are shown in Figure 8.

It is, of course, hard to judge the output of Figure 8 as a representation for a set of texts which are not available to the reader. It is the case that these articles (and others in the field) were primarily concerned at this time (mainly late 1960's) with the possible effects of digitalis on ion transport and on the activity of the enzyme ATPase (in particular Na^+-K^+-activated ATPase) as possible explanations for the ultimate effect of digitalis in enhancing the contractility of heart muscle. There is much discussion in the texts of ion movements in the cell, and the beginning of the discussion (which becomes more prominent in later papers) of events at the level of the contractile proteins, in particular those involving Ca^{++} movements in relation to actin and myosin. The triples are, then, a fair quick summary of what the articles deal with (though, of course, not of what they say).

Some additional comments are in order. Note that the f/f' ratios of the first group are much lower than those in the other groups. This is because the verbs in the first group are mainly sentence-connecting verbs rather than noun-connecting verbs, i.e., they carry the posited connection between *events*. These connections are not caught by triples because a verbal argument without its own arguments gives little information. For example, the computer output contained the group of triples:

----

* Pharmacology Noun classes for run of 3/10/76:
DRUG CLASS: STROPHANTHIDIN 3 BRO, DRUG, DIGITALIS, STROPHANTHIDIN, OUABAIN, AGENT, GLYCOSIDE, CG, ERYTHROPHLEUM ALKALO, QUINIDINE, CHLORPROMAZINE, STROPHANTHIN, INHIBITOR, COMPOUND;
ENZYME CLASS: NA^+ K^+ ATPASE, ENZYME, ATPASE;
ION CLASS: K^+, NA^+, ION, SODIUM, CALCIUM, POTASSIUM, GLUCOSE, CA^{++}, K, ELECTROLYTE, NA, CA.

ARG1	OP	ARG2	f/f'*
DRUG	affect	ENZYME	26/276
	inhibit		16/95
	stimulate		7/52
	bind		6/46
			55/469
ION	activate	ENZYME	8/17
ION	concentrate	cell :	15/121
	intra		16/20
	level (has)		16/12
	move		16/20
	concentrate	tissue	6/121
	flow	muscle	7/51
			56/114
cell	contain	ION	19/50
myocardium	contain		10/50
	lose		5/28
SR	find		9/46
	take (up)		23/41
			66/165
actin	interact	mocin	7/14
fiber	extract	glycerol	6/9
fiber	contract		6/105
muscle			6/105
protein			9/105
			21/105

Operator-argument triples are shown here for noun arguments only and for triples occurring with frequency f ≥ 5. Capitalized words stand for noun classes. The table includes pair due to 1-argument verbs (e. g. *contract*).

*f' = total no. of occurrences of operator in the corpus.

Fig. 8 -- Pharmacology Operator-Argument triples

which is really the first part of a larger h-tuple linking this group with some of the triples in other groups, e.g., combining a line of the above group with one from the fourth group in Figure 8 would give the quintuple:

DRUG affect SR take(up) ION

This corresponds to sentences such as *Digitalis may have an indirect effect on the Ca^{++} uptake of skeletal SR*, or sentence parts like *the effect of the cardiac glyocosides on the uptake of calcium by the sarcoplasmic reticulum*. We see that this quintuple begins to resemble one of the information formats found in this material (Figures 6, 7). There is even hope that with a large enough corpus the computer programs could discover such patterns in the textual material.

It appears, then, that triples consisting of an operator and its noun argument(s), obtained from transformational decomposition trees of text sentences, are a promising source of structured index terms. In addition, it may even be possible to recognize larger patterns as a step toward developing data structures for use in fact retrieval and in processing the textual content.

Structuring Natural Language Data Bases. What we cannot do quite yet in the case of the journal literature, namely transform the natural language form of the data into structures suitable for computerized information processing, appears to be more immediately feasible in natural language data bases of a more restricted kind, such as are found in reports and records, e.g., weather reports, medical records of various types, program specifications, special purpose files in business and industry as well as in science. Several factors conspire to make this one of the most promising areas currently for applying computational linguistic techniques, chiefly:

(a) From the CL side, the language problems are easier to handle: in such material, the sentences (or sentence fragments) are on the whole shorter than in full texts; sentence connectives are fewer and simpler; pronominal reference is limited; vocabulary is limited, and most important, it is used in a regular unambiguous way.

