Current Issues in Linguistic Theory

The Legacy of Zellig Harris

Language and information into the 21st century

Volume 2: Mathematics and computability of language

Edited by
Bruce E. Nevin
Stephen M. Johnson

AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE

General Editor
E. F. KONRAD KOERNER
(University of Ottawa)

Series IV – CURRENT ISSUES IN LINGUISTIC THEORY

Volume 229

Bruce E. Nevin and Stephen M. Johnson (eds.)

The Legacy of Zellig Harris
Language and information into the 21st century
Volume 2: Computability of language and computer applications

THE LEGACY OF ZELLIG HARRIS

LANGUAGE AND INFORMATION INTO THE 21ST CENTURY

VOLUME 2: COMPUTABILITY OF LANGUAGE AND COMPUTER APPLICATIONS

Edited by

BRUCE E. NEVIN
Cisco Systems, Inc.

STEPHEN M. JOHNSON
Columbia University

JOHN BENJAMINS PUBLISHING COMPANY
AMSTERDAM/PHILADELPHIA

PART 2

Computability of language

CHAPTER 4

The computability of strings, transformations, and sublanguage

Naomi Sager and Ngô Thanh Nhàn
New York University

1. Introduction

Zellig Harris's work in linguistics placed great emphasis on methods of analysis. His theoretical results were the product of prodigious amounts of work on the data of language, in which the economy of description was a major criterion. He kept the introduction of constructs to the minimum necessary to bring together the elements of description into a system. His own role, he said, was simply to be the agent in bringing data in relation to data.

Outsiders could see the genius and great insight into the workings of language that guided the application of rigorous methods of analysis, leading as they did to the formulation of grammatical systems, and ultimately to a penetrating theory of language and information (Harris 1982, Harris 1991). But it was not false modesty that made Harris downplay his particular role in bringing about results, so much as a fundamental belief in the objectivity of the methods employed. Language could only be described in terms of the placings of words next to words. There was nothing else, no external metalanguage. The question was how these placings worked themselves into a vehicle for carrying the 'semantic burden' of language.

Yet Harris's work did not start with a big question and search directly for the answer. His commitment to methods was such that it would be fair to say that the methods were the leader and he the follower. His genius was to see at various crucial points where the methods were leading and to do the analytic work that was necessary to bring them to a new result.

The close relation of Harris's grammatical descriptions to the real data of language invited the possibility of computation, and the close relation of the described structures to the information content of sentences suggested that such computations could lead to the performance by computer of practical informational tasks.

Harris himself had an interest in computation. A number of the procedures that he manually carried out were virtually dry runs of what a computer could be programmed to do. One example is the determination of morpheme boundaries in a phonemically represented utterance by noting peaks in the successive counts of possible next phoneme in utterances that share the same initial segment up to the point of counting (Harris 1968, Section 3.2). On the syntactic level, the cycling-cancellation automaton for sentence well-formedness (Harris 1962) was described in sufficient detail so that it could be implemented from its description and used to analyze medical documents (Shapiro 1967).

2. Linguistic string computation

First, we survey the early computational approaches to syntactic analysis.

2.1 The UNIVAC program

The first computer program to perform syntactic analysis of English sentences was developed by a group under the direction of Harris at the University of Pennsylvania in the period from 1957 to 1959. It ran on the UNIVAC I and successfully analyzed a short scientific text (Harris 1959).

The algorithm of the UNIVAC program incorporated the major constructions of English grammar in considerable detail. While the dictionary was small, lexically ambiguous words were multiply classified (i.e. assigned category symbols corresponding to their different parts of speech, e.g. *walk* noun *N* and verb *V*), with provision in the algorithm for recognizing these as potential sources of alternative analyses. Idioms were included, with provision in the algorithm for certain permitted interruptions in the textual occurrence of the idiom.

The UNIVAC sentence analyzer was not a toy program, nor was it specifically tailored for the sample text. Its generality was demonstrated again 40 years later when the program was reconstituted at the University of Pennsylvania and shown to be effective in computing sentence structure (Joshi & Hopely 1997). It was noted in published comments on this reconstruction that "many of the currently popular techniques for robust parsing are already present, fully articulated in the 1959 UNIVAC parser from UPenn" (Karttunen 1997).

The 1959 UNIVAC program used a grammatical formulation that was termed at the time 'substring analysis'. This was later generalized to 'axiomatic string theory', described along with a brief summary of the UNIVAC program in (Harris 1962a).

2.2 The NYU linguistic string program

A parsing program based on linguistic string analysis, with subsequent extensions to perform transformational and sublanguage analysis, underwent continuous development at New York University from 1965 to 1998. The system came to be known as the LSP (Linguistic String Project) system. The remainder of this chapter summarizes some of the experience of this effort.

The LSP parsing algorithm and grammar grew out of an attempt to solve a problem left over from the 1959 UNIVAC program, namely, how to obtain not just one valid analysis of a sentence, but all possible analyses consistent with the grammar embodied in the program, i.e. how to treat syntactic ambiguity.

The UNIVAC program performed multiple scans of the sentence, recognizing first the 'first order strings' such as noun phrases and prepositional phrases, then the 'second order strings' or 'verb-containing strings' of which the first order strings could be elements. The program left markers at points where decisions were made among alternative lexical categories or alternative ways of continuing the substring analysis.

After some study of how a changed decision at a point of ambiguity affected further processing, several conclusions could be drawn:

- Greater clarity regarding grammatical alternatives would result from separating the grammar from the analysis procedure.
- The elimination of levels in the definition of substrings ('first order' and 'second order') used in different stages of processing would make it easier to correlate a choice made at one point with a dependent choice made at another point.
- The definition of strings as composed solely of category symbols, and the definition of substring relations solely in terms of the possibilities of inserting given types of substrings into other substrings, would make possible a single left-to-right analysis procedure and allow for keeping track of decisions in an orderly way.

In particular the observation which led to the LSP algorithm was that if the grammar was constituted, as above, of elementary strings composed of category symbols, grouped into classes according to the points in other strings at which they have permission to occur, then as one proceeded from left to right through the sentence representation (the sequence of category symbols corresponding to the words of the sentence), each successive word's category symbol (one or more) was either the continuation of a string already begun in the analysis or the beginning of a string permitted to occur at that point. Whenever a category symbol of the current sentence word matched more than one category symbol of the grammar, an alternative analysis path through the sentence would be opened. Keeping track of the opening and closing of paths could be done in various ways (Sager 1960, 1967).

2.3 Implementation of the LSP string parser

The approach taken in the first implementation of the 1960 single-scan left-to-right procedure was to develop a fairly general, language-independent processor, with the grammar definitions and input sentences represented as list structures (Morris 1965). The parse procedure was top down, syntax driven, keeping the analysis in the form of a tree, with ability to back up and obtain another analysis when a branch failed or when the end of the sentence was reached with a successful parse. To apply linguistic constraints to the parse tree, the grammar writer called upon operators for navigating the tree and performing logical operations, and procedures for applying the tests (called 'restrictions') to the parse tree nodes or the sentence representation. On encountering a conjunction, the parser dynamically generated coordinate conjunction strings. As candidate definitions, it used copies of those that were used to analyze the immediately preceding words as properly nested string occurrences. Subsequent implementations have followed a similar approach. A computer grammar of English was written in this style (Sager 1981). The grammar was also adapted to process medical documents in French (Nhàn 1989), German (Oliver 1992), and Dutch (Spyns et al. 1996).

A parse tree obtained in the above manner is not transparently a linguistic string analysis. For one thing, the points of optional string insertion, before or after particular category symbols (e.g. before *N*, after *V*, or at stated inter-element points) become elements of the grammar definitions (Figure 1), hence are seen as nodes in the record of the analysis in the form of a parse tree (Figure 2). Thus, the position to the left of *N* at which a left adjunct of *N* has permis-

<ASSERTION>	::=	<SA> <SUBJECT> <SA> <TENSE> <SA> <VERB> <SA> <OBJECT> <SA-LAST> .
<LNR>	::=	<LN> <NVAR> <RN> .
<SA>	::=	<*NULL> | <SAOPTS> <SA> .
<SAOPTS>	::=	<PDATE> | <SUB11> | <SUB9> | <SUB12> | <SUB0> | <PN> | <PD> | <LDR> | <VENPASS> | <VINGO> | <NSTGT> | <RNSUBJ> | <RSUBJ> | <SUB5> | <SUB1> | <SUB2> | <SUB3> | <SUB8> | <TOVO> | <PVINGO> | <PWHERES>.

Grammar definitions are written in Backus Naur Form (BNF):
<X> <Y> means <X> AND <Y>; <X> | <Y> means <X> OR <Y>.

SA	sentence adjunct position;
SAOPTS	options of SA (modifiers of entire ASSERTION);
SA-LAST	options of SA in string-final position;
LNR	noun N with left and right adjuncts;
LN	left adjuncts of N;
NVAR	local variants of N;
RN	right adjuncts of N;
PDATE	preposition + date-form;
SUBn	subordinate conjunction strings;
PN	prepositional phrase;
PD	preposition + locative LDR;
LDR	adverb with left and right adjuncts;
VENPASS	passive string;
VINGO	gerund string;
NSTGT	noun string of time;
RNSUBJ	post-object adjuncts of subject N;
RSUBJ	roving adjuncts of subject;
TOVO	infinitive string;
PVINGO	Preposition + VINGO;
PWHERES	Preposition + WHERE string.

Figure 1. Definitions in the LSP string grammar

sion to occur is seen as a node of the tree, *LN*, whose value may be a string of the type 'left adjunct of *N*'. Similarly, the position of sentence adjunct occurrence, before or after any element of an elementary verb-containing string, appears as a node SA in the parse tree. In Figure 2, the first SA node represents the position where the sentence adjunct *Today* had permission to insert itself into the elementary assertion string *she has cough.*

In the example parse tree in Figure 2, only non-empty elements of the grammar definitions are shown, except for the ordered adjunct positions of LN:

