LSP25

ENCYCLOPEDIA OF COMPUTER SCIENCE AND TECHNOLOGY

EXECUTIVE EDITORS

Jack Belzer *Albert G. Holzman* *Allen Kent*

UNIVERSITY OF PITTSBURGH
PITTSBURGH, PENNSYLVANIA

VOLUME 11

Minicomputers to Pascal

MARCEL DEKKER, INC. • NEW YORK and BASEL

MARCEL DEKKER, INC.
270 Madison Avenue, New York, New York 10016

LIBRARY OF CONGRESS CATALOG CARD NUMBER: 74-29436
ISBN: 0-8247-2261-2

Current printing (last digit):
10 9 8 7 6 5 4 3 2 1

PRINTED IN THE UNITED STATES OF AMERICA

NATURAL LANGUAGE ANALYSIS AND PROCESSING;

see also Machine Translation

Because of the complexity of language material in its given form, the initial, and major, problem is to reduce the successive sentences of a discourse (i.e., a text or conversation) to a simpler regular form in which the meaning elements necessary for useful processing can be located by straightforward procedures. This process is generally thought of as consisting of a syntactic component and a semantic component. In some kinds of applications it is followed by a so-called pragmatic component related to the specific task to be performed. The syntactic component segments the sentence into subunits which are grammatically connected, and identifies the grammatical role of words within the subunits. The semantic component furnishes a representation of the sentence which aims to capture the underlying relations of words, which are essential in conveying the information in the discourse.

OBJECTIVES OF NATURAL LANGUAGE PROCESSING

The general objective of work in this field is to develop procedures which make it possible to process the informational content of texts and conversation. An implicit goal is to learn more about language structure and use. Since the procedures are for the most part analytical (start with the text, produce the analysis) and must be both precise and penetrating, there is promise that the ways of organizing the linguistic facts and carrying out the analysis will reflect, at least in part, the inherent organization of data in language and the means by which sentences are processed by the human mind.

Applicational goals include the use of these procedures to make information which is recorded in natural language more readily accessible to users. Where the documents are stored in computer readable form, it would be desirable to have computer programs that could extract and assemble the relevant information from the computer store by processing the natural language documents. This goal becomes more prominent as natural language data bases become increasingly available.

NATURAL LANGUAGE ANALYSIS AND PROCESSING

A major goal is to develop systems for man-machine communication in natural language. Such a "front end" would be useful in information retrieval, to enable users of computer search systems to state their requests in natural language. Question-answering systems are one type of development in this direction. In these systems questions in natural language which apply to a given structured data base (table or the equivalent) are interpreted and an answer is returned. There would also be applications of natural language "front ends" in automatic programming, where it would be desirable to have a system which can interpret instructions given in natural language (or in a language very like a natural language) and can interpret declarative statements giving information required for programming tasks.

Applications of automatic language analysis in the language domain itself include mechanical translation, one of the earliest goals, and the use of the computer in language and literary research, such as in the testing of grammars and in style analysis. Use of natural language processing in computer-aided-instruction (CAI) and specifically in language teaching is also a possibility.

NATURAL LANGUAGE VS FORMAL LANGUAGES

There are reasons why natural language can be processed by computer programs. Overtly, it is composed of linearly arranged discrete elements which, like the symbols of mathematics and various codes, occur only in particular combinations. For this reason, a properly formulated grammar, which specifies the rules of formation for well-formed sequences of the language, can be used in a procedure to recognize the syntactic structure of sentences. Such a procedure produces a structural description of a sentence (or more than one if the sentence is ambiguous) which shows how the sentence was composed of elements specified by the grammar, combined according to the rules of sentence formation. In this respect natural language resembles its near-neighbors, the formal languages of mathematics, logic, and programming, for which recognition procedures have been devised. But there are significant differences between natural languages and formal languages which have made the computer analysis of natural language a special area of research. Some of the unique features of natural language from a computational point of view are the following.

Differing Acceptabilities

While the symbols in syntactic formulas for natural language represent word classes (e.g., N noun, TV tensed verb) analogous to the variables of mathematics, not every combination of members of the classes in a formula makes an equally acceptable instantiation, as though the formula $(a + b)^2 = a^2 + 2ab + b^2$ were to be more true for some numbers than for others. Thus, "Fire spreads" is a well-formed assertion of the N TV type whereas "Fire requires" is not. No matter how refined the word classes represented by formula symbols are made (e.g., making transitive verbs and intransitive verbs major classes), some differences in the acceptability of particular words in particular syntactic relations remains.

Grammatical Subclasses

As a result of the above property of natural language, a grammar contains constraints which refer to subclasses of the major classes. Words must be assigned attributes that indicate subclass memberships, and analysis procedures must be equipped with the ability to access the lexical entries for words and to eliminate readings in which words are assigned syntactic roles incompatible with their subclass memberships. Otherwise, nonwell-formed input strings will be accepted (e.g., "Fire requires") and wrong analyses of correct sentences will be obtained (e.g., "Fire requires oxygen being present," analyzed like "Prices rise everything being equal").

Selection

In some cases the unacceptability of particular word combinations is grammatical in character (sharp yes-no) as in the example "Fire requires," where a transitive verb is in the position of an intransitive verb. Another example of a grammatical constraint involving subclasses is number agreement: "John walks," but not: "John walk." However, a characteristic of natural language is that over and above what can be captured in grammatical constraints, the acceptability of word combinations in grammatically well-formed sequences is graded and dependent on the universe of discourse. Thus "John walks to school" is normal and "John breathes to school" marginal, while "John floats to school" might be acceptable if John's school is on a waterway and someone gives his boat a strong schoolward push in the morning. For a given word *w*, the set of words which *w* commonly co-occurs with in a given syntactic relation is called (following the linguist Bloomfield) the selection of *w* in that relation; e.g., if *w* is a verb, then it has a selection of subject nouns, and if transitive a selection of object nouns.

The selection of a word is closely related to its meaning. If two words in the same grammatical class have identical selections, they are exact synonyms, and conversely, if they have few linguistic environments in common, their meanings are very different. Selection is difficult to state precisely for the words of a whole language and differs markedly from field to field (consider the varied uses of the word "field" itself). However, in particular areas of discourse, especially those where the vocabulary is limited and usage is regular (e.g., science subfields), selectional classes can be defined precisely and selectional rules have virtually the force of grammatical constraints. E.g., in cell biology: "Ions enter the cell," but not: "The cell enters ions."

