INFORMATION REDUCTION OF TEXTS BY SYNTACTIC ANALYSIS

Naomi Sager, Ph. D.

This paper will describe a working computer program for syntactic analysis of English sentences, and will indicate how the output of the program, when enriched by allied formal linguistic techniques, can be used to align and manipulate informationally related terms and portions in a scientific text.

The aim of the paper will thus be to show how at least one system of formal linguistic analysis can be used for processing the information in a scientific text to an extent which is usually thought to require semantic analysis. The aim is not to discount semantic analysis or the importance of meaning, but rather to try to put to the utmost practical use some of the methods and results which are now available from developments in formal linguistics.

1. The String Program for Sentence Analysis

The string program provides a good beginning point for automatic information processing because it segments a sentence into informational units which are related to the information in the original sentence. This can best be seen by examining computer outputs for sentences of a scientific text. The principles of string analysis can also be described more succinctly using such an output as illustration.

In this case the text examined was the following abstract from the Journal of the American Chemical Society, June 5, 1957, volume 79, "The Amino Acid Sequence of Glucagon. I. Amino Acid Composition and Terminal Amino Acid Analyses", by W. W. Bromer, A. Staub, E. R. Diller, H. L. Bird, L. G. Sinn, and Otto K. Behrens:

"Evidence is presented that glucagon is a small protein consisting of a single chain of 29 amino acid residues. The N-terminal amino acid is histidine as determined by the dinitrophenylation method: the C-terminal residue is threonine on the basis of evidence obtained from hydrazinolysis and carboxypeptidase treatment. Glucagon contains single residues of 7 amino acids: among them, methionine, tryptophan, valine and alanine are liberated from the C-terminus of the molecule by carboxypeptidase."

The sentence 2A of the Glucagon abstract which reads The N-terminal amino acid is histidine as determined by the dinitrophenylation method is decomposed into a number of word-strings, written on different lines, each having a fixed grammatical structure, as indicated by the sequence of grammatical names above the sentence-words (APPENDIX). These are called elementary strings, and they are of different kinds. The string in the first numbered line, acid is histidine, is an elementary sentence (disregarding for the moment the grammatical need for the), which is called a center string. The other strings are not sentences and are called adjectival strings, since they are added to the elementary sentence or other adjunct strings without changing the grammatical status of the string to which they are adjoined. Semantically, they are usually modifiers of the element [or string] they adjoin. Thus in the line numbered 1, the words the N-terminal amino, which are adjoined to the left of acid in the center string acid is histidine, are semantic modifiers of acid (again disregarding the status of the article, and accepting amino as adjective; amino acid could instead be treated as a single name). Note that the appearance of the number 1 to the left of acid in line 6 indicates that acid has left adjuncts which are to be found in line 1.

In order to carry out such a segmentation in a general way on all sentences of the language, it is necessary to treat words as members of word-categories; e.g., acid belongs to the word-category N (noun), is to the word-category V (verb) etc. Each sentence is then represented as a sequence of word-categories and the analysis is made on the sequence of category-symbols. Thus the sequence of words in sentence 2A

the N-terminal amino acid is histidine

is represented as the word-category sequence

T A A N V N

and the analysis is performed on this sequence of symbols. [In many cases individual words belong to more than one word-category, thus giving rise to a family of representations for a given sentence.]

A grammar is also stated in terms of word-categories, since it has been shown in linguistics that it is possible to characterize objectively and even mechanically the sentences of a language, taken as sequences of word-categories, in terms of a reasonably small number of elemen-
tary sequence of word-categories and operations on them. In a string grammar these elementary sequences are the elementary strings of the language seen as sequences of word-categories (e.g., \( N \) is \( N \)) corresponding to word occurrences in sentences (e.g., \( \text{acid} \) is \( \text{histidine} \)). The elementary strings combine to form more complex sentences by the sponding to word occurrences in sentences (e.g., \( \text{acid} \) is \( \text{histidine} \)). The operations of adjunction, and replacement expressed in the following string-class definitions, where \( A = X_1 \ldots X_n \) is an elementary string.