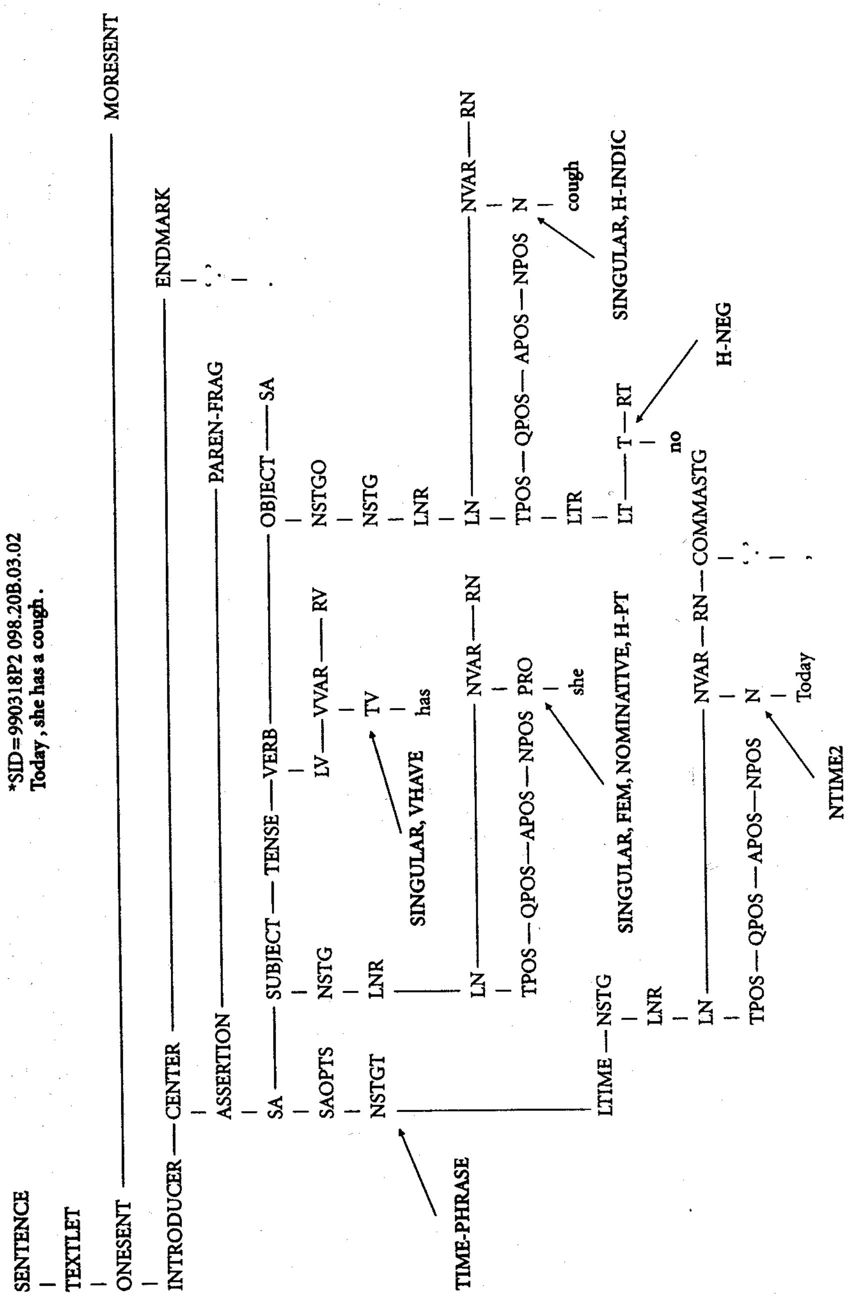


Figure 2. Linguistic string parse tree

TPOS	article position
QPOS	quantity position
APOS	adjective position
NPOS	left compound noun position

VERB is classed as an LXR-type node. English lexical attributes are SINGULAR, NOMINATIVE, FEM, NTIME2, and VHAVE. TIME-PHRASE is a computed node attribute, and H-PT and H-NEG are Healthcare-sublanguage lexical attributes.

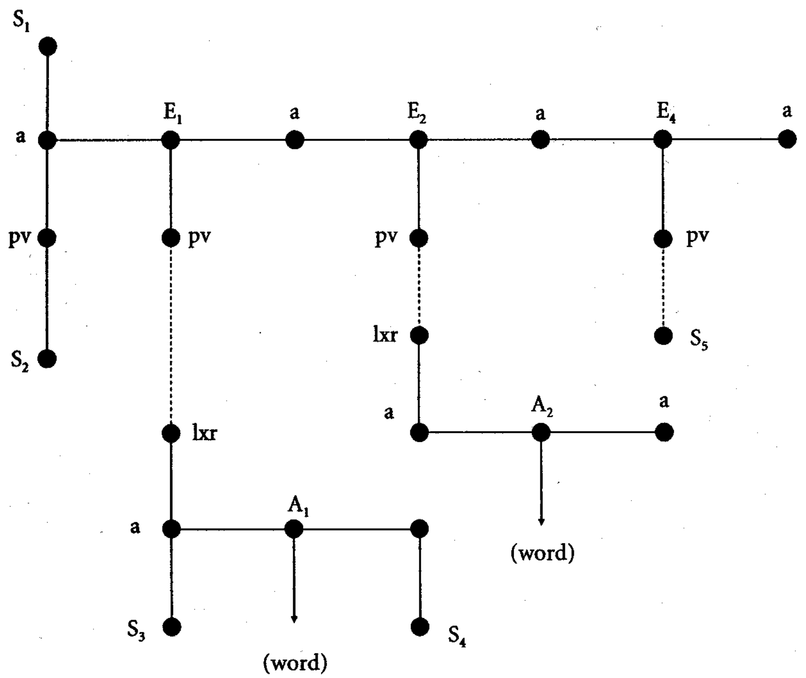
The definition of optional insertion points as elements of the computer representation may seem like a simple accommodation for efficiency of implementation, but it masks the linguistic string character of the underlying grammar by giving the same form to a linguistic relation as to a position of word occurrence in the sentence. Thus, the linguistic string parse tree looks like a tree formed by an immediate constituent grammar, but in essential respects it is not.

String analysis is better suited for computation than immediate constituent analysis is. One of the reasons is that linguistic constraints, whether grammatical (e.g. number agreement of subject and verb) or selectional (e.g. semantic compatability of a noun and its modifier), apply only to words occurring as coelements within an elementary string or as elements of strings related by string adjunction. (There is also a special case of noun replacement that accounts for subject and object strings.) Computationally, this means that the arguments of a test that is to realize a linguistic constraint (the words to be tested) can always be located in the parse tree based on their string relation. In an immediate constituent parse tree, it is not as straightforward to point to words that have a co-dependence.

To retain this advantage of string analysis for computation, grammar definitions in the form used by the parser (in later implementations written in Backus Naur Form, BNF, as seen in Figure 1) are divided into types according to their role in representing string grammar. The type *STRING* covers all definitions corresponding to the elementary strings of axiomatic string theory. In the BNF representation, the *STRING* elements are usually not category symbols but named sets of positional variants that terminate in the category symbols of elementary strings, as discussed above.

The type *LXR* covers all definitions consisting of a category symbol *X* preceded by the set of its left adjuncts *LX* and followed by the set of its right adjuncts *RX*, where *LX* and *RX* each has a null option to express the optionality of adjunct occurrence. The 'core' of an *LXR* node in the parse tree is uniformly its central category symbol *X* (or the value of *XVAR*, its local variants), which is also the core of the string element that lies above it in the parse tree and of any intermediate positional variants. Thus, in Figure 2, *PRO* (*she*) is the core of *LNR* under positional variant *NSTG* under the element *SUBJECT* of the string *ASSERTION*. In *ASSERTION*, the elementary string *N tV N* (*she has cough*) is the sequence of the cores of the elements *SUBJECT*, *VERB*, *OBJECT*.

Because navigation routines (*CORE*, *ELEMENT*, *COELEMENT*, etc.) are written in terms of the definition types (node types in the parse tree, Figure 3), they can locate the arguments of restrictions as though they existed in a simpler tree composed solely of string-related category symbols.



Node types

S	string	pv	positional variant
a	adjunct set	lxr	atom x with its adjuncts
E	element	A	atomic node

Figure 3. Generalized linguistic string parse tree

Over the years significant features have been added to the string parser. These include:

- The *Restriction Language*, a programming language for stating the restrictions on the parse tree (Sager & Grishman 1975).
- Procedures for checking the semantic well-formedness of the parse tree in terms of the co-occurrence of word subclasses in particular syntactic relations — for example, to check whether the words occurring as the noun-preposition-noun (*N P N*) relation are in compatible sublanguage word classes for the given subject area.
- Procedures for rearranging and augmenting the parse tree in accordance with established linguistic transformations — for example, to expand conjunction constructions.
- Mappings to different forms of output depending on further regularities observed in the data or the needs of particular applications.

2.4 From strings to transformations

Harris recognized the validity of different methods of analysis. In Section 1.4 of (Harris 1962a:18-19), he compared string analysis to transformational analysis, and these in turn to immediate constituent analysis:

> If we consider all three types of analyses, we note first that string analysis is intermediate between the other two: It isolates one elementary sentence out of each sentence; constituent analysis isolates no sentence; while transformational analysis reduces the whole sentence to elementary sentences (with primitive adjuncts) and constants [...].
>
> Nor does the difference lie in the power of the three to characterize different sets of sentences [...]. For each of these types of analysis can describe all the sentences of a language (though at very different cost in complexity of the description) [...].
>
> The difference is rather in how the three analyses interrelate the sentences and sentence-segments of the language: For each characterization of a sentence relates that sentence to its decomposition products and also to other sentences having a similar decomposition. Thus, constituent analysis shows to what extent the sentences can all be viewed as sequences of two constituents, subject and predicate, with sentence adjuncts deployed around them. String analysis relates all sentences having the same elementary sentence, the same adjuncts, etc. Transformational analysis goes far beyond either in bringing together the sentences which we feel should be brought together. Thus it relates *He is slow in learning* with *He learns slowly*; and *He began to speak* with *He spoke*; and *He seems young* with *He is young*; and *whom I saw* adjoining *man* with *I saw the man*; whereas neither

> constituent analysis nor string analysis shows direct relation between the members of each pair [. . .].
>
> Nevertheless, though transformational analysis is the most refined, all three analyses are relevant, for language has the properties of all three.

A transformational analysis of a sentence is more refined than other grammatical analyses in one respect in particular: it is closer to an informational decomposition of the sentence. It displays the component individual statements that were combined into one larger informational package, the sentence. This suggested strongly that the path from string parsing to informational applications would lead through transformations.

It was clear from the start that linguistic strings were closely related to transformations. The sentence forms of the transformational kernel set were virtually the same as the elementary center strings of linguistic string analysis, and many of the elementary adjunct strings could be described as 'deformed' elementary sentences, e.g. the adjective left adjunct of the noun in *A N* could be said to be a 'deformation' of *N is A* obtained by dropping the *is* and permuting *A* to before *N*. Thus, many linguistic strings can be seen as the form an elementary sentence takes as a result of an information-preserving form change that makes it available to be a component of a larger sentence.

3. Transformational computation

Transformational analysis brought with it new challenges for computation.

3.1 Initial considerations

As transformational analysis evolved from a relation among sentence forms to a theory of grammar (Harris 1968, Ch. 4), it was possible to base transformational computation on one or another of its formulations. Because a transformational decomposition of a sentence makes explicit how every element of meaning enters the sentence and the changes of form this entails, there was interest in finding a formal (and computable) representation of this process. Harris provided a representation in the form of Decomposition Lattices (Harris 1967, also Harris 1970), in which each node corresponds to a transformational operation and the lattice displays the order of their operation.

The requirements for an implementation of a decomposition-lattice analysis of sentences are formidable. A large number of detailed transformations must be formulated and formalized; a correspondingly detailed lexicon must be developed, in which derivational affixes are treated (e.g. the *ly* in *slowly* in Harris's example above: *He is slow in learning* ↔ *He learns slowly*). Unfortunately these imposing requirements have prevented such a computer program from being developed.