MAJOR GRAMMATICAL THEORIES

Certain major schemata, or theories, of grammatical structure in linguistics have also been used in computational efforts to treat language on a broad scale.

Immediate Constituent Analysis (ICA) was initially formulated by Leonard Bloomfield [1]. It describes the structure of a sentence as a sequence of certain kinds of segments (e.g., noun phrase + verb phrase), each of which is characterized as being composed in turn of certain segment sequences, down to the words or morphemes of the language. This type of grammar is readily expressed as a set of context-free productions, i.e., as rules of the form A → B, where A is a symbol from a given alphabet of symbols, and B is also such a symbol or sequence of symbols. This formalism is well suited for specifying the gross structure of sentences (whether in terms of ICA or of other grammatical formulations) but does not provide a means for treating such essential features of natural language as grammatical subclass constraints and selection, noted above. ICA was used in a context-free formulation in an early system developed by the RAND Corp. The system employed a fast bottom-up parsing algorithm, but because of the limits of context-free grammar was unable to overcome the problem of multiple analyses.

String analysis is a grammatical formulation that was developed (by Harris in 1959) specifically for language computation. It differs from ICA primarily in that it isolates in any given sentence an elementary sentence, called the center string, on which the rest of the sentence is built. Further additions to the center string are elementary word strings of given types, called adjunct strings, which adjoin to the left or right of particular elements in other strings in the sentence. When one compares a string analysis with an immediate constituent analysis of a sentence, one sees that the successive words of an elementary string are the centers of endocentric constructions (constructions of the type X containing an X in them) in the constituent analysis. E.g., "A small dog barks loudly" has the immediate constituents Noun phrase = "a small dog" and Verb phrase = "barks loudly," both of which are endocentric. The centers of these constructions are, respectively, "dog" and "barks," which together constitute the center string of the sentence under string analysis.

String analysis is used in the system developed by the Linguistic String Project at New York University (NYU). The utility of string analysis for computation lies in the fact that there are simple rules for combining elementary strings to form sentences, and that grammatical and selectional constraints apply to words within one elementary string or in contiguous strings related by adjunction. This means that the grammatical and selectional constraints that are needed to obtain the correct analysis can be applied locally, without extensive searches of the parse tree. For example, number agreement and other constraints can be tested by the same procedure in elementary sentences ("A dog barks") and in more complicated sentences ("A small dog which confronts a large dog barks loudly") because the relevant words are contiguous in the elementary string. This is true regardless of how many words intervene between the relevant words in the given sentence. It is also found that the component elementary strings in a given sentence S bear a close relation to the elementary sentences that are the components of S under transformational analysis.

Transformational analysis was introduced into linguistics (by Harris in 1952) as an outgrowth of work on discourse structure. In that work it was clear that many sentences and sentence parts, while differing in grammatical form, were similar in content and, in fact, contained the same vocabulary items except for certain grammatical words or affixes, e.g., "which," "by," "-ing." When two or more such forms satisfied certain conditions, they were called transforms of each other. The forms were understood to be sequences of variables of the type N noun, V verb, etc. (called variables because their values are different words in different sentences) and grammatical words or affixes, called constants of the transformation. A classic example is the active-passive transformation:

N_1 t V $N_2 \longleftrightarrow N_2$ t be V-en by N_1

Here N_1, t (tense), V, and N_2 are the variables and "be," "-en" (past participle marker), and "by" are the grammatical constants. For example, if N_1 has the value "Withering," t the value "-ed" (past), V the value "use," and N_2 the value "digitalis," we have the transformational relation between two sentences:

Withering used digitalis in 1784 ⟷ Digitalis was used by Withering in 1784

Note that the information content is the same in both forms though a shift in emphasis or other stylistic change may be introduced by the transformation.

Transformations become the basis for a grammar when it is seen that all (or virtually all) the word-string components of a sentence—say, under string analysis—are derivable by known transformations from independent sentences. Thus, from "Withering used digitalis in 1784," we might have in addition to the above: "digitalis, which was used by Withering in 1784," "the use of digitalis by Withering in 1784," "Withering's 1784 use of digitalis," etc. A sentence, then, under transformational analysis, is decomposed into elementary source sentences plus transformations operating on the elementary sentences or on already transformed sentences. Some transformations, such as the active-passive, and the replacing of a particular noun by a pronoun, are purely paraphrastic; that is, they add no information and are only a rearrangement of parts or a change in the shape of words. Others add a fixed increment of meaning to the operand sentence, the same addition of meaning to all operand sentences (for example, "seems" in "The heart seems to respond," "The child seems to understand," etc.).

A question-answering system (REQUEST) that includes transformational analysis has been developed at IBM. The NYU Linguistic String Project system includes transformational decomposition of text sentences, operating on the string parse outputs. The relevance of transformational analysis is that while being a grammatical procedure, it nevertheless yields a uniform representation of the information in sentences. If, for example, two sentences contain the same information presented in different styles, or if they overlap in their information content, then the information which they carry in common will appear in the transformational analysis of the sentences as identical component elementary sentences with the same incremental transformations; they will differ only in the paraphrastic transformations in the decomposition. This means, in principle, that sentences can be compared in a standard fashion for sameness or overlap in content.

Transformational generative grammar was introduced by Chomsky [2, 3] as a system for generating sentences. The system combined elements of immediate constituent analysis (in the form of phrase structure rules) with transformations. The grammar contains a small set of phrase structure productions, with the root symbol S for sentence (e.g., S → NP VP, NP → Det N, VP → verb), and a mechanism for realizing terminal symbols as words. These are used to generate the "deep structure" of the sentence. Transformational rules operate on the deep structure tree and on successive trees produced by the operation of transformations until a final "surface structure" tree is obtained. This structure is then interpreted by means of phonological, semantic, and logical rules.

This formulation of grammar has been used in several computational systems, chiefly at the MITRE Corp. and IBM. However, reversing the generative process in order to analyze sentences requires a number of extra steps that are necessitated by the fact that the grammar is oriented toward generation rather than analysis. First a set of surface structure trees must be obtained for the sentence. This requires a parsing, or covering, grammar which is different from the transformational grammar. The set of surface structure trees which are obtained should contain the one which would be generated for the given sentence by the transformational grammar. The procedure must determine which sequence of transformations operated to produce the surface structure and what was the deep structure tree that was generated using the phrase structure rules. One source of difficulty is that too many surface structure trees may be generated, requiring that many different sequences of reverse transformations be tried. Another problem is that at each point where a reverse transformation is to be applied, a number of reverse transformations might qualify, thereby increasing the number of paths that have to be checked. In practice, special measures to increase efficiency are required or the process takes too long. Also, because of the multiple sources of error, each analysis which is obtained must be verified by a forward generation process.