(b) From the user side, the need for such natural language processing is large. Files and records are growing, and while many of them can be assigned an overall format, the entries under the structured headings are for the most part unavoidably in natural language. At the same time many offices, hospitals, services, etc. are computerizing their files for quick input-output or other transactional purposes, so that the natural language data store is available for processing, if the programs for such processing can be devised.

The combination of the developing need, the developing natural language data bases, and the developing language techniques is an auspicious setting for a cooperative effort at developing new services. Sparck Jones and Kay suggest in their conclusion (1973, p. 198) that in the task of relating linguistics and documentation more effectively, "the onus is on the documentalist to provide a proper specification of what he needs to satisfy his retrieval objectives". While clear specifications are certainly needed, it is still possible for the impetus to come from the computational linguist, by demonstrating what can be done and working with the documentalist (or information specialist in a given area) to specify new objectives. The area of natural language data bases is a good place to start since the user and data specialist are close at hand -- sometimes even the same individual, and cooperation is thus easier to achieve.

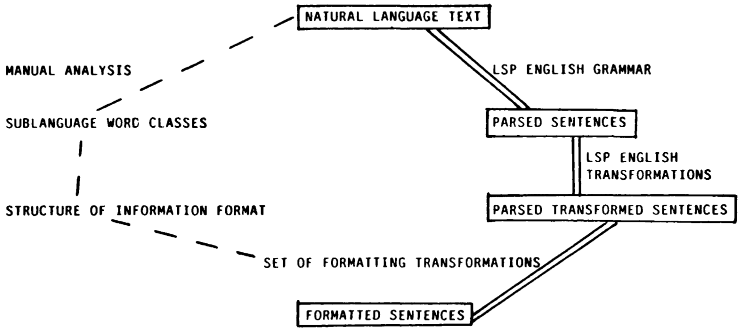

NATURAL LANGUAGE TEXT
MANUAL ANALYSIS
LSP ENGLISH GRAMMAR
SUBLANGUAGE WORC CLASSES
PARSED SENTENCES
LSP ENGLISH TRANSFORMATIONS
STRUCTURE OF INFORMATION FORMAT
PARSED TRANSFORMED SENTENCES
SET OF FORMATTING TRANSFORMATIONS
FORMATTED SENTENCES

Fig. 9 -- Flowchart Showing Processing in Medical Record Analysis

Work on the computerized structuring of information in natural language records is just beginning. Some work on weather reports is being done in Canada in the framework of mechanical translation (Chandioux, 1976). The example provided below is from work on medical records, in particular a small corpus of X-ray reports on patients who were being followed after cancer surgery. The material ranged from simple one or two word entries like *none, not done* to textlets like the following:

> CHEST X-RAY ON 10-14-69 SHOWED MARKED REACTIONS SUPERIORLY OF THE LEFT HILUM WITH CONSIDERABLE PLEURAL REACTION AND/OR LOCULATED FLUID IN THIS AREA IS SEEN. THIS, HOWEVER, HAS REMAINED UNCHANGED OVER THE PAST FOUR MONTHS. THE HEART REMAINS NORMAL IN SIZE AND CONFIGURATION. NO EVIDENCE FOR PULMONARY OR BONY METASTASIS.