Without going so far as to do a complete transformational decomposition, it is possible to use the transformational relations among sentence forms to bring into alignment such segments as carry the same or similar information, somewhat in the spirit of transformations as a tool for discourse analysis (Harris 1952, Harris 1963). For example, in one form of output of the LSP system, mapping the output to a relational database, transformations are used implicitly by placing in the same column the words that would have been aligned linguistically by transformations. Thus, *she broke her ankle, broken ankle, ankle break, a break in the ankle bone,* will all have *ankle* in a column of the database table labeled *BODYPART,* and *broke, broken, break* in a column labeled *SYMPTOM,* without having rearranged the parse tree in accord with the applicable linguistic transformations.

3.2 Implementation of transformations in the LSP system

Some transformations in the LSP system are implemented as changes to the parse tree and some transformations are utilized rather than implemented. One example of the latter was given above. For another example, the passive transformation N_2 *is Ven by* $N_1 \leftrightarrow N_1\ tV\ N_2$ need not be executed on the parse tree in order for selectional (word choice) compatibility in a passive construction to be checked, based on a list of acceptable subject-verb-object patterns stated for $N_1\ tV\ N_2$. Similarly, it is not necessary to reconstruct *N is A* from an *A N* occurrence in a sentence in order to check the compatibility of the adjective and noun in this relation. There is some advantage in retaining the original word order of the sentence unless the goals for the representation or the application require the rearrangement of sentence parts.

The transformations that change the parse tree primarily serve to obtain complete, or relatively complete, informational units of the *ASSERTION* type from the more diverse adjunctive and conjunctive forms in the original sentence. Coordinate conjunction constructions are expanded up to the *ASSERTION* level, i.e. the 'understood' or 'zeroed' elements are copied from

Original sentence from an anonymized patient document:

Today, she has no cough, chest pain, or shortness of breath.

is transformed into single information units:

Today, she has no cough
, *today, she has no* chest pain,
and today, she has no shortness of breath.

where:
- Time word *today* is distributed to the basic statements;
- Negative word *no* is distributed to every object;
- *or* in distributed negative statement is transformed to *and.*

Figure 4. Transformation of a parsed sentence into information units

the full form into the reduced form in the positions dictated by parallel construction. Thus, *Extremities revealed clubbing and cyanosis* becomes *Extremities revealed clubbing and [extremities revealed] cyanosis.* In the case of a construction of the type *NO X OR Y*, the negative is distributed and the conjunction is changed to *AND*: *Extremities revealed no clubbing or cyanosis* ↔ *Extremities revealed no clubbing and [extremities revealed] no cyanosis.* The antecedent of the bound pronoun in a *WH* construction likewise is supplied: *A peripheral neuropathy workup was initiated which revealed normal folate levels* ↔ *A peripheral neuropathy workup was initiated which [workup] revealed normal folate levels.* Other modifiers are similarly expanded with the goal of obtaining elementary *ASSERTION* units that are informationally relatively complete. Figure 4 illustrates the expansion process.

4. Sublanguage computation

This application of transformational analysis computationally to texts led naturally to a more detailed consideration of domain-specific word relations, i.e. to sublanguage grammar.

4.1 The sublanguage method

Natural Language Processing (NLP), so named to distinguish it from the processing of computer languages, needs to arrive at a representation of the

content of texts in order to provide further procedures that depend upon it, such as information extraction and word-pair indexing. Some attempts have been made to move directly to semantic characterization without syntactic analysis, and even for those who believe that syntax is part of the information there is a recognition that there is another part. The particular words that occur in the given syntactic relations are what convey the specific information.

Linguists had not been unaware of the role that word choice plays in language. Leonard Bloomfield discussed this as the phenomenon of 'selection' (Bloomfield 1933:164–169, 190–199, 229–237). However, no rules could be imposed as to which word choices make acceptable as opposed to unacceptable sentences. The flexibility of language that enables it to accommodate nonsense, fairy tales, untruths, and so on, leaves it to the speaker to choose whichever words seem suitable as long as they are assembled into an understandably grammatical sentence.

It was Harris's work that first brought word choice into the realm of grammar, albeit in this case as a criterion, not a rule: the definition of the transformational relation between sentence forms, based on the similarity (on some scale) of the acceptability of the word choices in the candidate forms. However, when Harris introduced the notion of sublanguage grammar, particularly with regard to science sublanguages (Harris 1968, Section 5.9.1), the door was opened to extending the rules of grammar into the realm of selection. In a science sublanguage, some types of sentences are possible while others are simply outside the subject area or are such combinations of sublanguage words as are simply not sayable within the science. To use Harris's example (1968:152), in the language of biochemistry, contrast the possible (1) *The polypeptides were washed in hydrochloric acid*, with (2) *Hydrochloric acid was washed in polypeptides*, which if it ever occurred would not be in the discourse of biochemistry.

What was immediately appealing about sublanguage was its methodology. Word classes of semantic specificity could be established objectively based on their sublanguage co-occurrence properties, and in terms of these word classes, sublanguage statement types could be defined to serve as templates to house the information in sublanguage texts.

Experimentally, it was possible to show that the semantically relevant word classes of a particular biomedical sublanguage could be established on purely distributional grounds, using a clustering program (Hirschman *et al.* 1975). In the same vein more recently, a computer program (named ZELLIG in honor of its co-occurrence basis) was developed to obtain semantic classes for French medical documents by applying distributional criteria to noun phrases in parsed documents. The classes obtained corresponded well to the major term types of an established medical terminology (Nazarenko *et al.* 2001).

Frequently co-occurring sublanguage word classes in particular syntactic relations lead to the formulation of a very large array of detailed sublanguage statement types that can be grouped for convenience in different ways. Harris et al. (1989) developed a formulaic representation of the sentence types in a science sublanguage. The purpose of the work, as stated by Harris in the Foreword, was

> [. . .] to develop a formal tool for the analysis of science, and more generally of information [. . .]. In respect to the history of science, the formulaic representation of research done over a period shows, for example, changes in the way words for the objects of the science co-occur with words for the processes, changes which exhibit the actual development of the science.

Another form for grouping related statement types was termed an 'information format' (Sager 1978). This form proved convenient for computer operation on the data. As applied to clinical documents (the Healthcare sublanguage), statement types with a common feature (e.g. the occurrence of a treatment-type word class, or a laboratory-type word class) were combined into one information format that covered the occurrence of all the statement types of that class (Sager *et al.* 1987).

Since the concept of sublanguage was introduced, it has proved especially fruitful in language computation, as attested by chapters in this volume and other publications (e.g. Kittredge & Lehrberger 1982, Grishman & Kittredge 1986, Marsh & Friedman 1985).

4.2 A medical sublanguage

Illustrations of sublanguage computation will be drawn from the LSP treatment of the sublanguage of clinical reporting, i.e. narrative accounts of patients' conditions and treatments as recorded primarily in hospital discharge summaries and visit reports. Reports have been drawn from the areas of Cardiology, Restricted Airways Disease (RAD, mainly, asthma), Rheumatoid Arthritis, Epilepsy, Sickle Cell Disease, Orthopedics, and to a lesser degree from a variety of other specialties. There has been some experience with other

types of documents, such as imaging reports, pathology reports, and surgical reports, each of which employs some specialized vocabulary and usages related to the techniques employed. The French experience was with texts in Digestive Surgery. Portions of a patient visit report are shown in Figure 5.

4.2.1 *Syntax of the healthcare sublanguage*

The first thing that strikes one about most free text clinical documents (once they are typed or otherwise made legible) is their seemingly wild departures from normal syntax. Some 'sentences' are series of noun phrases and other forms, punctuated only by commas. Others are grammatical but endlessly long, as though stopping to form a new sentence would compromise the information. Single-word sentences are not uncommon, where all the words that make the one word into a statement are understood.

HISTORY DIAGNOSIS: Stage I left breast cancer, diagnosed February 19xx.

INTERVAL HISTORY: Ms. XXX returns for her semi-annual visit approximately one month earlier than scheduled. In the last week, she has had tenderness in the mid to lower right axilla as well as in 2 or 3 spots in her right breast including laterally at about the 9:00 position and inferiorly along the inframammary fold. She has not been able to palpate any specific lumps in these areas although she thought she could at 1 point feel a lymph node in the underarm.

On review of systems, the patient has hip pain which is from degenerative joint disease. She under the care of Dr. YYY of ZZZ Dept. of Orthopaedics. She is also recently recovering from a upper respiratory infection felt to be bronchitis. She is taking the last day of an Azithromycin long-acting schedule. She has had improvement in symptoms in the last 1-2 days.

REVIEW OF SYSTEMS: She denies headaches or visual symptoms. Today, she has no cough, chest pain, or shortness of breath.

PHYSICAL EXAMINATION:
Vitals: weight 58.2 stable, pulse 98, BP 131 / 73, temp 36.4, resp 16 unlabored.
The patient appears well.
HEENT: Head atraumatic and normocephalic.
Fundi: benign.
Mouth and throat: clear.
Neck: supple
. . .

Figure 5. Portions of a patient visit report

Table 1. Shortened sentence forms in the healthcare sublanguage

[N V] N	Stiff neck and fever
N [*be*] A | Ving | Ven	Brain scan negative
	Patient complaining of increased breathlessness
	No growth seen
[N *be*] A | Ving	Positive for heart disease and diabetes
	Feeling better
[N] tV O	Has Paget's disease
[N *be*] Ven O	Treated for meningitis
[N *be*] to V O	To be followed in Pain Clinic
[N *be*] P N	On folic acid

The key to this lack of grammaticality is to realize that in most cases what is observed is the residue of a properly formed sentence after all words that would be obvious to another clinician are dropped, or rather are still present but reduced to zero form ('zeroed' in Harris's term). Sometimes this relies on an understood *the patient*, the default subject of all manner of clinical observations (*Fever.* ↔ *Patient has fever.*). It is interesting that for the most part the reduced sentence forms (Table 1) are strings that occur otherwise in English string grammar, similarly also often involving the zeroing of the verb *be*. For example, compare *Brain scan negative*, in Table 1, with *They pronounced the brain scan negative*, in which the same shortened sentence form occurs grammatically as an object string.

Thus, it is possible to write a grammar of the ungrammatical, by observing that the departures from grammaticality are not arbitrary, but follow patterns of reduction that are for the most part already familiar. The BNF part of the Healthcare sublanguage grammar contains a definition *FRAGMENT* whose options are definitions that also occur in the English computer grammar on which the sublanguage grammar is based.

4.2.2 *Word classes of the healthcare sublanguage*

Word classes of the Healthcare sublanguage have been developed manually, first by studying texts for patterned occurrences, then by defining diagnostic frames for further classification of vocabulary (Sager *et al.* 1987). The word classes in current use are listed with examples in Table 2.