\( L_X \) left adjuncts of \( X \): adjoined to a string \( A \) to the left of \( X \) in \( A \) or to the left of an \( L_X \) adjoined to \( A \) in this manner.

\( R_X \) right adjuncts of \( X \): adjoined to a string \( A \) to the right of \( X \) in \( A \), or to the right of an \( R_X \) adjoined to \( A \) in this manner.

\( N_X \) replacement strings of \( X \): adjoined to a string \( A \) replacing \( X \) in \( A \).

\( S_A \) sentence adjuncts of the string \( A \), adjoined to \( A \) at any inter-element point or to the left of \( X_i \) or to the right of \( X_{ni} \) or to the right of an \( S_A \) which has been adjoined to \( A \) in one of these manners.

\( C_A \) conjunctional strings of \( A \), conjoined to the right of \( X_i \) in \( A \) (1 \( \leq i \leq n \)) or to the right of an \( C_A \) conjoined in this manner.

\( Z \) center strings, not adjoined to any string.

There are various restrictions on the repetition and the order of various members of the classes of adjuncts.

An analysis of a sentence consists of decomposing the sentence into its component elementary strings, and showing that each such string \( B \) enters another elementary string \( A \) at the point of entry stated in the definition of the string-class to which \( B \) belongs.

We can now see how the computer output for sentence 2A constitutes an analysis of the sentence. The word-string \( \text{acid} \) is \( \text{histidine} \) in line 6 is the center string \( N \) is \( N \). To the left of \( \text{acid} \) are the left adjuncts of \( N \), the \( N \)-terminal amino (T A A), which are found in line 1. The string as determined by the dinitrophenylation method, beginning in line 5, is a sentence-adjunct adjoined to the entire center string. This is indicated by the appearance of 5 at a point, labeled * in the center spring in line 6. This segment is here further decomposed into a preposition-plus-noun string (P N) by method in line 3 as right adjunct of determined in line 4, and the dinitrophenylation in line 2, as left adjuncts of method in line 3.

In brief about the program itself. The text is input without pre-editing, as it appears on the printed page. The words of the text are also separately input with their syntactic classification and subclasses (but of course not on the basis of their particular use in this text); i.e., a grammatical dictionary must be made for the texts which are to be analyzed. The grammar is independent of the program and as indicated above, consists of classes of strings totaling about 125 strings in all, in about 20 classes, with about 30 more detailed grammatical restrictions on the strings. All syntactic analyses of a sentence are obtained. The program does not print (but can print) those analyses of a sentence which can be obtained from some previous analysis by a predictable reassignment of an adjunct string to be a modifier of another element in the sentence. The analyses then number 1–5 per sentence, with the first analysis most often expressing the author's intended meaning. A current version of the program written in FAP for the 7094 obtains the first analysis of a typical sentence in about 1 second and all analyses in about 5 seconds.

2. Availability of Transformational Analysis

Additional information about the structure of a sentence is obtained using the method of transformations. Transformational analysis enables one to bring out structures far deeper than string analysis can hope to bring out in a natural way. It is thus a powerful tool for various applications, such as discourse analysis, information retrieval, etc. In the University of Pennsylvania, Dr. Joshi has been presently working on a transformational grammar of English, in particular various representations of transformational grammars from the point of view of constructing a decomposition procedure. His procedure is entirely based on some rather basic and quite general properties of transformations and does not depend on any prior analysis, such as string analysis described above.

However, many of the further refinements which are established by transformational analysis can be added to the string program because the operands of each transformation on a sentence are located as distinguished parts of particular strings of the sentence; the string analysis required in order to use the subclasses and restrictions arising from a transformation \( T \) covers only such parts of the sentence as does the transformational analysis required for the operation of \( T \). For example, no transformation will have as its operands part of one string and an adjunct of another string.

Dr. Joshi had designed an algorithm (machine independent) for transformational decomposition, and is currently writing a substantial portion of a transformational grammar of English in a form
suitable for the algorithm. This representation is linguistically well motivated and is a natural one for the algorithms designed. Independence of grammatical information and the procedure is maintained in the same way as in the string analysis program.