The project proceeded according to the flow chart in Figure 9. The labeled broken lines represent manual steps to obtain the appropriate word classes and formats (the clustering program was not used in this experiment), and the double dark lines represent steps in computer processing, equipped as indicated with a grammar and transformations. A rather detailed information format was developed, shown with one instance of a formatted entry in Figure 10. The computer parse and computer format for this entry are appended in Figures 11a, 11b, and 12. The format has two main parts, roughly corresponding to the subject and verb phrase of a full entry sentence, where the subject describes the TEST and the verb phrase describes the FINDING. Each part is further divided. In this example, the TEST part of the format has 3 filled slots: TESTNoun (flat plate), TESTLOCation (of the abdomen), and TESTDATE (3/28/68). Since the verb phrase contained a conjunction, the sentence was expanded into two sentences by the conjunction expansion transformation, resulting in two formats having the same TEST part but different FINDING parts. The first FINDING contains a CHANGE entry (mild degenerative changes); the second FINDING contains an entry in the MEDical FINDings slot (osteoporosis, recognized by the fact that osteoporosis is in a subfield word-class), and entries in REGION slots (of the sacral spine, also recognized by the occurrence of words in subfield word-classes).

This is not the place to discuss the formatting experiment, which will be presented elsewhere (Hirschman et al., 1976). However a few points are worth noting with regard to how this work differs from other uses of natural language processing. In clustering or

TEST					FINDING									CONN
NO-TEST	TESTN	TESTLOC	VERB-DONE	TESTDATE	NEG	VERB-ELEMENT		CHANGE	STATUS	MED-FIND	REGION			
						BE-SHOW	INDIC.				POS	PT-BODY	STRUC	
	(FLAT) PLATE	(OF THE) ABDOMEN		3-28-68		—		(MILD DEGENERATIVE CHANGES						AND
	(FLAT) PLATE	(OF THE) ABDOMEN		3-28-68		—				OSTEOPORSIS	OF	SPINE		

Adjuncts are written in ( ); left adjuncts *above* the entry, right adjuncts *below* the entry.

Fig. 10 -- Format of BROSS 1.1.2

indexing, for example, the correct assigning of a negative element does not affect the result, but here it is crucial. *No data* is very different from *No evidence of metastasis.* The formatting therefore requires both a careful syntactic analysis and detailed subclassifications of the subfield words.

Of what use are such data structures? Properly organized within the computer (perhaps in more conventional data base formalism), they would enable specific queries to be answered from the data, and statistical operations to be performed. In the health field such formatting might eventually be used to study the frequency and severity of recurring symptoms in certain diseases, as reflected in the patients' records, and to monitor on-going health care in clinics.

Aiding the User. Subscribers to retrieval services often have mixed reactions; they are glad to have the service because they need it and there is no alternative way to search the large literature store; at the same time they wish there was some way to *tune* the search more specifically to their needs. It has been suggested by many that what is needed is a facility whereby the user can direct the search himself. Similar remarks have been made about other uses of computers, such as tailoring or designing software packages for particular users. The naturally occurring image is that of the user before the console, interacting with the system and directing it (or being directed by it) by means of a dialogue carried out in natural language. This image is not universally applicable, however. Some users want no contact with computers; they just want better service. Both avenues are presently leading toward more emphasis on natural language processing.