Table 2. Word classes of the healthcare sublanguage

Medical Classes	Description	Examples in English and French
*** PATIENT AREA		
H-PT	references to patient	*she, candidate, Mrs. XXX, patient*
H-PTAREA	anatomical area	*edge, left, surface, rebord, gauche*
H-PTDESCR	patient description	*American, homeless, works*
H-PTFUNC	physiological function	*BP, auditory, appetite, tonalité*
H-PTLOC	location relation	*branching, radiating, localisé*
H-PTMEAS	anatomical measure	*height, bulk, depth, corpulence*
H-PTPART	body part	*arm, adrenal, carotid, liver, foie*
H-PTPALP	palpated body part	*abdomen, liver, foie*
H-PTSPEC	specimen from patient	*sample, scraping, frozen section*
H-PTVERB	verb with patient subj	*complain, endure, suffer*
*** TEST / EXAM AREA		
H-TXCLIN	clinical exam procedure	*Babinski, palpation, auscultation*
H-TXPROC	diagnostic procedure	*MRI, xray, ultrasound,*
H-TXSPEC	test of specimen	*CBC, immunoassay, urinanalysis*
H-TXVAR	test variable	*Iodide, iron, glucose, GB*
*** TREATMENT AREA		
H-TTGEN	general medical mgmt	*follow-up, admit, discharge, soins*
H-TTMED	treatment by medication	*aspirin, clamoxyl*
H-TTSURG	surgical interventions	*excise, hysterectomy*
H-TTCOMP	complementary therapy	*bedrest, repos, physiothérapie*
*** RESULT AREA		
H-AMT	amount or degree	*much, partly, total, sévère*
H-DESCR	neutral descriptor	*amber, amorphous, amphoric,*
H-DIAG	diagnosis	*asthma, diabetes mellitus*
H-INDIC	disease indicator word	*fever, swelling, pain, thrombose*
H-NORMAL	non-problematical	*within normal limits, bon état*
H-ORG	organism	*renibacterium, rickettsial, rod*
H-TXRES	test/exam result word	*gram-negative, positive, positif*
H-RESP	patient response	*better, improve, relief*
H-CHANGE-MORE	quantity increase	*peak, rise, increase, spikes*
H-CHANGE-LESS	quantity decrease	*lower, recede, reduce, taper*
H-CHANGE-SAME	quantity constant	*keep, remain, same, maintain*
H-CHANGE	indication of change	*alteration, changing, drift, modify*
*** EVIDENTIAL AREA		
H-NEG	negation of finding	*no, not, cannot, denied, ne pas*
H-MODAL	uncertainty of finding	*probable, seems, suspicion*

(Table 2 *cont.*)

Medical Classes	Description	Examples in English and French
*** CARE ENVIRONMENT AREA		
H-FAMILY	family, friends, . . .	*she, sister, father, boy friend*
H-INST	doctors, institutions, . . .	*Dr. XXX, hospital, social service*
*** TIME AREA		
H-TMBEG	beginning	*onset, new, dévelopé, apparition*
H-TMEND	termination	*end, terminal, discontinue, arrêt*
H-TMLOC	location in time	*current, previous, actuelle, post-op*
H-TMPER	duration	*brief, persistent, constant*
H-TMREP	repetition	*frequently, intermittent, habituelle*
H-TMPREP	time preposition	*during, after, since, après, depuis*
*** CONNECTIVE AREA		
H-BECONN	classifier verb	*is (a), represent, resemble, est (un)*
H-CONN	connects 2 IF's	*due to, along with, secondaire à*
H-SHOW	connects test & result	*shows, confirmed, notable for*

One might ask why manual as opposed to automatic methods of sublanguage word class generation have been used. For one reason, the frequently occurring reduced sentence forms in this sublanguage deprive the automatic procedure of the explicit syntactic relations upon which the clustering depends. In addition, in complete sentences, syntactic ambiguity can muddy the results. Without constraints that utilize the very sublanguage classes one is trying to generate, too many syntactically valid parse trees are generated for clear co-occurrence patterns to emerge. Bootstrapping approaches could probably be developed.

Another group of reasons is special to the medical domain and the intended applications. There is a very large special vocabulary of medicine. Co-occurence analysis of free text input could probably determine major classes, but it is far more efficient to use medical dictionaries and text elicited from experts. Morphological analysis of the Latinate medical vocabulary can result in automatic classification of many medical words, and this has been done (Wolff 1984). Finally, the quality of the output of language processing depends crucially on the quality of the dictionary used in the processing. The standards for the delivery of information to support healthcare are particularly high so that whatever human or computer means are used to create the dictionary, human quality control is an absolute necessity.

To build the Healthcare sublanguage dictionary, words are coded for their syntactic properties according to the scheme described in Appendix 3 of (Sager 1981), to which are added the appropriate medical classes as additional attributes, relying in large part on the contexts in which the words occur. The English Healthcare dictionary currently numbers about 51,000 words, supplemented by lists derived from published sources, e.g. drug lists.

4.2.3 *Creation of new connectives*

Harris envisioned that sublanguage analysis would stimulate the definition of new connectives.

> Even the small classes that fill the role of transformational constants, such as prepositions and conjunctions, which have always been considered to be unextendable objects in grammar, can receive new members in particular subsets of sentences, thus increasing the grammar for these sentences. The creation of new members of prepositions *P* and conjunctions *C* is possible because certain grammatical sequences of morphemes have the same neighbors within a sentence form as do *P* or *C*. (Harris 1968: Section 5.9.2)

In the course of developing the Healthcare sublanguage grammar and applying it to texts, the issue of what constituted an information unit arose. Some prepositions (e.g. *with*) when occurring between two nouns of the same predicate-type subclass (e.g. H-INDIC) could be seen as a connective between two reduced-sentence-form units of information, e.g. *headache with fever* similar to *headache and fever*. Extending this process, similar sublanguage environments became the criterion for defining many new idiom prepositions and some new subordinate conjunctions. A partial list of idiom prepositions in the Healthcare sublanguage dictionary is shown in Table 3.

4.3 Healthcare sublanguage processing

The overall sequence of procedures in the Medical Language Processor, or MLP, as it has come to be called, is shown in Figure 6. In practice, the processing of clinical documents requires a number of preliminary procedures, which are not specifically linguistic in character but are necessary if the documents are to be parsed. Examples include recognizing names, determining section heads, finding sentence boundaries, treating abbreviations, and normalizing number, date, and unit formats. These and other operations are combined into a preprocessing stage. After preprocessing, every sentence carries a sentence identifier (SID), which locates it as an element of a document set, a

Table 3. Idiom prepositions in the healthcare sublanguage

accompanied by	free of	prior to
according to	halfway up	regardless of
accounting for	improved by	relieved by
aggravated by	in absence of	remarkable for
akin to	in anticipation of	resulting from
along with	in association with	resulting in
alternating between	in between	s / p
alternating with	in competition with	secondary to
apart from	in contrast to	significant for
as a consequence of	in light of	similar to
as a result of	in regard to	situated in
as distinct from	in spite of	situated on
as exemplified by	in terms of	specific for
as part of	in the absence of	status post
associated with	in the course of	subsequent to
at the time of	in view of	such as
because of	inconsistent with	suggestive of
bounded by	independent of	suspicious for
characterized as	instead of	suspicious of
characterized by	located in	tolerant of
close to	made worse by	triggered by
compatible with	manifest as	typical of
confined to	mediated by	unassociated with
consistent with	more than	up to
consisting of	notable for	w / o
down to	notable only for	with and without
due to	on basis of	with involvement of
evolving to	on the basis of	with regards to
except for	on top of	with respect to
exemplified by	other than	without evidence of
followed by	out of	worsened by
free from	precipitated by	

particular document, a section of the document, a paragraph in the document and a sentence in the paragraph.

MLP dictionaries include the basic Healthcare sublanguage dictionary described above, along with outside sources and special subarea dictionaries that add special terms and alternative definitions in case of conflict. The parsing engine provides for dictionary lookup to obtain the parts of speech and syntactic and sublanguage attributes of document words, calls on the parsing grammar to obtain the syntactic analysis of the sentence, and applies

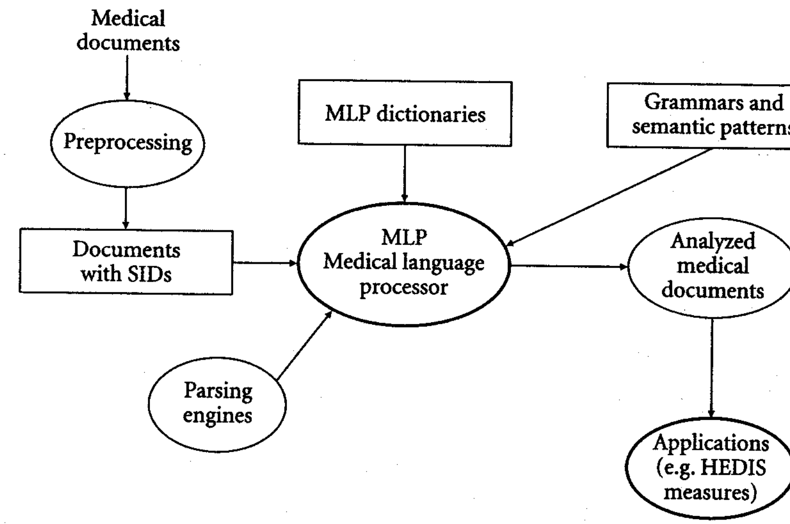


Figure 6. The MLP system overview

the sublanguage (semantic) patterns to resolve syntactic and lexical ambiguity. It then applies the transformational grammar and the information formatting procedures, after which the output can be mapped into the desired form. An overview of the MLP system is given in (Sager *et al.* 1994).

4.3.1 *Sublanguage constraints in parsing*

A parsing grammar that contains most of the constructions found in English sentences, plus reduced sentence forms, is very likely to produce multiple analyses of an input string. To constrain the number of analyses and, hopefully, arrive at the intended one, the grammar must be further restricted, and this is the primary role of sublanguage in parsing. Some of the more interesting situations are noted here.

Conjunctional equivalence

For the MLP to end up with correctly segmented and characterized information units, it is important that coordinate conjunction strings be composed of 'like' elements, not any parsable *N CONJ N*. In sublanguage terms, the conjoined *N*s should be in the same or similarly occurring sublanguage classes, e.g. all H-PTPART words, or an H-INDIC word with an H-DIAG word. This problem can arise even in a straightforward medical sentence, such as

The concurrent weight loss raises a concern in regard to malignancy of the stomach, pancreas, colon, and female organs.

Structural definitions (the BNF component) in the Healthcare sublanguage grammar would generate (among others) a parse showing *malignancy* and *pancreas* conjoined. Compare the syntactically similar sentence in which *malignancy* and *ulcer* are conjoined:

The concurrent weight loss raises a concern in regard to malignancy of the stomach, or benign gastric ulcer.

To prevent inappropriate conjoinings, the Healthcare sublanguage grammar contains lists of subclasses that are compatible in conjunction constructions. For example, two sublists of the list CONJ-EQUIV-CLASSES from the grammar are:

(H-TTSURG, H-TXCLIN [*refused surgery or workup*]),
(H-TTSURG, H-INDIC, H-DIAG [*Past medical history includes hypertension, left hip arthroplasty and Perth's disease*]),

A restriction checks conjuncts using these lists. If the test fails, it is likely that conjoining will succeed if the conjunct is detached from its current position in the parse tree and re-attached to another available host.