3. Informational Reduction of a Text

Given a short discourse; e.g., an abstract or the collection of result sentences of an article, we propose to arrange it in a form more useful for locating and processing the information in the text, and to reduce it by dropping locally irrelevant material. Table 1 and the accompanying notes give a preliminary illustration of this work.

Starting with the computer outputs, the main methods are:

1. We can shift strings or parts of string around to arrive at an alignment (or format) by means of
   (a) information preserving; i.e., paraphrastic, transformations (this step is in principle mechanizable);
   (b) informationally neutral discourse analysis methods (the validity of any particular application of discourse analysis to a text can be checked formally).

   “Alignment” here means, roughly putting into one column the repetitions, synonyms, and classifiers of a term, if the verbs relating them to the entries in another column are themselves repetitions, synonyms, or classifiers of each other.

2. We can drop locally irrelevant word-sequences on the basis of the occurrence or absence of these words or their synonyms in particular related positions of neighboring sentences.

3. We make use of a set of synonyms and a hierarchical classification for the terms of the science, in order to recognize the repetition of concepts.

Table 1 is obtained from the computer outputs by means of the standardized operations listed in the notes accompanying the table.

Every line is an assertion. There are two main families of assertions. One, those whose subject is glucagon, has as its predicates information about the amino acids: that they are 29 in number and in a single chain, and that seven are single residues. The other family of assertions has names of amino acids as subject, and as its predicates locations of the amino acids; each of these assertions is accompanied, in a separate column, by the conditions by means of which the location was determined.

It is obviously possible to search for and to compare particular types of information here, because they fall into a format. Various further alignments and word-omissions could be made to tighten this format even more.

What this means is that one of the things one can think of is the reduction of selected portions of articles to informationally inspectable form; i.e., the material in the sentences would be arranged so that the different kinds of information in the sentence (e.g., what relation is being asserted, under what conditions, etc.) are in predetermined parts of a format. This is possible because we have informationally neutral processors of language and because we are speaking here about discourse in specific fields, where we can use the constraints both of the grammar and of the particular field.

The tabulation of textual information by means of such alignment is an extreme result, which one can hope to mechanize. There are various other easier results, the possibilities of which are partly indicated by the material presented above. It seems possible, for example, by grammatical properties, to extract from an article those sentences which present the results obtained, or those sentences which discuss the validity of the methods used, or those sentences describing the laboratory or calculational activities, or sentences relating one set of results to another.

One might also consider a mechanical utilization of a pair index. An article would be indexed for a particular pair of terms if it states some relation between those terms. A mechanical processing of an article or a sentence without using linguistic analysis cannot do such indexing; e.g., two terms can be next to each other in a sentence and still have no index-relation. In contrast, two terms can have a semantically strong relation in a sentence only if they have a string relation to each other within that sentence.

Hence it is possible to conceive of a pair-index search of an article, given a string analysis of it (and a list of synonyms and classifiers for the science).

EXAMPLES

PARSE NO. 1

SENTENCE 1. EVIDENCE IS PRESENTED THAT GLUCAGON IS A SMALL PROTEIN, CONSISTING OF A SINGLE CHAIN OF 29 AMINO ACID RESIDUES.

SENTENCE = INTRODUCER CENTER END-MARK

10. C1 ASSER- = * SUBJECT * VERB * OBJECT R-V * TION EVIDENCE IS 1
9. C132 PAS- = VEN SIVE PRESENTED 9
8. C108 = THAT C1 ASSERTION 8
7. C112 = SUBJECT * VERB * OBJECT R-V * TION GLUCAGON IS 2 PROTEIN, 7
6. L-N = ARTICLE QUANTIFIER TYPE NS NOUN A SMALL
7. O VING OBJ = VING * OBJECT R-V * 
   CONSISTING 6
6. C20 P N = L-P P N 
   OF 3 CHAIN 5
3. L-N = ARTICLE QUANTIFIER ADJECTIVE TYPE NS NOUN A 
   SINGLE
5. C20 P N = L-P P N 
   OF 4 RESIDUES
4. L-N = ARTICLE QUANTIFIER ADJECTIVE TYPE NS NOUN 
   29 AMINO ACID

SENTENCE 2A. THE N-TERMINAL AMINO ACID IS HISTIDINE AS DETERMINED BY THE DINITROPHENYLATION METHOD.