Taking up first the matter of improving service, there is a distinct possibility that a natural language processor capable of handling scientific English could be used in connection with existing retrieval systems to increase precision. Syntactic relations can be checked, and some retrieval errors, such as *false coordinations* might be caught this way. If this technique were perfected (and made more efficient), recall could be relaxed so that fewer relevant documents would be missed, and the processor would then be used to narrow down the set of retrieved documents to meet high precision standards. A caution here is a point made earlier: the problem of ambiguity in text processing remains. There is still the possibility that a great many selectional constraints are needed in order for parsing (or transformational analysis) to be feasible, or useful. And selectional constraints are precisely what is lacking in the language corpus of a diverse collection. On the other hand, queries and profiles can be sorted as to subject matter, and a *library* of selectional constraints slowly accumulated for the different subject areas. The view here is that the user could then be indirectly in natural language communication with the information sources, by virtue of a system which confronts the user's request with (some of) the natural language material of the retrieved documents. Limited experiments of this type could be set up using existing means.

```
* BROSS  1. 1. 2
* LINEARIZED PARSE TREE

SENTENCE
↓
TEXTLET
↓
OLD-SENTENCE---MORESENT
↓
INTRODUCER---CENTER-----ENDMARK
             ↓          ↓
             FRAGMENT   # . #
             ↓          ↓
             ↓          .
             ↓
             SA------SOBESHOW
             ↓       ↓
             PDATE   SUBJECT---BE-OR-SHOW---OBJBE
             ↓       ↓         ↓            ↓
             ↓       NSTG      #--#         NSTG
             ↓       ↓         ↓            ↓
             ↓       LNR       --           * 1 *
             ↓       LN-----------------------------------------NVAR---RN
             ↓       ↓                                          ↓      ↓
             ↓       TPOS---QPOS----------APOS---NSPOS---NPOS   N      RNP
             ↓                            ↓                     ↓      ↓
             ↓                            ↓                     ↓      PN
             ↓                            ↓                     ↓      ↓
             ↓                            ↓                     ↓      LP---P-----NSTGO
             ↓                            ↓                     ↓           ↓     ↓
             DATEPREP---DATE---MOREDATE   ADJADJ                PLATE       OF    NSTG
                        ↓                 ↓                                       ↓
                        3-28-68           LAR1                                    LNR
                                          ↓                                       ↓
                                          LA---AVAR---RA1                         LN----------------------------------NVAR---RN
                                               ↓                                  ↓                                   ↓
                                               LCDA---ADJ                         TPOS---QPOS---APOS---NSPOS---NPOS   N
                                                      ↓                           ↓                                   ↓
                                                      FLAT                        LTR                                 ABDOMEN
                                                                                  ↓
                                                                                  LT---T
                                                                                       ↓
                                                                                       THE
```

Fig. 11a -- Linearized Parse Tree

```
* BROSS  1. 1. 2
*

* 1 *
↓
LNR
↓
LN----------------------------------NVAR---RN---ANDSTG
↓                                   ↓           ↓
TPOS---QPOS---APOS---NSPOS---NPOS   N           #AND#---SACONJ---Q-CONJ
              ↓                     ↓           ↓                ↓
              ADJADJ                CHANGES     AND              LN---NVAR---RN
              ↓                                                       ↓      ↓
              ADJADJ---LAR1                                           N      RNP
              ↓        ↓                                              ↓      ↓
              ↓        ↓                                              ↓      PN
              ↓        ↓                                              ↓      ↓
              ↓        ↓                                              ↓      LP-------P-----NSTGO
              ↓        ↓                                              ↓               ↓     ↓
              LAR1     LA-------AVAR---RA1                            OSTEOPOROSIS    OF    NSTG
              ↓                 ↓                                                           ↓
              LA---AVAR---RA1   LCDA---ADJ                                                  LNR
                   ↓                   ↓                                                    ↓
                   LCDA---ADJ          DEGENERATIVE                                         LN----------------------------------NVAR---RN
                          ↓                                                                 ↓                                   ↓
                          MILD                                                              TPOS---QPOS---APOS---NSPOS---NPOS   N
                                                                                            ↓             ↓                     ↓
                                                                                            LTR           ADJADJ                SPINE
                                                                                            ↓             ↓
                                                                                            LT---T        LAR1
                                                                                                 ↓        ↓
                                                                                                 THE      LA---AVAR---RA1
                                                                                                               ↓
                                                                                                               LCDA---ADJ
                                                                                                                      ↓
                                                                                                                      SACRAL
```

Fig. 11b -- Linearized Parse Tree

```
BROSS 1.1.2  3-28-68 FLAT PLATE OF ABDOMEN -- MILD DEGENERATIVE CHANGES AND
             OSTEOPOROSIS OF THE SACRAL SPINE .

FORMAT A
↑
DATA
↑
TEST----------------------------------------FINDING
↑
TESTN----------TESTLOC----------TESTDATE    VERB------CHANGE-OVER-TIME
↑              ↑                ↑           ↑         ↑
PLATE---LADJ   ABDOMEN---LADJ   3-28-68     BE-SHOW   CHANGE
        ↑                ↑                  ↑         ↑
        FLAT             OF THE             --        CHANGES---LADJ
                                                                ↑
                                                                MILD DEGENERATIVE

FORMAT B
↑
DATA
↑
TEST----------------------------------------FINDING
↑
TESTN----------TESTLOC----------TESTDATE    VERB------MED-FINDING----REGION
↑              ↑                ↑           ↑         ↑              ↑
PLATE---LADJ   ABDOMEN---LADJ   3-28-68     BE-SHOW   OSTEOPOROSIS   POSITION---PT-BODY
        ↑                ↑                  ↑                        ↑          ↑
        FLAT             OF THE             --                       OF         SPINE---LADJ
                                                                                        ↑
                                                                                        THE SACRAL
```