For dependency grammar, see the article under that heading in Volume 7.

BRIEF HISTORY OF LANGUAGE COMPUTATION

Early research projects in natural language processing, in the period dating from the late 1950s through the mid-1960s, for the most part had large applicational goals and were optimistic about rapid progress. Many projects were directed toward mechanical translation (MT), and some projects were concerned with processing texts for information retrieval. In virtually all the major efforts, the goal was to be able to treat the language as a whole, not a particular subset relating to a specific subject matter or application. One might mention from this period the earliest translation program (Russian to English) of Ida Rhodes at the Bureau of Standards in 1959, based on Predictive Analysis, a method which was later extended and adapted to English by Oettinger and Kuno in the system known as the Harvard Predictive Analyzer. The earliest English syntactic analysis program was developed at the University of Pennsylvania in 1958-1959 in a system which combined ICA with string analysis. This program was used to provide the language analysis in the first question answering system (Baseball 1961). String analysis was extended and implemented for practical applications of text analysis at New York University from 1965 onward, and was used in a program for analyzing narrative surgical reports at Roswell Park Memorial Institute in Buffalo. A major effort in language analysis for information retrieval in French was the SYNTOL Project (cf. references in Sparck Jones and Kay [4]). In the 1960s several large projects were based on the transformational generative grammar, principally those of the MITRE Corp. and IBM. The early systems were mainly documented in report series, of which the major ones are listed in the Resources section at the end of this article.

The initial hopes of rapid progress toward large-scale applications received serious setbacks when the language problem proved to be more complex and unwieldy than had been realized at first. Many MT projects had relied primarily on dictionary look-up and were faced with problems due to the syntactic complexity of text sentences and the differences in grammatical requirements in language pairs, as well as the lexical problems of translation. The MT field fell under serious criticism when the large expenditure of research funds failed to yield commensurate results (1966 Report of the Automated Language Processing Advisory

Committee of the National Academy of Science, Language and Machines—Computers in Translation and Linguistics). Also, in the area of single-language text analysis, although there was progress, the magnitude of the linguistic data which had to be organized and the problem of ambiguities in the analysis turned out to be major stumbling blocks, requiring the development of special tools.

The result of these developments was that in the late 1960s and in the 1970s, much of natural language processing research concerned itself with what could be accomplished using sophisticated software on narrowly delineated language areas. Question-answering systems flowered, since here the scope of the problem was limited syntactically to question forms and semantically to the exact categories of the given data base, which was usually a numerical table or the equivalent. Some systems reached practical operational status (e.g., Bolt Beranek and Newman's system for lunar rock data retrieval, and IBM's REQUEST system operating on business statistics). Systems which combined syntactic, semantic, and task-oriented procedures were referred to as integrated systems, and included experiments in robotics with natural language instructions. Of the early projects, the Linguistic String Project has continued the development of a large grammar with a broad capability for text analysis.

Renewed interest in natural language processing on the part of information scientists and others arose in the 1970s with the advent of machine-readable natural language data bases, interactive on-line systems (that can take over some of the burden of analysis in man-machine dialog), and the steady advance on all fronts in the computer field, providing lower costs, increased availability of components, and more sophisticated software. Medicine, education, business, and government all appeared to have one eye cocked toward natural language processing should it indeed prove possible to communicate with machines in natural language and to access computer-stored written information by means of computer programs.

REQUISITES FOR COMPUTER ANALYSIS OF LANGUAGE

Part of the reason why progress in natural language processing was slower than anticipated was the fact that the requisite linguistic and computational tools were not available and had to be developed.

Enriched Formalisms

As was noted above, natural language differs from formal languages in important respects. Before successful language computations could be performed, enriched formalisms and their associated recognition procedures had to be developed. One approach has been to specify the grammar on two levels. Context-free productions, or the equivalent, are used to define parse tree structures, and procedures which operate on the generated parse trees are the means for expressing grammatical and semantic restrictions and transformations. This approach has proved suitable in several forms, e.g., using string analysis and transformations as the framework, and using augmented transition network (ATN) grammars.

Computer Grammars

The possibility of incorporating into a computer program the large number of linguistic facts needed for correct analysis of sentences depends on having an appropriate linguistic theory and a sufficiently rich formalism. But this is not enough. The actual grammar has to be written, and if semantic interpretation is to be done, the lexical coding and the procedures for this stage of processing also have to be written. The writing of a computer grammar of a natural language is a big undertaking. One could imagine, as many early investigators did—hence their optimism—that existing grammars of languages would provide the necessary rules and that the problem would simply be one of coding the rules for use in procedures. But that has proven not to be the case. A conventional grammar, such as Jespersen's monumental seven-volume work on English, while rich in detail, is not organized or formulated in such a way that the observations can easily be assimilated into a formal framework. On the other hand, much of the grammatical work that has been done in a formal framework, chiefly in transformational generative grammar, has not led to an integrated grammar of the whole language, and does not deal with the issues faced in recognition. Computational linguists have therefore had to write their own grammars, or parts of grammars, covering the range of language material their systems are intended to handle.

The positive side of this is that some linguistic phenomena have been formulated in greater detail than ever before. The negative side is that in most cases a "grammar" which is specified for a particular subset of a language is not readily extendible to cover the whole language, or a significant portion thereof. The reason is that the treatment of one phenomenon affects how others can be treated, so that a global view of the grammar is needed from the start or the system becomes unwieldy on expansion, with conflicts and redundancies in the analysis. Thus the field has seen very many small coverage systems which are extendible in principle but which are short-lived in practice because of this problem.