Computed attributes
When sublanguage conjunction constraints are applied, it becomes apparent that testing core *N*s is not always effective, because in some contexts it is the semantic value of the *N + adjunct* that enters into conjunction equivalency. For example, in *fatigue and swollen ankles* the subclasses H-INDIC (*fatigue*) and H-PTPART (*ankles*) are not in a CONJ-EQUIV-CLASS sublist, but if we allow the *N + LN* (*swollen ankles*) to take on the 'computed attribute' H-INDIC (from *swollen*), then the conjoining will be approved.

In applying the conjunction equivalency test, numerous situations have to be accounted for. For example, in *Fatigue and swollen ankles and knees*, the implicit computed attribute for *swollen knees* must be inferred in order for the triple conjunct to be accepted.

4.3.2 *Selection using sublanguage co-occurrence patterns*
By far the greatest source of syntactic ambiguity is the situation in which an adjunct string can be parsed as adjunct to different candidate hosts, especially in the ubiquitous *N PN PN* sequences. This problem can be compounded by the presence of conjunctions. The approach taken by the LSP has been to collect well-formed patterns of *host + adjunct*, specified with regard to the syntactic relation and the sublanguage word classes that occur correctly in that relation, and to use these authenticated patterns as 'filters' to reject occurrences that do not conform.

For example, in the parse tree for *Rash over abdomen over past week*, the final analysis will show both *PN*s with *P = over* adjoined to *rash* (H-INDIC) in the parse tree, since there is no stored *N + PN* pattern (for *P = over*) corresponding to *abdomen over past week*, i.e. a host *N* of class H-PTPART with a time expression as adjunct. It should noted that 'host *N*' here refers to the core *N* as carrier of node attributes, so that if the core *N* carries a 'computed attribute' it is that attribute that will be used in the filtering test. Thus, *Swollen abdomen over past week* will pass the test, because *abdomen* in this case carries the computed attribute H-INDIC (from *swollen*), which can be adjoined by a time adjunct.

The computed attribute is another instance of employing a transformational relation without carrying out the transformation. In a transformational analysis of the above example, one step would be: *swollen abdomen over past week* ↔ *abdomen was* (or *has been*) *swollen over past week*, where the time phrase adjoins the predicate. Another step might take *swollen* to its verbal source *swell*, where the result would assert that the swelling occurred over the past week.

Several thousands of patterns are stored in a compact notation in 'Selection Lists' that are used in selection restrictions (the filtering tests). Selection patterns are stored for each individual preposition. Some entries from the stored authenticated pattern occurrences for *P = over* are shown in Figure 7.

Selection patterns are also helpful in resolving lexical ambiguity such as occurs when a word has several sublanguage class assignments in the dictionary, e.g. *discharge* H-TTGEN/H-INDIC (*discharge from hospital* vs. *discharge from nose*). There is a stored pattern H-INDIC *from* H-PTPART, but no stored pattern H-INDIC *from* H-INST, so in an occurrence of *discharge from nose*, *discharge* will be stripped of its H-TTGEN class, and *discharge* will be treated by the information formatting procedure as an H-INDIC word.

4.3.3 *Forms of output*
Figure 8 shows the principal output of the information-formatting component of the MLP. This output represents the results of converting the parse tree to a medical representation composed of Information Format (IF) occurrences and connectives. Each IF occurrence corresponds to a statement type of the

SUBLANGUAGE CO-OCCURRENCE TABLE
Approved HOST-P-N for preposition "OVER"

Layout of table:
— Column 1: Pattern name and frequencies [*n:m*],
— *n* : frequency of same exact word cooccurrences in row;
m : frequency of sublanguage class cooccurrences in row.
— Columns 2-3-4: Words and their sublanguage classes;
— Column 5: Sentence ID and source text.

Pattern	HOST	P	N	Sentence ID
HOST-P-N [1:10]	**spiders** N:H-INDIC	over P:OVER	**extremities** N: H-PTPART	*SID=CPRIS 007.01D.01.06 there were very few **spiders over** the upper **extremities** .
HOST-P-N [1:1]	**centered** VEN: H-PTLOC	over P:OVER	**pubis** N: H-PTPART	*SID=991121 098.36F.01.06 she is to return again 11/19/1999 for her six month follow up , with ap pelvis **centered over** the **pubis** , and ap and lateral of the left hip .
HOST-P-N [1:1]	**inversion** N:H-TXRES	over P:OVER	**precordium** N: H-PTPART	*SID=CABG1 051B.1.07 there was t wave **inversion over** the anterior **precordium** and t wave flattening laterally which was new compared to an electrocardiogram done approximately one month earlier .
HOST-P-N [1:1]	**syncope** N:H-INDIC	over P:OVER	**winter** N: H-TMLOC	*SID=CPRIS 006.01E.01.03 due to his rhythm problems , as well as a history of near **syncope over** the **winter** , we will admit him to the hospital for further evaluation of his arrhythmia and the need for possible permanent pacemaker placement .
HOST-P-N [1:2]	**recover** V:H-RESP	over P:OVER	**five to ten minutes** QN: NTIME1	*SID=MGHPT 005A.02.02 at that time , without warning , she would fall and have generalized tonic-clonic movements with accompanying loss of consciousness from which she would **recover over** the next **five to ten minutes** .

Figure 7: Approved selection pattern occurrences


```
****************************************
****************************************
*SID=990318P2 098.20B.03.02
[ HISTORY-OF-PRESENT-ILLNESS ] Today , she has no cough , chest pain , or
shortness of breath .

(CONNECTIVE (CONJOINED (CONN = , <',':()> )))

(PATIENT-STATE-IF
  (PT-DEMOG (GENDER = [FEMALE] <GRAM-NODE:(FEM)> ))
  (SUBJECT = she <PRO:(H-PT)> )
  (VERB = has <TV:(VHAVE)>
    (EVENT-TIME (REF-PT = Today <N:(NTIME2)> , <',':()> ))
    (TENSE = [PRESENT] <GRAM-NODE:(H-VTENSE)> ))
  (PSTATE-DATA
    (S-S = cough <N:(H-INDIC)>
      (MODS (NEG = no <T:(H-NEG)> ))))
  (TEXTPLUS = ))

(CONNECTIVE (CONJOINED (CONN = AND <'AND':()> )))

(PATIENT-STATE-IF
  (PT-DEMOG (GENDER (GENDER = [FEMALE] <GRAM-NODE:(FEM)> ))
  (SUBJECT = = she <PRO:(H-PT)> )
  (VERB = has <TV:(VHAVE)>
    (EVENT-TIME (REF-PT = Today <N:(NTIME2)> , <',':()> ))
    (TENSE = [PRESENT] <GRAM-NODE:(H-VTENSE)> ))
  (PSTATE-SUBJ (PTPART = chest <N:(H-PTPART)> ))
  (PSTATE-DATA
    (S-S = pain <N:(H-INDIC)>
      (MODS (NEG = no <T:(H-NEG)> ))))
  (TEXTPLUS = ))

(PATIENT-STATE-IF
  (PT-DEMOG (GENDER (GENDER = [FEMALE] <GRAM-NODE:(FEM)> ))
  (SUBJECT = = she <PRO:(H-PT)> )
  (VERB = has <TV:(VHAVE)>
    (EVENT-TIME (REF-PT = Today <N:(NTIME2)> , <',':()> ))
    (TENSE = [PRESENT] <GRAM-NODE:(H-VTENSE)> ))
  (PSTATE-DATA (S-S = shortness of breath <N:(H-INDIC)>
    (MODS (NEG = no <T:(H-NEG)> ))))
  (TEXTPLUS = ))
```

Figure 8. Output of the MLP system

sublanguage and constitutes a basic unit of healthcare information.

In the parenthesized information-format tree display in Figure 8, only non-empty elements of the definitions are shown. The node names that are not obvious are:

*SID=	A unique sentence identification number
[HISTORY-OF-PRESENT-ILLNESS]	A section reference
CONNECTIVE	A node that connects two following IFs (Polish notation)
CONJOINED	A type of connective
CONN	A connective word
PATIENT-STATE-IF	An information format type, in this case, Patient-State
PT-DEMOG	Patient demographic information referred to in the sentence
GENDER	The gender of patient
SUBJECT	A grammatical subject (if not otherwise assigned)
VERB	A grammatical verb (if not otherwise assigned)
EVENT-TIME	A chronology modifier of the reported event
REF-PT	A time reference point
TENSE	The tense of the sentence verb
PSTATE-DATA	Data of the patient state
S-S	Signs and symptoms
MODS	Modifiers
NEG	A negative modifier
PTPART	A body part
TEXTPLUS	Words not included in IF

English parts of speech (or generated placeholder GRAM-NODE) and Healtcare-sublanguage lexical attributes are indicated by angle brackets: <GRAM-NODE:(FEM)>, <PRO:(H-PT)>, <TV:(VHAVE)>, <GRAM-NODE:(H-VTENSE>, <N:(NTIME2)>, <N:(H-INDIC)>, <N:(H-PTPART)>, <T:(H-NEG)>. Values generated by the MLP grammar are [PRESENT] (from verb *has*), and [FEMALE] (from pronoun *she*).

Depending on the type of applications, the MLP output is converted from the IF form into a simple table or XML trees, as follows:

– *A simple 2-dimensional table.* Each row corresponds to one IF occurrence and has the following 35 fields: the sentence SID (1 field), the section of the document (1 field), the number of this IF in this sentence (1 field), how it is

connected to other IFs in the same sentence (3 fields), the NIMPH marking for this IF (1 field) (see 4.3.4, below), and a flat layout of the major data points of the IF (remaining fields). For example, the 3 IFs from Figure 8 are presented in 3 rows. The symptom phrases (e.g. *no cough, no pain* and *no shortness of breath*) are housed in the fields Negation (NEG = *no*) and fields Sign-Symptom (S-S = *cough*, S-S = *pain*, and S-S = *shortness of breath*). Studies such as Healthcare Quality Assurance, (5.1 below) were done using the database management systems INGRES and Informix, and web-based HTML (HyperText Markup Language) (Sager *et al.* 1996).

– *XML-trees.* This is another variation of the IF trees (Figure 9), fully equivalent to the ones in Figure 8. XML (eXtensible Markup Language) is a representation formalism which is part of a web-based 'family of technologies' (see W3C:XML 1999). XML promises flexibility in representation and presentation of information. Using XML, the original text after MLP is tagged with lexical and syntactic information. However, this is not just another variation of the IF trees. It is a richer representation where each node is now capable of housing attribute information.

In the XML representation, each node in the IF is represented as one tagged item (opening with '<tag>' and closing with '</tag>'); each unit of lexical information at a terminal node is represented as a triple consisting of one category tag, followed by sublanguage word class tags, followed by the word (where 'word' here stands for the word or phrase at the terminal node). For example, the phrase *no cough* in the IF tree is represented as follows:

```
<S-S>
   <NEG><T><(H-NEG)>no</(H-NEG)></T></NEG>
   <N><(H-INDIC)>cough</(H-INDIC)></N>
</S-S>
```

Here, <S-S>, <N>, etc. are opening tags, and </S-S>, </N>, etc. are closing tags.