FIG. 12 -- Computer Format for Example BROSS 1.1.2

One of the attractive aspects (at least at the outset) of the approach based on user interactive systems, is the dynamic features. Not only can the user see right away whether he is getting what he wants and re-direct the system if necessary (as in the LEADERMART retrieval system developed by Hillman, 1969), but even language ambiguity problems could be resolved by the user. In practice, the user is not going to want to look at sentence analysis trees or the equivalent, and answer questions about them -- at least, not about very many. So, a high quality language analyzer is a prerequisite for this avenue also. The interactive approach has a strong appeal because it allows the user to modify the request in response to what is available, but much remains to be understood about natural language interaction (discourse) before this solution will work well. There is interesting work in the area of artificial intelligence on discourse problems and language understanding. Some of this work may yet bear on practical problems of providing interactive *front ends* for information retrieval and information processing systems.

A few concluding remarks, on the important topic of processing times. Figure 13 shows a table of sample times for processing relatively simple sentence types. The top two-thirds of the table are a comparison made by Petrick (1976) of published LSNLIS times with times he obtained for comparable sentences. Taking account of the differences in machine, time-sharing system, etc. (the IBM LISP system is faster), he concludes that a normalization would yield roughly comparable times for the two systems. The bottom section of the table shows some figures from LSP runs on sentences and sentence fragments of medium complexity. The times given in Figure 13 should not be used to rate the efficiency of the different systems but to gain an impression of the processing times that are currently possible. One sees that short sentences are processed in from 1-20 seconds. Long sentences from scientific articles, such as are parsed by the LSP, may run to over one minute. One should keep in mind, also, that research groups are using laboratory, not production, models.

If the times given here are representative of the current state of processing, the situation is not discouraging. Serious experiments in natural language information processing using existing processors are feasible, and it is not unreasonable to think of limited practical applications in the areas of question-answering from structured data bases, and automatic structuring of specialized natural language data bases. Mass processing of texts would not yet be economical if, indeed, it proves possible. However, we still have basic work to do on language processing, and in that time computer technology may catch up with our needs. To reverse a metaphor, one might say that if we can beat a path to an effective natural language analyzer, the world will build the better mousetrap (read ultra-fast special purpose computer) and lay it at our door.

LSNLIS		TIMES	
Sentence	Parsing	Interpretation	Total
(1) How many lunar samples are there?	2.039	5.152	7.191
(2) How many breccias do not contain Europium?	6.272	8.593	14.865
(3) How many samples contain chromite?	3.579	8.277	11.856
(4) Which rocks contain chromite and ulvospinal?	7.743	9.782	17.525
REQUEST			
Sentence			
(1) How many profitable companies are there?	3.868	0.163	4.031
(2) How many companies do not produce gas?	3.369	0.238	3.607
(3) How many companies sell computers?	3.226	0.239	3.465
(4) Which people sold IBM and Xerox?	3.046	0.189	3.235

LSP		TIMES	
Sentence or Sentence Fragment*	Parsing	Formatting	Total
(1) 5-2-67 chest--no change since 2-7-67	1.19	2.89	4.08
(2) Chest X-ray 3-3-70 negative.	2.04	2.09	4.13
(3) 6-6 6-7-68 chest film 7-30 shows no improvement	1.07	3.58	4.65
(4) 3-28-68 flat plat of the abdomen--mild degenerative changes and osteoporosis of the sacral spine	6.52	8.38	14.90

Fig. 13 -- A Few Examples of Sentence Processing Times

*The numbers in these sentences are dates, and are analyzed as such by the system.