Treatment of Ambiguity

In addition to the fact that a suitable grammar has to be written, the computational linguist faces the problem that a syntactic analyzer "sees" possible readings of the sentence that a person is unaware of. Partly, this points to the need for selectional or semantic constraints. However, in addition to these, some weighting as to which are the more likely grammatical configurations appears to be necessary. For example, in the following instance of syntactic ambiguity, all three readings obtained by a parsing program were semantically possible, but the grammatical analysis on the first parse was the more likely of the three. The program obtained the following three analyses for the sentence "We have studied membrane potentials recorded with intracellular microelectrodes." The first was the intended reading, in which "have studied" is a verbal unit occurring with "membrane potentials recorded with intracellular microelectrodes" as its noun phrase object. In this noun phrase, "recorded . . ." is a passive adjunct of "potentials." However, the program also found two other possible uses of "have" in this sentence, each of which meshed with another source of ambiguity so as to produce an additional reading. In one, the main verb is "have" ("We have potentials") and "studied" is a passive modifier of "potentials," paraphrastic to "We have potentials which were studied." The other spurious reading contains the same analysis of "studied potentials" as a noun group, but this time as the embedded subject of "recorded" under the verb "have": "We have potentials recorded," paraphrastic to "We have arranged that potentials be recorded." The ambiguity in this sentence illustrates a new kind of problem to be studied: what are the more likely combinations of grammatical forms.

In addition to those syntactic ambiguities which involve different segmentations of the sentence, illustrated above, alternative readings arise due to the reassignment of a given segment to be a modifier of a different element in the sentence, e.g., "He wrote a book on cooking in the Chinese style." Eliminating the unintended readings in these cases usually depends on selectional constraints for the type of discourse or on regularities observable in the context. Ambiguity resolution often requires constraints on several levels and is not usually attainable without restricting the system to a particular subject domain.

Treatment of Implicit Elements

In addition to treating the words that are present in a discourse, a natural language processor has to deal with some words that are present in "zeroed" form, that is, are not physically present but are reconstructible from the context, e.g., "left" after "she" in "He left and she too." Also, some words refer to other words or stretches of the discourse, e.g., personal pronouns and phrases like "this process," "the foregoing." To complete the analysis and to perform content-oriented tasks, the zeroed words must be supplied and the antecedents of referentials identified. In addition, on the discourse level, one might have to supply entire implicit sentences (sentences assumed by the reader in order to read the text as a coherent discourse). These and related matters are discussed in the literature in papers on conjunctions, reference resolution, discourse structure, and inference systems.

Approaches to Meaning

There is no single path, valid for the language as a whole, from the syntactic analysis of sentences to their meaning. The syntactic analysis clearly provides some information (e.g., what is the subject and what is the object of the action expressed by a verb in an assertion), but to arrive at an exact and useful characterization of the contents of sentences, special processing using specialized lexicons is required. One approach is to derive the relevant categories for the words in an area of discourse by examining detailed co-occurrence patterns (selection) of words in subject-verb-object and similar relations in samples of the textual material. Another approach has been to provide the relevant categories from general knowledge, based on the view held by various investigators that semantic categories are largely independent of syntactic relations in the texts and can be stated a priori for the language as a whole, or significant portions of it. The purely semantic approach has yet to be demonstrated on a broad enough range of subject matters so that its merits can be evaluated.

STAGES OF PROCESSING

Although systems differ as to how they distribute the burden of analysis between the syntactic component and the semantic component, both types of processing are present in some form, and can be presented here as operating serially. It should be noted that both components require a lexicon in which words are given classifications needed for processing, and that this represents a considerable share of the cost and effort of setting up a workable system.

Parsing

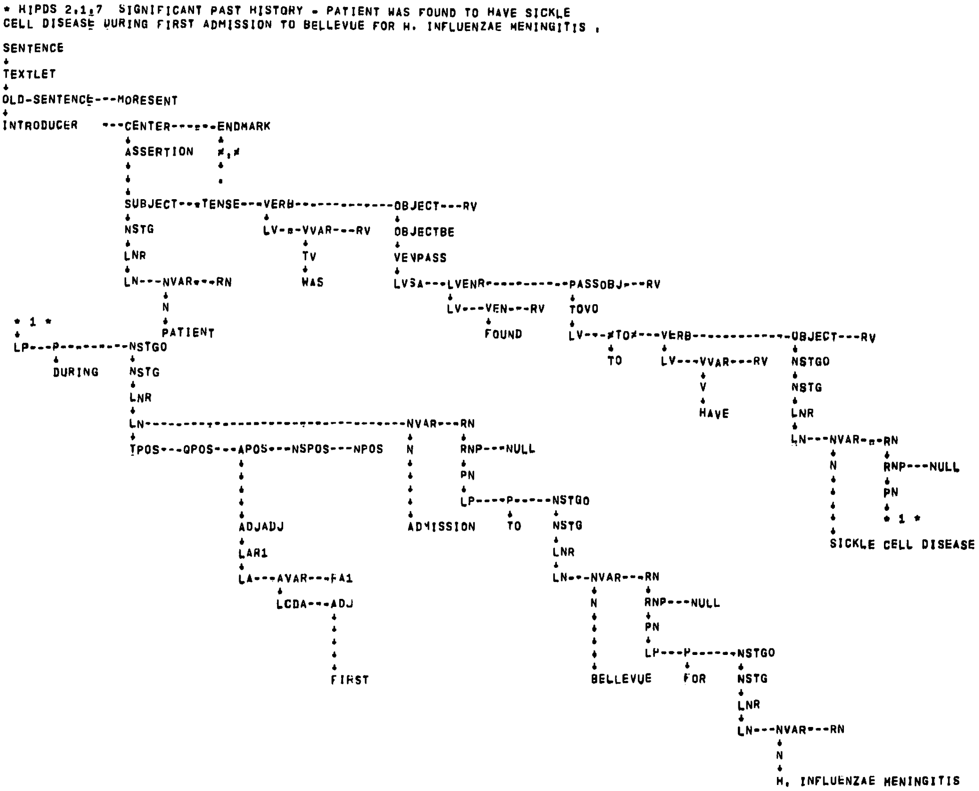
When a sentence is read into the computer for processing, the first step is to look up the words in the lexicon and associate with each successive word of the input string its major classifications and subclassifications. (In some applications not every word need be coded prior to processing.) For example, the major classes associated with each word of the following sentence from a medical record are

Patient	was	admitted	to	hospital	for	meningitis.
N/ADJ	TV	TV/VEN	P	N	P	N.

In addition, grammatical subclass information such as the following might be noted: "Patient" as a noun is singular, and so are the nouns "hospital" and "meningitis" and the verb "was." Also, for semantic processing it might be noted that in the medical sublanguage from which the words are drawn, "patient" is in a distinguished class (PATIENT), "admit" is a medical action verb (V-MD), "hospital" is a medical institution word (INST), and "meningitis" is a diagnosis word (DIAG). The set of subclasses for a particular sublanguage can be obtained by the analysis of word co-occurrence patterns in samples of the textual material.