Furthermore, it allows an application to extract data by scanning the MLP IF output. For example, the extraction of sign-symptom information in the first XML IF-tree of Figure 9 is accomplished by scanning from left to right and picking up everything between <S-S> and </S-S>, i.e. *no cough*, within the context of one IF, that is, between <PATIENT-STATE-IF> and </PATIENT-STATE-IF>.

This technology allows the designer to embed any number of tags that need not be seen by the user but can direct the retrieval and display of content

```
*SID=990318P2 098.20B.03.02
[ HISTORY-OF-PRESENT-ILLNESS ] Today , she has no cough , chest pain , or shortness of breath .

<CONNECTIVE><CONJOINED>
            <CONN><',>,</',></CONN></CONJOINED></CONNECTIVE>
<PATIENT-STATE-IF>
  <PT-DEMOG><GENDER><GRAM-NODE><(FEM)>
                      [FEMALE]</(FEM)></GRAM-NODE></GENDER></PT-DEMOG>
  <SUBJECT><PRO><(H-PT)>she</(H-PT)></PRO></SUBJECT>
  <VERB><TV><(VHAVE)>has</(VHAVE)></TV>
    <EVENT-TIME><REF-PT>
      <N><(NTIME2)>Today</(NTIME2)></N> <',>,</',></REF-PT></EVENT-TIME>
    <TENSE>
      <GRAM-NODE><(H-VTENSE)>[PRESENT]</(H-VTENSE)></GRAM-NODE>
      </TENSE></VERB>
  <PSTATE-DATA><S-S><N><(H-INDIC)>cough</(H-INDIC)></N>
          <MODS><NEG> <T><(H-NEG)>no</(H-NEG)></T></NEG></MODS>
        </S-S></PSTATE-DATA>
  <TEXTPLUS></TEXTPLUS></PATIENT-STATE-IF>
<CONNECTIVE><CONJOINED>
          <CONN><'AND'>AND</'AND'></CONN></CONJOINED></CONNECTIVE>
<PATIENT-STATE-IF>
  <PT-DEMOG><GENDER><GRAM-NODE><(FEM)>
                      [FEMALE]</(FEM)></GRAM-NODE></GENDER></PT-DEMOG>
  <SUBJECT> <PRO><(H-PT)>she</(H-PT)></PRO></SUBJECT>
  <VERB> <TV><(VHAVE)>has</(VHAVE)></TV>
    <EVENT-TIME><REF-PT>
      <N><(NTIME2)>Today</(NTIME2)></N> <',>,</',></REF-PT></EVENT-TIME>
    <TENSE>
      <GRAM-NODE><(H-VTENSE)>[PRESENT]</(H-VTENSE)></GRAM-NODE>
      </TENSE></VERB>
  <PSTATE-SUBJ><PTPART>
      <N><(H-PTPART)>chest</(H-PTPART)></N></PTPART></PSTATE-SUBJ>
  <PSTATE-DATA><S-S> <N><(H-INDIC)>pain</(H-INDIC)></N>
        <MODS> <NEG><T><(H-NEG)>no</(H-NEG)></T></NEG></MODS>
      </S-S></PSTATE-DATA>
  <TEXTPLUS></TEXTPLUS></PATIENT-STATE-IF>
<PATIENT-STATE-IF>
  <PT-DEMOG><GENDER><GRAM-NODE><(FEM)>
                      [FEMALE]</(FEM)></GRAM-NODE></GENDER></PT-DEMOG>
  <SUBJECT><PRO><(H-PT)>she</(H-PT)></PRO></SUBJECT>
  <VERB> <TV><(VHAVE)>has</(VHAVE)></TV>
    <EVENT-TIME><REF-PT>
      <N><(NTIME2)>Today</(NTIME2)></N> <',>,</',></REF-PT></EVENT-TIME>
    <TENSE>
      <GRAM-NODE><(H-VTENSE)>[PRESENT]</(H-VTENSE)></GRAM-NODE>
      </TENSE></VERB>
  <PSTATE-DATA><S-S><N><(H-INDIC)>shortness of breath</(H-INDIC)></N>
          <MODS><NEG> <T><(H-NEG)>no</(H-NEG)></T></NEG></MODS>
        </S-S></PSTATE-DATA>
  <TEXTPLUS></TEXTPLUS></PATIENT-STATE-IF>
```

Figure 9. XML output of the MLP system

It has made it possible to add medical knowledge to the MLP output, as described in Section 5.2 below.

4.3.4 *Quality control of MLP*

One of the bars to the use of NLP is the recognition that the very flexibility that gives language its widespread utility makes it difficult to ensure that a computer representation arrived at via NLP has captured the intended meaning. At the least, a control of the output in relation to the target representation is essential. To that end, in the case of medical language processing, the LSP-MLP system includes an error-detection program that is applied to each Information Format and Connective in the MLP output. The program is called NIMPH for the 5 types of problems it monitors: *N* for possible mis-analysis of Negation; *I* for Ill-formed semantic output (wrong assigning of subclass occurrence to Information Format slot); *M* for possible mis-analysis of a Modal word; *P* for Partial parse (a correct analysis of an *ASSERTION* or *FRAGMENT* up to a point in the sentence, not the end); *H* for total HangUp (no parse returned).

After each processing run, a report is issued that includes the NIMPH numbers as well as a breakdown of problems by component. In the case of failures of Selection filters, a separate report is issued so that the failures can be evaluated. A Selection failure may be due to the absence of a pattern that should be added to the grammar; it may be due to a mistake in the classification of a word (dictionary error); or it may signal some other problem in the processing.

5. Validation and application

Different objectives can motivate the development of computer programs for language analysis. One objective might be to test the validity of a theory of grammar. For this, one develops a parsing program and writes a grammar, with associated dictionary, based on the theory. If a representative sample of sentences is correctly parsed by such a system, one can claim that up to some level of detail incorporated in the grammar, language structure is 'computable' using this theory.

The initial motivation for developing the Linguistic String Analysis program was of this type. In the grammar of (Sager 1981) great attention was paid to many forms, particularly those involving deep nesting and zeroing, that would not likely occur in most texts but are possible in the English language. The goal was to 'prove' that Linguistic String Analysis was an effective grammatical formulation for the analysis of English sentences.

Harris's theories of language structure do not need computer programs to validate them. His was a different style wherein the theory emerged from a great deal of sentence analysis in which problems were anticipated and dealt with in great detail. And in later work, such as the grammar in (Harris 1982), the analysis is far deeper than what we are in a position to compute today. The string analysis experiment was fitting at a time when there were serious claims that natural language (even just syntactic parsing of sentences) was beyond the reach of machine analysis.

Another motivation for developing computerized language processing is practical. Assuming that such computer programs can be written, can they be made to provide some useful service? This might be considered another type of validation of linguistic analysis, the 'proof of the pudding' type. Whether or not applications are seen as validations of the theory that underlies the linguistic processing, they have their own standing in the larger world. The goal of developing practical applications has driven much of the work in NLP since the early days.

In particular, work on the medical sublanguage by the LSP group has been strongly motivated toward finding useful applications in patient care and related activities. Two examples are given here.

5.1 Healthcare quality assurance

The need to monitor the quality of healthcare that is delivered to patients has been recognized for a long time, but with the recent radical changes to the U.S. healthcare delivery system the issue has become prominent. One of the obstacles to such monitoring is the difficulty of obtaining the data it requires, and, as a prerequisite to that, the specification of what data are required. A step in that direction was made by the National Committee for Quality Assurance by defining a minimal set, called the Health Plan Employer Data and Information Set (HEDIS), for a number of medical conditions.

One of the HEDIS measures concerned whether patients who had suffered a heart attack (acute myocardial infarction, AMI) received beta blocker medication, which was considered desirable unless they had a contraindication as specified in the measure ("Beta blocker treatment after a heart attack", HEDIS 3.0/1998, Volume 2).

To test whether MLP applied to hospital discharge summaries could extract data pertaining to the HEDIS Beta blocker measure, an experiment was performed in which 95 discharge summaries that had been coded by a particular hospital for a diagnosis of AMI were processed by the MLP. The output was mapped to a relational database table (one information format to one row) and retrieval queries were written to extract the rows with pertinent data.

Figure 10 summarizes the experiment and the retrieval results. Figure 11 shows a portion of the combined table of results for the following two queries:

Was the patient given a beta blocker medication?
Did the patient have any contraindications?

HEDIS MEASURE
"Beta blocker treatment after a heart attack (AMI)"

- 95 discharge summaries of patients whose diagnosis had been coded by the hospital as **Acute Myocardial Infarction** (AMI), ICD-9-CM code 410.01 - 410.91
- These discharge summaries had been divided into Sections, such as
  HISTORY OF PRESENT ILLNESS
  PAST MEDICAL HISTORY
  PHYSICAL EXAMINATION
  LABORATORY DATA
  HOSPITAL COURSE
  DISCHARGE STATUS, etc.
- These discharge summaries were analyzed by the Medical Language Processor. Retrieval was performed on the MLP output.
  1. Was the patient given a beta blocker medication?
  2. Did the patient have any contraindications?
- Summary of Retrieval Results:

  – TOTAL NUMBER OF PATIENTS: 95
  – Patients not considered:
    – Under 35: |009|024|
    – Incomplete Documents: |048|093|
  – Results from database queries:

	Beta Blocker Given	Beta Blocker Not Given	Total
With contra-indications	42	19	61
Without contraindications	28	2	30
Total	70	21	91

Figure 10. HEDIS retrieval from MLP output

HEDIS MEASURE
"Beta blocker treatment after a heart attack (AMI)"
from database queries over data
structured by Medical Language Processing

QUERY 2A: Patients given beta blockers who have contraindications
Number of patients: 42
List of patients: 003 005 006 008 010 015 016 018 020 027 030 033 034 036 037 038 041 042 043 045 049 051 055 059 061 062 063 064 068 069 070 071 072 073 076 079 084 085 089 091 092 094

Sent ID	Beta Blockers	Contraindications	Time
003B.06.02 5	LOPRESSOR		
003D.04.01 1		SINUS BRADY-CARDIA AT 55	
003F.01.03 1	LOPRESSOR		
005B.04.02 1	TENORMIN		
005E.01.04 1		MILD RIGHT VENTRICULAR DIASTOLIC DYSFUNCTION	
005E.02.02 5	WITH # LOPRESSOR #		
005E.03.01 4	WITH # LOPRESSOR #		
005F.03.04 1	LOPRESSOR		
006B.02.08 4	LOPRESSOR		
006B.04.03 1		COMPLETE HEART BLOCK	SUBSEQUENTLY #
006B.04.04 2		SINUS BRADY-CARDIA	
006B.05.02 1	LOPRESSOR		

Figure 11. 'Snapshot' of HEDIS retrieval output

It may be considered surprising that of the 91 patient documents that qualified for review, 42 indicated that patients received beta blocker even though they had contraindications. Many of these contraindications (in 29 patients) were congestive heart failure. It was reported during 1997, just one year before the edition of the HEDIS measures available for the experiment, that beta blockers reduce deaths from congestive heart failure. Possibly clinical practice was ahead of the measure.