SENTENCE = INTRODUCER CENTER END-MARK 
10 SEMICOLON
6. C1 ASSERTION = *SUBJECT *VERB *OBJECT R-V * 
   1 ACID IS HISTIDINE 5
1. L-N = ARTICLE QUANTIFIER ADJECTIVE TYPE NS N-TERMINAL NOUN
5. C162 = L-CS CS2 C132 PASSIVE AS 4
4. C132 PASSIVE = VEN *PASSIVE-O R-V * 
   DETERMINED 3
3. C20 P N = L-P P N 
   BY 2 METHOD
2. L-N = ARTICLE ADJECTIVE NOUN QUANTIFIER TYPE NS DINITRO- 
   THE PHENYLATION

SENTENCE 2B. THE C-TERMINAL RESIDUE IS THRENONINE ON THE BASIS OF EVIDENCE OBTAINED FROM HYDRAZINOLYSIS AND CARBOXYPEPTIDASE TREATMENT.

SENTENCE = INTRODUCER CENTER END-MARK 
9 SEMICOLON
9. C1 ASSERTION = *SUBJECT *VERB *OBJECT R-V * 
   1 RESIDUE IS THRENONINE 8
1. L-N = ARTICLE QUANTIFIER ADJECTIVE TYPE NS THE C-TERMINAL NOUN
8. C20 P N = L-P P N 
   N 2 BASIS 7
2. L-N = ARTICLE QUANTIFIER ADJECTIVE TYPE NS THE NOUN
7. C20 P N = L-P P N 
   OF EVIDENCE 6
6. C132 PASSIVE = VEN *PASSIVE-O R-V * 
   OBTAINED 5
5. C20 P N = L-P P N 
   FROM 4 TREATMENT
4. L-N = ARTICLE QUANTIFIER ADJECTIVE TYPE NS NOUN M1 
   HYDRAZINOLYSIS AND 3
3. Q1 = NOUN 
   CARBOXYPEPTIDASE

SENTENCE 3A. GLUCAGON CONTAINS SINGLE RESIDUES OF 7 AMINO ACIDS.

C = INTRODUCER CENTER END-MARK 
4 SEMICOLON
4. C1 ASSERTION = *SUBJECT *VERB *OBJECT R-V * 
   GLUCAGON CONTAINS NS 1 RESIDUES 3
1. L-N = ARTICLE QUANTIFIER ADJECTIVE TYPE NS NOUN SINGLE
3. C20 P N = L-P P N 
   OF 2 ACIDS
2. L-N = ARTICLE QUANTIFIER ADJECTIVE TYPE NS NOUN 
   7 AMINO

SENTENCE 3B. AMONG THEM, METHIONINE, TRYPTOPHAN, VALINE AND ALANINE ARE LIBERATED FROM THE C-TERMINUS OF THE MOLECULE BY CARBOXYPEPTIDASE.

SENTENCE = INTRODUCER CENTER END-MARK 
11 SEMICOLON
11. C1 ASSERTION = *SUBJECT *VERB *OBJECT R-V* 
   1 METHIONINE, 4 ARE 10
1. C20 P N = L-P P N 
   AMONG THEM.
4. Q1 = A20 M16 
   TRYPTOPHAN .3
10. C132 PASSIVE = VEN *PASSIVE-O R-V* 
   LIBERATED 9
3. Q1 = A 20 M1 
   VALINE AND 2
9. C20 P N = L-P P N 
   FROM 5 C-TERMINUS 8
2. Q1 = A20 
   ALANINE
5. L-N = ARTICLE QUANTIFIER ADJECTIVE TYPE NS NOUN THE 
8. C20 P N = L-P P N 
   OF 6 MOLECULE 7
6. L-N = ARTICLE QUANTIFIER ADJECTIVE TYPE NS NOUN THE 
7. C20 P N = L-P P N 
   BY CARBOXYPEPTIDASE