The determination of syntactic structure is accomplished by a parsing algorithm that draws upon the grammar and the lexical attributes of the sentence words. The object of parsing is to produce a structural representation of the sentence, such as the one shown in Fig. 1, for the sentence: "Significant past history—Patient was found to have sickle cell disease during first admission to Bellevue for H. Influenzae Meningitis," a typical sentence from a hospital discharge summary. (The dropping of the definite article is typical of the telegraphic style of reports and records.) In the parse tree of Fig. 1, sibling nodes are connected by a horizontal line and the parent node is attached to the left-most daughter node only; branches end in terminal nodes (e.g., N, TV) or literals (e.g., "to") associated with sentence words (shown just below the terminal symbol or literal) or in NULL (not shown). The paragraph heading ("Significant Past History") is not parsed.

Parsing strategies differ. Figure 1 was obtained using a top-down algorithm in conjunction with a grammar that had two components, a set of context-free productions for generating parse tree structures of the linguistic string type, and a set of procedures, called restrictions, associated with particular elements in the productions, for checking the well-formedness of word-subclass combinations (e.g., number agreement, selection). Briefly, the top-down parser generates a parse tree from the context-free productions and attempts to match each successive terminal node of the tree with a word class assignment of the current sentence word, stepping from left-to-right through the sentence. If a terminal node X matches the X category of the current sentence word, a pointer is created from X



```
* HIPDS 2.1.7  SIGNIFICANT PAST HISTORY - PATIENT WAS FOUND TO HAVE SICKLE
CELL DISEASE DURING FIRST ADMISSION TO BELLEVUE FOR H. INFLUENZAE MENINGITIS .
SENTENCE
↓
TEXTLET
↓
OLD-SENTENCE---MORESENT
↓
INTRODUCER   ---CENTER------ENDMARK
                ↓           ↓
                ASSERTION   ≠,≠
                ↓           ↓
                ↓           .
                ↓
                SUBJECT---TENSE---VERB------------OBJECT---RV
                ↓                 ↓               ↓
                NSTG              LV---VVAR---RV  OBJECTBE
                ↓                      ↓          ↓
                LNR                    TV         VENPASS
                ↓                      ↓          ↓
                LN---NVAR---RN         WAS        LVSA---LVENR-----------PASSOBJ---RV
                     ↓                                   ↓               ↓
                     N                                   LV---VEN---RV   TOVO
 * 1 *               ↓                                        ↓          ↓
 ↓                   PATIENT                                  FOUND      LV--≠TO≠----VERB-------------OBJECT---RV
 LP----P--------NSTGO                                                         ↓      ↓                ↓
       ↓        ↓                                                             TO     LV---VVAR---RV   NSTGO
       DURING   NSTG                                                                      ↓           ↓
                ↓                                                                         V           NSTG
                LNR                                                                       ↓           ↓
                ↓                                                                         HAVE        LNR
                LN----------------------------------NVAR---RN                                         ↓
                ↓                                   ↓      ↓                                          LN---NVAR---RN
                TPOS---QPOS---APOS---NSPOS---NPOS   N      RNP---NULL                                      ↓      ↓
                              ↓                     ↓      ↓                                               N      RNP---NULL
                              ↓                     ↓      PN                                              ↓      ↓
                              ↓                     ↓      ↓                                               ↓      PN
                              ↓                     ↓      LP----P-----NSTGO                               ↓      ↓
                              ↓                     ↓            ↓     ↓                                   ↓      * 1 *
                              ADJADJ                ADMISSION    TO    NSTG                                ↓
                              ↓                                        ↓                                   SICKLE CELL DISEASE
                              LAR1                                     LNR
                              ↓                                        ↓
                              LA---AVAR---FA1                          LN---NVAR---RN
                                   ↓                                        ↓      ↓
                                   LCDA---ADJ                               N      RNP---NULL
                                          ↓                                 ↓      ↓
                                          ↓                                 ↓      PN
                                          ↓                                 ↓      ↓
                                          ↓                                 ↓      LP---P------NSTGO
                                          ↓                                 ↓           ↓      ↓
                                          FIRST                             BELLEVUE    FOR    NSTG
                                                                                               ↓
                                                                                               LNR
                                                                                               ↓
                                                                                               LN---NVAR---RN
                                                                                                    ↓
                                                                                                    N
                                                                                                    ↓
                                                                                                    H. INFLUENZAE MENINGITIS
```

FIG. 1. Parse tree (Figs. 1-3 are outputs of the NYU Linguistic String Project System.) Node names—terminal symbols: N noun, P preposition, ADJ adjective, TV tensed verb, VEN past participle.

Types of nonterminal symbols: For X = a terminal symbol: LX left adjuncts of X; RX right adjuncts of X; LXR a sequence of LX + X + RX or of LX + XVAR + RX; XVAR local variants of X; XPOS position of X-occurrence among ordered adjuncts.

Other node names: SA sentence adjunct; NSTG noun string; NSTGT noun string of time; NSTGO noun string in object; PN prepositional phrase; ADJADJ repeating adjectives; LCDA left adjuncts of compound adjective; OBJECTBE object of be; VENPASS passive string; LVSA left adjuncts of verb in participial SA string; PASSOBJ object in passive string.

Output conventions: A prepositional phrase PN which has several possible positions of adjunction is assigned in the parse tree to the nearest PN slot (here BELLEVUE FOR MENINGITIS rather than ADMISSION FOR MENINGITIS). The later stages of processing correct the assignment on the basis of word co-occurrence classes.
]

in the parse tree to X in the lexical entry for that word. Thus, in Fig. 1, when the terminal node N under SUBJECT matched the N category of "patient," the subclassifications of "patient" in the lexicon (e.g., SINGULAR became attributes of the N in the parse tree.

The parser is also equipped with a restriction interpreter, tree climbing operators, logical operators, and attribute-testing operators. These enable it to test the parse tree, including the attributes (subclasses) that have been attached to terminal nodes. Thus, after the SUBJECT and VERB subtrees have been completed, the parser can apply an agreement restriction by making the following test: Starting at the node VERB, descend to the node TV and test for the attribute SINGULAR/PLURAL; store the result. Now start again from VERB, go to the sibling SUBJECT; descend to the noun; test for SINGULAR/PLURAL; compare this with the stored attribute and register success if the items match. (This is a much simplified version of the real restriction.) If all the restrictions associated with a node succeed, the parsing continues; if not, the subtree is rejected and the parser "backs up" to try other grammar options for building the tree.