References

Boyce, R. F., et al. "Specifying Queries as Relational Expressions" in Proceedings of the ACM SIGPLAN/SIGIR Meeting. *ACM SIGPLAN Notices*, 1975, 10(1). [Also in *ACM SIGIR Forum* 1974, 9(3).]

Boyce, R. F., Chamberlin, D. D. King, W.F., III, and Hammer, M.M. "Specifying Queries as Relational Expressions: The Square Data Sublanguage." *Communications of the ACM*, 1975, 18, 621-628.

Bross, I. D. J., and Stermole, D. F. "Computer-Assisted Discourse Analysis of a Jargon." *Computer Studies in the Humanities and Verbal Behavior*, IV, 2, 65-76 (1973).

Chandioux, J. "METEO: An Operational System for the Translation of Weather Forecasts." In Hays, D.G. and Mathias, J., eds., *FBIS Seminar on Machine Translation*. American Journal of Computational Linguistics, 1976, 2, Microfiche 46, 27-36.

Chomsky, N. *Aspects of the Theory of Syntax*. Cambridge, Massachusetts, M.I.T. Press, 1965.

Codd, E.F. "A Relational Model of Data for Large Shared Data Banks." *Communications of the ACM*, 1970, 13, 377-387.

Damerau, F. "Automated Language Processing." In Williams, M.E., ed., *Annual Review of Information Science and Technology*, Volume 11. Washington, D.C., American Society of Information Science, 1976.

Grishman, R. "Implementation of the String Parser of English." In R. Rustin, ed., *Natural Language Processing*. New York, Algorithmics Press, 1973. Pp. 89-109.

Grishman, R., Sager, N., Raze, C., and Bookchin, B. "The Linguistic String Parser." 1973 National Computer Conference. AFIPS Conference Proceedings, Volume 42. Montvale, New Jersey, AFIPS Press, 1973. Pp. 427-434.

Harris, Z. S. *Mathematical Structures of Language*. New York, Wiley, 1968.

Hillman, D. J., and Karsada, A. J. "The LEADER Retrieval System." Proceedings of the 1969 Spring Joint Computer Conference. *AFIPS Conference Proceedings*, 1969, 34, 447-455.

Hirschman, L., Grishman, R., and Sager, N. "Grammatically-Based Automatic Word Class Formation." *Information Processing and Management*, 1975, 11, 39-57.

Hirschman, L., Grishman, R., and Sager, N. "From Text to Structured Information: Automatic Processing of Medical Reports." 1976 National Computer Conference. *AFIPS Conference Proceedings*, Volume 45. Montvale, New Jersey, AFIPS Press, 1976. Pp. 267-275.

Hobbs, J., Grishman, R., and Raze, C. "The Automatic Transformational Analysis of English Sentences: An Implementation." *International Journal of Computer Mathematics*, 1976, SEC. A, 5, 267-283.

Petrick, S. R. "On Natural Language Based Computer Systems." *IBM Journal of Research and Development*, 1976, 20, 314-325.

Plath, W. J. "REQUEST: A Natural Language Question Answering System." *IBM Journal of Research and Development*, 1976, 20, 326-335.

Sager, N. "The String Parser for Scientific Literature." In R. Rustin, ed., *Natural Language Processing.* New York, Algorithmics Press, 1973. Pp. 61-87.

Sager, N. "The Sublanguage Technique in Science Information Processing." *Journal of the American Society for Information Science*, 1975, 26, 10-16.

Sparck Jones, K., and Kay, M. *Linguistics and Information Science.* New York, Academic Press, 1973.

Walker, D. E. "Automated Language Processing." In Cuadra, C. A., and Luke, A. W., eds., *Annual Review of Information Science and Technology*, Volume 8. Washington, D.C., American Society for Information Science, 1973.

Woods, W. A., Kaplan, R. M., and Nash-Webber, B. *The Lunar Sciences Natural Language Information System: Final Report.* BBN Report No. 2378, Bolt Beranek and Newman, Cambridge, Massachusetts, 1972.