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Connective Σ</th>
<th>V</th>
<th>Ω</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>glucagon</td>
<td>small</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(protein)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>-ing</td>
<td>consists of 29 amino acid residues in single chain</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>glucagon</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Some detailed questions were raised about the resolution of ambiguities. Sager pointed out that while it was true that there were ambiguities, these could be resolved by the computer in the last analysis. The computer program suggests more detailed restrictions because it sees ambiguities which the reader may not. Often these are the result of some small grammatical restriction on the class of words. Garvin asked about the manner in which the program scans the input string. Sager said that the program scans input in a single pass from the left, and that there is an order in which the input is matched against a table of strings. She made the general observation from string theory that as a sentence is scanned from the left, at a given point, if there has been an analysis to that point, the n-th word, then the n-minus-one word has been analyzed as a member of a substring. The n-th word must be either a continuation of that substring or the beginning of some new string. There is therefore a very limited number of strings to be considered at every point. Sager discussed the transformational relations which emerge in the course of string analysis. These relations provide subclasses which can be used to restrict the strings. She also made a distinction between gross string analysis and refined string analysis. In gross string analysis, the aim is to show that a grammar has certain intrinsic properties which the language also has because it is describable by its string analysis. In the refined string analysis members of a set are distinguished.

Asked by Kerr to list the advantages of string analysis over transformational, Sager remarked that one advantage of string technique is that it is more easily computable. Pierce remarked that Sager's program runs fast, as programs go, because it does not produce many parsings. It has the advantage that the natural meaning is often the first one. Garvin commented that the real problem was not different types of parsings, but how deep one has to go for a certain purpose. In tabular comparison, a really deep analysis might not be needed.

There was some discussion of the differences between string analysis and transformational analysis, and Kuno stated that he did not think string analysis was strong enough for the description of natural languages. Sager replied that string analysis, as here defined, was not strong enough, but that Joshi's or Harris's method which deals with segments of the sentences as the domain of transformation, is strong enough. Other comments were made on the relatively low number of rules, and Kuno said there were other factors besides the number of rules, such as the type of computer and the language used, and that it would therefore be dangerous to make comparisons with respect to the speed of sentence recognition. Sager agreed with Kuno that the way to evaluate programs was on the basis of how correct the analyses were.

### Table 1—Continued

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Connective</th>
<th>Σ</th>
<th>V</th>
<th>Ω</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2A</td>
<td>histidine</td>
<td></td>
<td>the amino-acid at N-terminus</td>
<td>by dinitrophenylation method</td>
<td></td>
</tr>
<tr>
<td>2B</td>
<td>threonine</td>
<td></td>
<td>residue at C-terminus</td>
<td>from hydrasinosysis and carboxypeptidase treatment</td>
<td></td>
</tr>
<tr>
<td>3A</td>
<td>glucagon</td>
<td>contains single residues of 7 amino acids</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3B.1</td>
<td>methionine</td>
<td>are among &quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>tryptophan</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>valine</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>alanine</td>
<td>and</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3B.2</td>
<td>&quot;</td>
<td>&quot;</td>
<td>are liberated C-terminus by carboxypeptidase from the molecule</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Table 1: Notes)

Transformational and Discourse Analysis processing of Glucagon abstract.

S1. Drop sentence operator Evidence is presented that on the discourse analysis result that these operators form a metascience frame for the object-language report.

Move classifier (protein) of glucagon to glucagon column; special case of the commutative wh-connective.

Nₐₙ, of N→N in Nₐₙ, where Nₐₙ= nouns of arrangement and containers.

Here, chain of . . . residues → . . . residues in chain.

Drop automatic a and is.

S2A. N₁ a N₂→N₃ P N₁, where a is adjectivizer and P is preposition: N-terminal amino acid→amino acid at N-terminus.

N₁ is N₂→N₃ is N₁, except for storable subclasses. Here we invert the N-terminal amino acid is histidine to histidine is the N-terminal amino acid.

Drop sentence operator as determined.

S2B. Drop sentence operator on the basis of evidence.

Remaining transformations as in S2A.

S3B. In among them, reinsert the subject which has been scoed from the following clause, and reinsert the object which has been pronounced from the proceeding clause.

Note.—This is only a partial alignment obtained by using the well-established transformations. Further alignment and reduction is possible with the aid of additional transformations.
References