For a description of alternative parsing algorithms, a survey such as Grishman 1975 in the Courant Report Series (cf. Resources section below) can be consulted.

Grammatical Regularization

The parse tree gives valuable information about the sentence but it has several limitations as a structural representation for information processing. Language has too many different grammatical structures to deal with, and some of the different structures contain the same information. Adding a stage of transformational analysis eliminates grammatical paraphrases; it results in fewer forms and leads to a canonical representation of information. For example, two sentences which contain, respectively, "Ca^{++} exchanges with other cations in SR" and "the exchange of Ca^{++} with other cations in SR," carry the same information over these stretches, 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 transformational analysis will be closer to a representation of its contents than either the original word string or, in most cases, the syntactic parse.

An example of a transformational decomposition obtained by applying (reverse) transformations to the output of a string parsing program is shown in Fig. 2 for the same sentence as in Fig. 1. Figure 2 shows the computer output for a transformational decomposition obtained by applying reverse transformations to the parse tree output shown in Fig. 1. In this form of output the sentence is decomposed into elementary ASSERTION structures in which the VERB (or adjective or preposition—also labeled VERB) precedes the SUBJECT and OBJECT node as in Polish notation. The transformations used to obtain each ASSERTION appear as T-nodes above the ASSERTION. Thus the main ASSERTION was obtained from the original parse tree structure for "Patient was found to have sickle cell disease" by the application of the PAST tense transformation and the PASSIVE transformation. The PAST tense transformation recognized "was" as a past tense verb and replaced it by a tenseless form of "be" (later itself replaced). The PASSIVE transformation in effect turned "Patient be (past) found to have sickle cell disease" into "( ) find (past) patient to have sickle cell disease," where ( ) represents an unstated subject. Under the OBJECT node of this ASSERTION the T-NTOVO transformation converted the

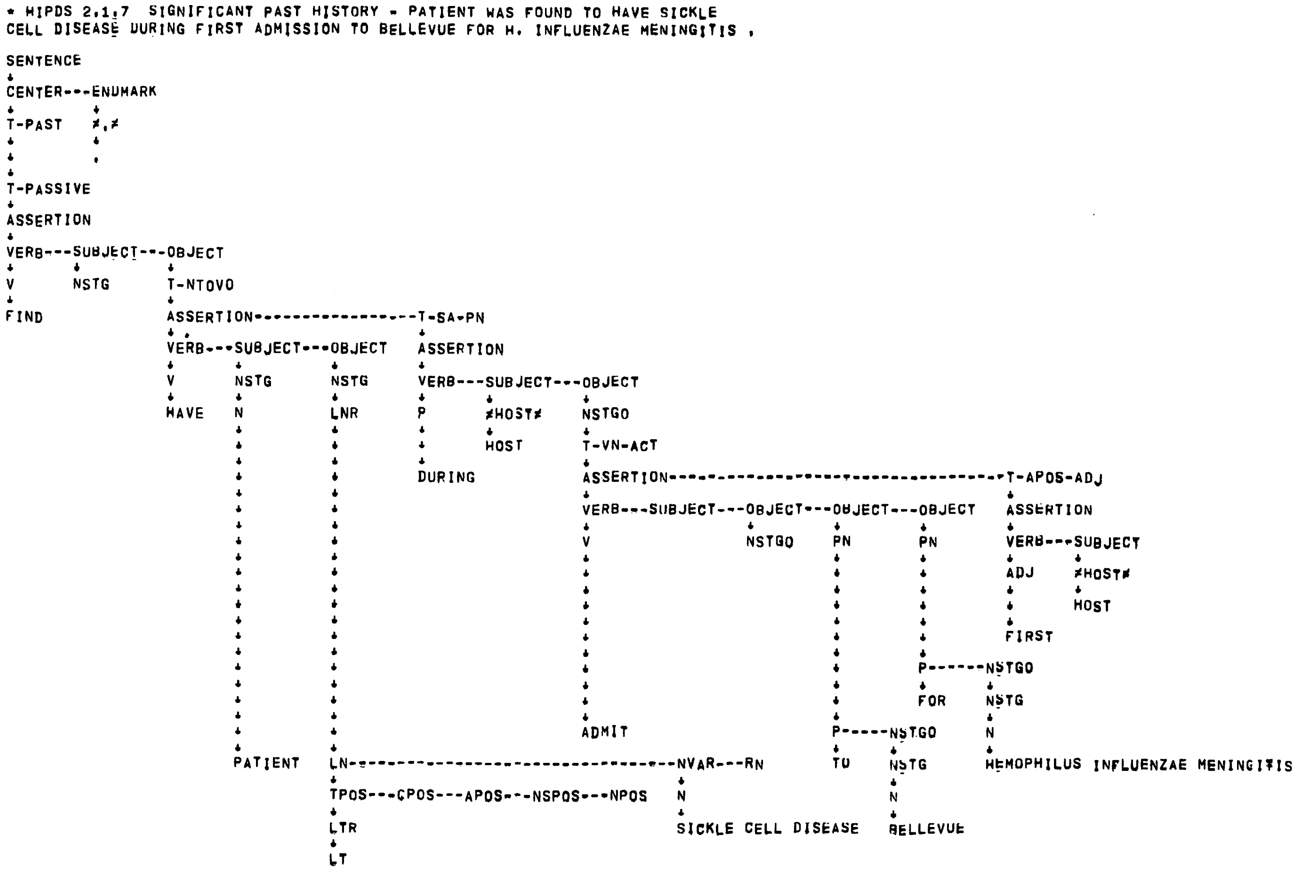
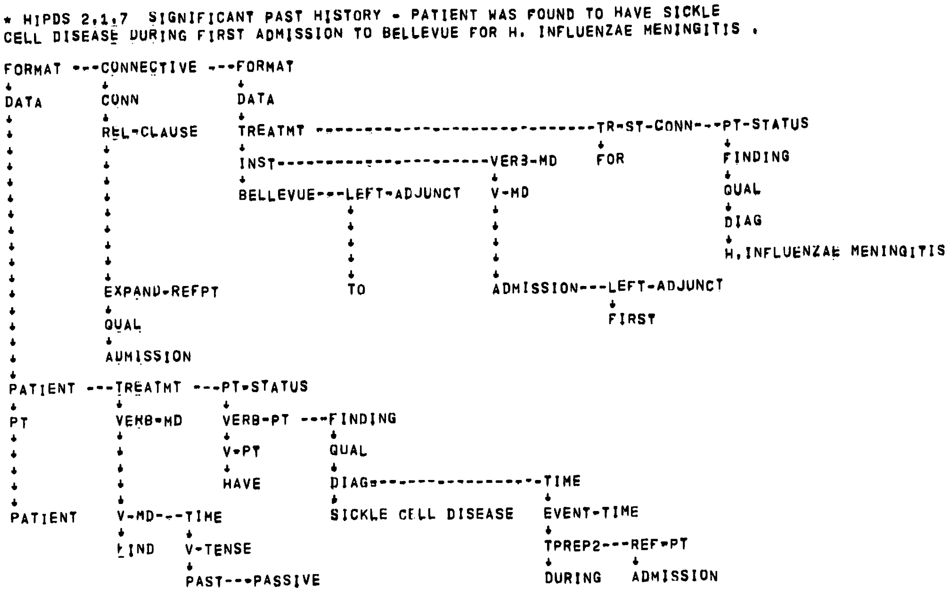