5.2 Access to narrative data

One of the key problems facing clinicians is access to the right information, at the right time, organized in the optimal way for management of the specific clinical question to be addressed. Effective, high quality care depends on the ability to access, review, and interpret a large amount of information on a given patient as part of the decision making process. Due to cost and time constraints, attempts to have clinicians structure their clinical documentation in order to facilitate this process have been largely unsuccessful, despite the apparent benefits. Consequently, the vast majority of clinical information has remained locked within dictated medical notes, unavailable for retrieval and efficient review. The use of MLP, enriched with medical knowledge, may help to address this problem.

5.2.1 *Adding medical knowledge to MLP*

Currently, there is under development an XML-based medical terminology which can be used to enrich the medical representation obtained by the MLP. The Structured Health Markup Language (SHML) is an organized, highly specialized set of tags that are aimed at describing the medical content of terms encountered in medical text. More than 40 distinct SHML categories have been created, each a description of medical content in patient documents, and each with multiple subcategories. Thus, conceptually, the phrase *pneumonia, right lower lobe, superior, due to Klebsiella* is tagged in XML-based SHML format as

<diagnosis> *Pneumonia* ,
<location> *right lower lobe* </location> ,
<position> *superior* </position> ,
<link> *due to*
<org> **Klebsiella** </org>
</link>
</diagnosis>

Table 4. SHML tag system–correspondence of the anatomic structure and body region hierarchies

Description	SHML Tag Class	Tag Class	Description
anatomic system	a-s	b-r	body region
neurologic system	a-s_nr	b-r_h-n_h	head-neck head
central nervous system	a-s_nr_cns	b-r_h-n_h	head-neck head
brain	a-s_nr_cns_brn	b-r_h-n_h	head-neck head
cardiovascular system	a-s_cv	b-r	body region
heart	a-s_cv_hrt	b-r_tk_thx_msty	mediastinum
chest	a-s	b-r_tk_thx	trunk thorax
respiratory system	a-s_rsp	b-r_tk_thx	trunk thorax
upper respiratory tract	a-s_rsp_u-r	b-r_tk_thx	trunk thorax
lower respiratory tract	a-s_rsp_l-r	b-r_tk_thx	trunk thorax
lung	a-s_rsp_l-r_lng	b-r_tk_thx	trunk thorax
stomach	a-s_gi_gi-tr_u-gi_stm	b-r_tk_thx	trunk thorax

SHML defines several vectors of description of a term found in medical text. Major vectors include sign-symptoms, diagnoses, procedures, organisms, allergies, social behaviors, activities, medications, chemicals, persons, demographics, etc., besides time (frequency, repetition, event-time,[. . .]), links (connective, preposition,[. . .]), modifiers (certainty, negation, changes, amounts,[. . .]).

A term in SHML contains several hierarchical vectors, the first of which is the principal tag, and two of which are always anatomic structure and body region, as shown in Table 4. Thus, terms like *cough* and *shortness of breath* as *N* (noun) and H-INDIC are tagged as

<s-s><a-s_rsp><b-r_tk_thx>
cough
</b-r_tk_thx></a-s_rsp></s-s>

<s-s><a-s_rsp><a-s_cv_hrt><b-r_tk_thx>
shortness of breath
</b-r_tk_thx></a-s_cv_hrt></a-s_rsp></s-s>

This says that

- *Cough* is a sign-symptom, associated with respiratory system, and thorax (in body region trunk).
- *Shortness of breath* is a sign-symptom, associated with both the respiratory system and the [cardiovascular] heart, and the thorax (in body region trunk).

An MLP-SHML correspondence dictionary has been established which currently numbers over 64,000 row entries. Each entry in this dictionary is a row, which is currently defined by one unique MLP triple consisting of a term (word, or several words treated as an idiom), one of its MLP categories, and one of its MLP sublanguage classes. A term having more than one MLP category is represented in more than one row; a term having more than one MLP sublanguage class is represented in more than one row. Thus, every MLP lexical ambiguity is made explicit so that the SHML tag corresponding to each meaning can be unambiguously assigned. Each entry contains:

- the term
- two fields: an MLP category and an MLP class
- SHML tags in 4 fields, laid out as a multi-vector description of the term

SHML is here used as an extension to the MLP in which each triplet of term, MLP category, and MLP sublanguage class defines one unique entity (i.e. one entry).

5.2.2 *A browser for medical narrative data*

The combined MLP-SHML representation of clinical narrative supplies a richly textured clinical data store obtained by linguistic processing and medical tagging of free text patient documents. It remains to make the results selectively viewable by the clinical (or administrative) user. To provide this function, a prototype browser has been developed by InContext Data Systems, Inc. using a relational database system, and HTML and XML web technologies. This is an attempt to integrate different technologies into a system for flexible access to pertinent medical data (Figure 12).

Input to the relational database includes only a preprocessed source medical text, its SHML-tagged MLP output, and an administrative section of the source text. All interchanges between the MLP and the browser are done in ASCII format. The information format (IF) generated by the MLP, now enhanced with medical knowledge from SHML classes, is called a health information unit, or HIU.

The HIU table is then indexed for major SHML tags, such as Signs and Symptoms, Diagnoses, Vital Signs, Labs, Procedures, Medications, Patient Social Behaviors, etc. which can be further sorted by Anatomic System, Body Region, Chemical Classes, and other categories.

To illustrate how the user might access analyzed narrative patient data using the Browser, Figure 13 shows a snapshot of the Browser using the ‘Signs

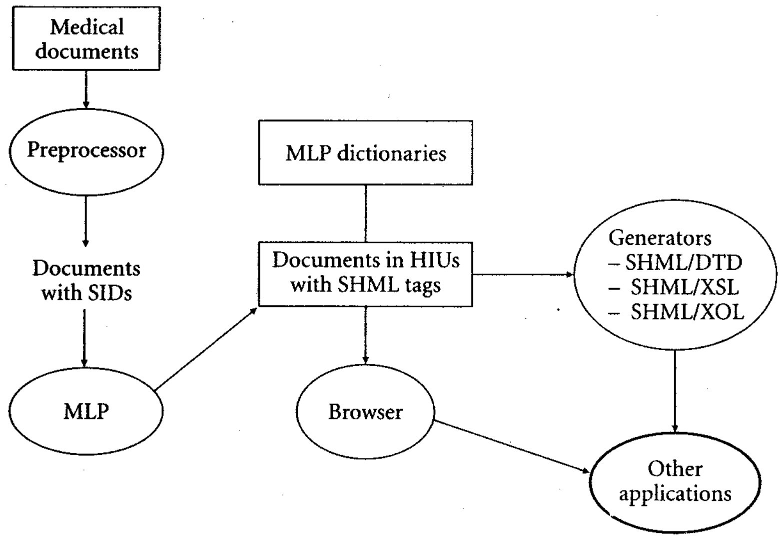


Figure 12. MLP and SHML linkage

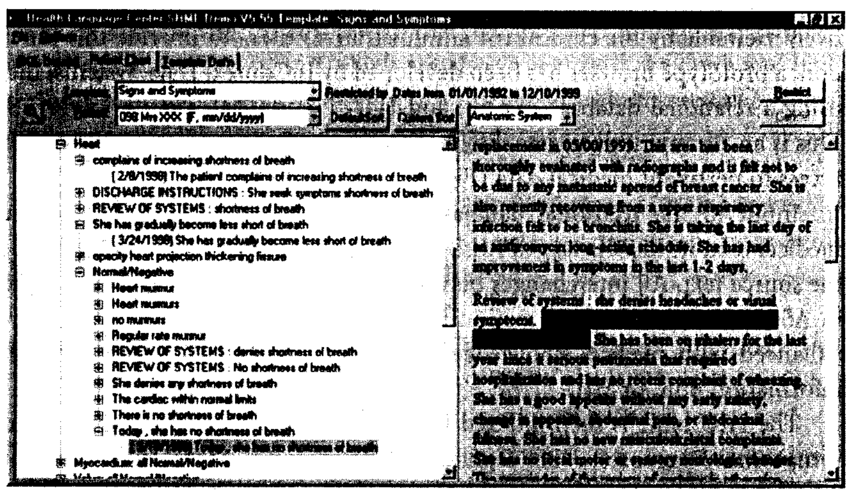


Figure 13.

and Symptoms' template, custom sorted by 'Anatomic System', to present the 'Patient Chart' for Patient 098, Mrs. XXX, female, born *mm/dd/yyyy*, for whom there are 36 documents in the system. There are 544 HIUs found, each tagged with the date of visit. Looking under 'Heart' and then under 'Normal/Negative' subbranch, we find the HIU *Today, she has no shortness of breath.* This HIU is highlighted together with the source text of the sentence, the same sentence as shown in Figure 8 and Figure 9. Note that the HIU containing *shortness of breath* is shown here, correctly, because *shortness of breath* has the SHML anatomic tag <as_cv_hrt> (i.e. 'heart' of the anatomic cardiovascular system). The MLP-SHML tagged form of this HIU is shown in Figure 14.

```
<SID id="990318P2 098.20B.03.02">
<!-- Row 3: Today , she has no shortness of breath . -->

<PATIENT-STATE-HIU id="990318P2 098.20B.03.02"
                sect="REVIEW OF SYSTEMS" row="3" ARG="2" CONNid="2">
  <PT-DEMOG><GENDER>[FEMALE]</GENDER></PT-DEMOG>
  <SUBJECT> <per><_SD14065>she</_SD14065></per> </SUBJECT>
  <VERB><li><li_vhv><_SD7168>has</_SD7168></li_vhv></li>
    <EVENT-TIME>
      <REF-PT>
        <tm><tm_loc><_SD4152>Today</_SD4152></tm_loc></tm>
        ,
        </REF-PT></EVENT-TIME>
    <TENSE>[PRESENT]</TENSE></VERB>
  <PSTATE-DATA>
    <SIGN-SYMP>
      <s-s><a-s_rsp><a-s_cv_hrt><b-r_tk_thx><_SD7917>
        shortness of breath
          </_SD7917></b-r_tk_thx></a-s_cv_hrt></a-s_rsp></s-s>
        <MODS><NEG>
          <md><md_ng><_SD3440>no</_SD3440></md_ng></md>
        </NEG></MODS>
      </SIGN-SYMP>
  </PSTATE-DATA>
</PATIENT-STATE-HIU>
```

Figure 14. An SHML-tagged health information unit

According to the correspondence of the anatomic structure and body region table (Table 4), the HIU *Today, she has no chest pain* is also retrieved as a 'Normal/Negative' finding related to heart. In this case *pain* is a non-specific symptom, and *chest* is in a body region thorax, which contains the heart (<a-s_cv_hrt>).