FIG. 2. Transformational decomposition. The subtree under a transformational node (labeled T-x) is the result of applying the reverse transformation named by T-x to the appropriate parse tree structure. If the subtree under such a node contains a node N_1 which is a copy of a node N_2 in the structure to which T-x is connected, then the value of N_1 is a node HOST which contains a pointer to N_2. In the figure the HOST nodes each contain a pointer to the second ASSERTION above HOST in the tree.



```
* HIPDS 2,1,7  SIGNIFICANT PAST HISTORY - PATIENT WAS FOUND TO HAVE SICKLE
CELL DISEASE DURING FIRST ADMISSION TO BELLEVUE FOR H. INFLUENZAE MENINGITIS .

FORMAT ---CONNECTIVE ---FORMAT
↓         ↓             ↓
DATA      CONN          DATA
↓         ↓             ↓
↓         REL-CLAUSE    TREATMT -----------------------------TR-ST-CONN---PT-STATUS
↓         ↓             ↓                                    ↓            ↓
↓         ↓             INST----------------------VER3-MD    FOR          FINDING
↓         ↓             ↓                         ↓                       ↓
↓         ↓             BELLEVUE---LEFT-ADJUNCT   V-MD                    QUAL
↓         ↓                        ↓              ↓                       ↓
↓         ↓                        ↓              ↓                       DIAG
↓         ↓                        ↓              ↓                       ↓
↓         ↓                        ↓              ↓                       H,INFLUENZAE MENINGITIS
↓         ↓                        ↓              ↓
↓         EXPANU-REFPT             TO             ADMISSION---LEFT-ADJUNCT
↓         ↓                                                   ↓
↓         QUAL                                                FIRST
↓         ↓
↓         AUMISSION
↓
PATIENT ---TREATMT ---PT-STATUS
↓          ↓          ↓
PT         VERB-MD    VERB-PT ---FINDING
↓          ↓          ↓          ↓
↓          ↓          V-PT       QUAL
↓          ↓          ↓          ↓
↓          ↓          HAVE       DIAG---------------------TIME
↓          ↓                     ↓                        ↓
PATIENT    V-MD---TIME           SICKLE CELL DISEASE      EVENT-TIME
           ↓      ↓                                       ↓
           FIND   V-TENSE                                 TPREP2---REF-PT
                  ↓                                       ↓        ↓
                  PAST---PASSIVE                          DURING   ADMISSION
```

FIG. 3. Information structure.

object structure (called NTOVO) covering "patient to have sickle cell disease" into a tenseless ASSERTION "Patient have sickle cell disease." With regard to T-SA-PN in this analysis, the time expression "during first admission . . .," which modifies the main assertion, is transformed into an assertion which has the main assertion as its subject, the transformational paraphrase being: "Patient was found to have sickle cell disease, and this (act of finding) was during the first admission . . ." (see legend of Fig. 2 regarding HOST). An example of the nominalization transformation T-VN-ACT in Fig. 2 is the expansion of "admission to Bellevue for . . . meningitis" to "( ) admit ( ) to Bellevue for . . . meningitis," where again ( ) represents unstated arguments. A difficult part of the nominalization transformation is identifying the arguments of the verb which usually appear as prepositional phrases among other prepositional modifiers.

Complete transformational decomposition is important for calculating word co-occurrence similarities, since then we wish to have every occurrence of the same root word in a form where it can be counted with the others. In other applications it is possible to regularize the representation without actually carrying out every reverse transformation. For example, "admission" can be recognized as related to "admit" via subclassification rather than decomposition (as in Fig. 3) if the purpose is to develop an underlying semantic representation for the particular material rather than to reduce the sentences to a very general informational representation.

Semantic Representation

Most applications require a representation of sentences that is quite specific to the subject matter of the language material to be treated. At the same time the method of obtaining that representation must be general enough so that it covers paraphrastic variations and is not so special to the subject matter that a new system has to be built for each application. A sufficiently rich syntactic component which includes the reduction of grammatical paraphrases, such as was illustrated above, can with few adjustments provide the gross structure of sentences in any language material. This leaves the well-defined task of determining the more detailed relations of the words within these structures. It is found that in a given subject area the patterns of word co-occurrence in these structures correlate with the different kinds of information being transmitted. For example, in medical reporting, verbs that characteristically take as their subject such nouns as "hospital," "doctor," "clinic," and some other words, conveniently labeled the "medical institution class," generally have the semantic character of "actions taken in treating a patient." This is hardly surprising. However, the fact that this type of correlation between distribution and meaning occurs widely means that the classes needed for information processing and other applications can be derived from a study of co-occurrence patterns in samples of the language material.

In addition to determining semantic word classes, word co-occurrence patterns that are characteristic of a given subject matter can be systematized into formats for the information that is carried by the discourses in that subject area. A program can then map regularized parse outputs into the format. An example of such a structure for the medical reports sentence that has been carried through the stages of computer processing here is shown in Fig. 3. Each format unit for this material contains a TREATMENT part and a PATIENT STATUS part, though both are not present in every sentence. (Only format headings that subsume sentence words are printed.) References to the patient are brought out to the left; otherwise the order of entries follows normalized syntactic order. In Fig. 3 the first format unit has the medical action "find" and the patient status "have sickle cell disease," each with its own associated time. The second format unit has the medical action verb "admit" (in its untransformed state "admission") under TREATMENT and a connective ("for") between TREATMENT and PATIENT STATUS. In this case the admission was for the diagnosis "meningitis" (under FINDING under PATIENT STATUS). The connective between the two format units in Fig. 3 is the expansion of the time reference point "admission" of the first unit.

Figure 3 illustrates one kind of semantic structuring; other types are referenced in the Resources section. Here, node labels correspond to the types of entities and relations that are important in the subject area and are present in a regular way in the discourse. The syntactic relations of words in the sentence (after removing the effect of paraphrastic transformations) provide the skeletal structure of the format. In subject areas where such information formats can be defined, a system equipped with the three stages of processing—parsing, regularizing, and formatting—can map the sentences of documents into the formats. This provides a structured data base containing the same information as the documents. From this data base, programs can answer questions automatically and generate statistical summaries, such as (for the data base of the sample sentence): How many patients with symptom X also had symptom Y within time period Z? And so forth. It can be seen that while computers cannot "understand" natural language, they can be programmed to utilize the structural regularities of the language—both the syntactic regularities of the language as a whole and the usage regularities of specialized areas—so as to analyze the sentences of a discourse and arrange its informational content in relevant ways.

REFERENCES

1. L. Bloomfield, Language, Holt, Rinehart and Winston, New York, 1933.
2. N. Chomsky, Syntactic Structures, Mouton, The Hague, 1957.
3. N. Chomsky, Aspects of the Theory of Syntax, M.I.T. Press, Cambridge, Massachusetts, 1965.
4. K. Sparck Jones and M. Kay, Linguistics and Information Science, Academic, New York, 1973.

RESOURCES

Books on Language

Bloomfield, L., Language, Holt, Rinehart and Winston, New York, 1933.

Chomsky, N., Syntactic Structures, Mouton, The Hague, 1957.

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

Harris, Z. S., (Methods of) Structural Linguistics, University of Chicago Press, Chicago, 1951.

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

Harris, Z. S., Papers in Structural and Transformational Linguistics, Reidel, Dordrecht, 1970.

Jespersen, O., A Modern English Grammar on Historical Principles, Allen & Unwin, London, 1961.

Books on Language Computation

Charniak, E., and Y. Wilks (eds.), Computational Semantics; An Introduction to Artificial Intelligence and Natural Language Comprehension, North Holland, Amsterdam, 1976.

Friedman, J., et al., A Computer Model of Transformational Grammar, American Elsevier, New York, 1971.

Gross, M., Methodes en Syntaxe, Hermann, Paris, 1976.

Hays, D. G., Introduction to Computational Linguistics, American Elsevier, New York, 1967.

Rustin, R. (ed.), Natural Language Processing, Algorithmics Press, New York, 1973.

Schank, R., Conceptual Information Processing, North Holland, Amsterdam; Elsevier, New York, 1967.

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

Walker, D., H. Karlgren, and M. Kay, Natural Language in Information Science (FID Publ. 551), Skriptor, Stockholm, 1977.

Winograd, T. H., Understanding Natural Language, Academic, New York, 1972.

Conference Proceedings (sessions on natural language processing)

American Society for Information Science (ASIS)
Association of Computational Linguistics (ACL)

Association of Computing Machinery (ACM). Also ACM Special Interest Groups in Language, Arts and Studies in the Humanities (SIGLASH) and in Artificial Intelligence (SIGART)
International Federation of Information Processing Societies (IFIPS)
MEDINFO—Medical Informatics, in conjunction with IFIPS

Periodic Review Articles

Annual Review of Information Science and Technology (C. Cuadro, ed.), American Society for Information Science, Washington, D.C.
Advances in Computers (M. Rubinoff and M. Yovitz, eds.), Academic, New York.

Periodicals

Frequent Articles:

American Journal of Computational Linguistics
Computers and the Humanities
SIGART Newsletter (of the ACM)
SIGLASH Newsletter (of the ACM)

Occasional Articles:

Communications of the Association for Computational Machinery
Information Processing and Management
Journal of the Association for Information Science
Artificial Intelligence

Report Series[1]

Bolt, Beranek and Newman Reports. No. 2378, W. A. Woods et al., *The Lunar Sciences Natural Information System: Final Report*, 1972. [A question answering system for lunar rock data using an augmented transition network (ATN) parser.]
No. 2976, W. A. Woods et al., *Natural Communication with Computers, Final Report*. Vol. 1, *Speech Understanding Research at BBN, October 1970–December 1974*.

The Computation Laboratory of Harvard University. Reports to the National Science Foundation. S. Kuno et al., *Mathematical Linguistics and Automatic Translation*. (Using the Harvard Predictive Analyzer.)

Courant Computer Science Reports, New York University. No. 2, J. Hobbs, *A Metalanguage for Expressing Grammatical Restrictions in Nodal Spans Parsing of Natural Language*, 1974.
No. 7, R. Grishman (ed.), *Directions in Artificial Intelligence: Natural Language Processing*, 1975.
No. 8, R. Grishman, *A Survey of Syntactic Analysis Procedures for Natural Language*, 1975.

IBM Watson Research Center Research Reports. No. 4396, W. J. Plath, *Transformational Grammar and Transformational Parsing in the REQUEST System*, 1973.
No. 4457, S. R. Petrick, *Semantic Interpretation in the REQUEST System*, 1973.

[1] This list is representative but not complete.

Linguistic String Program Reports. Nos. 1-11. (These document the work of New York University Linguistic String Project from 1965 to the present.)

Stanford Research Institute Annual Technical Reports. Cf. D. Walker et al., Speech Understanding Research, 1975.

Transformations and Discourse Analysis Papers, University of Pennsylvania.
Nos. 15-19, The 1959 English Syntactic Analyzer.
Nos. 27, 28, N. Sager, Procedure for Left-to-Right Analysis of Sentence Structure.
Nos. 42, 67, A. K. Joshi and D. Hiz, A Procedure for Transformational Decomposition.
No. 75, A. K. Joshi et al., String Adjunct Grammars and Mathematical Linguistics.

See also reports on computational linguistics at the following institutions:

California Institute of Technology, REL System
Information Sciences Institute, University of Southern California
Massachusetts Institute of Technology, Artificial Intelligence Laboratory
MITRE Corp.
University of California at San Diego
University of Illinois at Urbana
University of Texas at Austin
Yale University, Department of Computer Science

Naomi Sager