By contrast, if one selects 'Custom Sort' by 'Body Region', the display area will show 544 HIUs organized under 'Body Region'. We will find under 'Thorax' and then under 'Normal/Negative' subbranch, the three HIUs shown in Figure 8 and Figure 9, because all three terms *cough, chest pain,* and *shortness of breath* have the 'supporting' SHML tag <b-r_tk_thx> (for the thorax in the trunk body region).

In Figure 13, two tabs are concealed by the 'Patient Chart' tab: 'Template Def'n' and '(SQL Details)'. The 'Template Def'n' tab displays two subwindows. The left window presents the current SHML tag set and their hierarchies; the right window is a template building window. By dragging tags from the left window to the right one, the user can build new queries. Retrievals of these queries are displayed on the 'Patient Chart' tab. The '(SQL Details)' tab, for debugging purpose, displays SQL database queries translated from the right 'Template Def'n' subwindow.

The Browser, using SHML-tagged MLP formatted output of original natural language text, enables physicians to (a) create templates best suited for their particular view of patient information from actual documents, (b) see the selected units of information in the context of the original documents for verification, and (c) study patterns across the entire set of patient documents.

6. Summary and conclusion

Harris's string analysis, transformations and the sublanguage method provide a sound basis for language computation, particularly as the basis for representing the information content of scientific and other fact-reporting texts.

In this chapter we have summarized an experience of building upon this basis to arrive at an operational 'real world' system, a medical language processor that can help healthcare workers obtain the data they need from narrative reports.

This effort has been singular in several respects which may not recur. Much of the linguistic input (e.g. the dictionary) was developed manually, demanding a great amount of human resources. We were fortunate that the

project began in a period when the Federal government was still supporting long-range development efforts, and funding was forthcoming from the National Science Foundation and the National Institutes of Health. We were also fortunate in having highly skilled labor contributed on a voluntary basis by persons who believed in the goals of the project.

At the same time, because of the early origin and long history of this work, computer tools that could lighten the burden were not always available as they are now in many places. In general, as the computer field advances, new ways of doing old, still needed, tasks are developed and new tasks for new goals emerge. It is likely that the need for information that is recorded in natural language will not disappear, so there is hope that the methods of language analysis that marked Harris's oeuvre will find their application in the future of language technology, along with their proper place in the history of the field of linguistics.

Acknowledgements

First and foremost we wish to acknowledge the essential contribution of Margaret Lyman, M.D., to the development of the medical language processor. In addition to her clinical practice, from 1977 until her death in November 2000, Dr. Lyman was undaunted in her dedication to this effort. She believed in the importance of the freely dictated medical report and in the possibility of using computers to organize and make accessible its content for the betterment of patient care and the advancement of medical knowledge. Dr. Lyman was a tireless worker and one who inspired all of us toward greater effort and dedication. We cannot match her in either of these, but we hope to realize at least some of her goals. Dr. Leo Tick provided technical support for Dr. Lyman's activities and also reimplemented, improved, and maintained the parser; he added much to the development of the medical language processor.

Over the years of the Linguistic String Project many coworkers participated in its activities. It is not possible to acknowledge them all, but some (in alphabetical order) are: Barbara Anderson, Jeff Bary, Beatrice Bookchin, Shiun Chen, Emile Chi, Pascale Claris, Judith Clifford, Eileen Fitzpatrick, Carol Foster, Carol Friedman, Dan Gordon, Ralph Grishman, Lynette Hirschman, Stephen Johnson, Michiko Kosaka, Joyce London, Catherine MacLeod, Elaine Marsh, James Morris, Neal Oliver, Morris Salkoff, Richard Schoen, Guy Story, Susanne Wolff, and Su Yun.

References

Bloomfield, Leonard. 1933. *Language.* New York: Holt, Rinehart & Winston.

Gleitman, Lila R. 1959. "Word and word-complex dictionaries". Transformations and Discourse Analysis Project (TDAP) 16. Philadelphia: Department of Linguistics, The University of Pennsylvania.

Grishman, Ralph & Richard Kittredge (eds.). 1986. *Analyzing Language in Restricted Domains: Sublanguage description and processing.* Hillsdale, NJ: Lawrence Erlbaum.

Harris, Zellig S. 1952. "Discourse analysis". *Language* 28.1: 1–30.

Harris, Zellig S. 1959. "Computable syntactic analysis". Transformations and Discourse Analysis Project (TDAP) 15. Philadelphia: Department of Linguistics, The University of Pennsylvania.

Harris, Zellig S. 1962a. *String Analysis of Sentence Structure.* (= *Papers on Formal Linguistics,* No. 1.) The Hague: Mouton, 70 pp. (Repr., 1964 and 1965.) [Revised version of Harris 1959.]

Harris, Z.S. 1962b. "A cycling cancellation-automaton for sentence well-formedness". Transformations and Discourse Analysis Project (TDAP) 51. Philadelphia: Department of Linguistics, The University of Pennsylvania. [Also in *International Computation Centre Bulletin* 5(1966): 69–94. Reprinted in Harris (1970: 286–309).]

Harris, Zellig S. 1963. *Discourse Analysis Reprints.* The Hague: Mouton & Co.

Harris, Zellig S. 1967. "Decomposition lattices". Transformations and Discourse Analysis Project (TDAP) 70. Philadelphia: Department of Linguistics, The University of Pennsylvania. [Reprinted in Harris (1970: 578–602).]

Harris, Zellig S. 1968. *Mathematical Structures of Language.* New York: John Wiley/ Interscience Publishers.

Harris, Zellig S. 1970. *Papers in Structural and Transformational Linguistics.* Dordrecht: D. Reidel Publishing Company.

Harris, Zellig S. 1982. *A Grammar of English on Mathematical Principles.* New York: John Wiley & Sons.

Harris, Zellig S. 1991. *A Theory of Language and Information: A mathematical approach.* Oxford: Clarendon Press.

Harris, Zellig S., Michael Gottfried, Thomas Ryckman, Paul Mattick, Jr., Anne Daladier, Tzvee N. Harris, & Suzanna Harris. 1989. *The Form of Information in Science: Analysis of an immunology sublanguage.* Preface by Hilary Putnam. (= *Boston Studies in the Philosophy of, Science,* 104). Dordrecht: Kluwer Academic Publishers.

Hirschman, Lynette, Ralph Grishman, & Naomi Sager. 1975. "Grammatically-based automatic word class formation". *Information Processing and Management II,* pp. 39–57.

Joshi, Aravind K. 1960. "Recognition of local substrings". Transformations and Discourse Analysis Project (TDAP) 18. Philadelphia: Department of Linguistics, The University of Pennsylvania.

Joshi, Aravind K. & Philip D. Hopely. 1999. "A parser from antiquity: An early application of finite state transducers to natural language parsing", in *Extended Finite State Models of Language* (Proceedings of the ECAI 96 Workshop), ed. by Andras Kornai, 6–15. New York & London: Cambridge University Press.

Kaufmann, Bruria. 1959. "Right-order substrings and wellformedness". Transformations and Discourse Analysis Project (TDAP) 19. Philadelphia: Department of Linguistics, The University of Pennsylvania.

Karttunen, Lauri. 1997. "Comments on Joshi", in *Extended Finite State Models of Language* (Proceedings of the ECAI 96 Workshop), edited by Andras Kornai. New York & London: Cambridge University Press.

Kittredge, Richard, & Jack Lehrberger. 1982. *Sublanguage: Studies of language in restricted semantic domains.* Berlin: Walter de Gruyter.

Marsh, E., and Friedman, C. 1985. Transporting the Linguistic String Project system from a medical to a Navy domain, *ACM Transactions on Office Information Systems* 3:2, pp. 121–140.

Morris, J. 1965. The IPL string analysis program, in *First Report of the String Analysis Programs,* Philadelphia: Department of Linguistics, The University of Pennsylvania.

Nazarenko, A., P. Zweigenbaum, B. Habert, & J. Bouaud. To appear. "Corpus-based extension of a terminological semantic lexicon", in *Recent Advances in Computational Terminology,* ed. by D. Bourigault, C. Jacquemin, & M.-C. L'Homme. Amsterdam: John Benjamins.

Nhàn, Ngô Thanh, Naomi Sager, M. Lyman, L. J. Tick, F. Borst, & Y. Su. 1989. "A medical language processor for two Indo-European languages". *Proceedings of the 13th Annual Symposium on Computer Application in Medical Care* (L. C. Kingsland, ed.), 554–558. Washington, D.C.: IEEE Computer Society Press.

Oliver, N. 1992. *Sublanguage Based Medical Information Processing System for German.* Ph.D. thesis, New York University.

Sager, Naomi. 1959. "Elimination of alternative classification". Transformations and Discourse Analysis Project (TDAP) 17. Philadelphia: Department of Linguistics, The University of Pennsylvania.

Sager, Naomi. 1960. "Procedure for left-to-right recognition of sentence structure", in Transformations and Discourse Analysis Project (TDAP) 27. Department of Linguistics, The University of Pennsylvania.

Sager, Naomi. 1967. "Syntactic analysis of natural language". *Advances in Computers 8,* 153–188. New York: Academic Press.

Sager, Naomi, & Ralph Grishman. 1975. "The restriction language for computer grammars of natural language". *Communications of the ACM 18,* pp. 390–400.

Sager, Naomi. 1978. "Natural language information formatting: The automatic conversion of texts to a structured data base". In *Advances in Computers* 17, edited by M. C. Yovits, 89–162. New York: Academic Press.

Sager, Naomi. 1981. *Natural Language Information Processing: A computer grammar of English and its applications.* Reading, Massachusetts: Addison-Wesley.

Sager, Naomi, Carol Friedman, M. S. Lyman, MD, & members of the Linguistic String Project. 1987. *Medical Language Processing: Computer management of narrative data.* Reading, Massachusetts: Addison-Wesley.

Sager, Naomi, M. S. Lyman, C. Bucknall, N. T. Nhàn, & L. J. Tick. 1994. "Natural language processing and the representation of clinical data", *Journal of the American Medical Informatics Association,* 1.2: 142–160.

Sager, Naomi, N.T. Nhàn, M.S. Lyman, & L.J. Tick. 1996. "Medical language processing with SGML display". *Proceedings of the 1996 AMIA Annual Fall Symposium.* Hanley & Belfus. Pp. 547–551.

Shapiro, P.A. 1967. "Acorn", in *Methods of Information in Medicine* 6: 153-162.

Spyns, P., N.T. Nhàn, E. Baert, N. Sager, & G. De Moor. 1996. "Combining medical language processing and mark-up technology: An experiment applied to Dutch", in *Proceedings of MIC96*, edited by C. Sevens & G. De Moor, pp. 69–77. Brussels.

Wolff, S. 1984. "The use of morphosemantic regularities in the medical vocabulary for automatic lexical coding". *Methods of Information in Medicine,* 23: 195–203.

The World Wide Web Consortium (W3C). 1999. "W3C technologies: XML". (http: //www.w3.org). [See under XML, XML technical reports, XML Base, XML Encryption, XML Protocol, XML Query, XML Schema, XML Signature, etc.]