[*Mechanical Translation and Computational Linguistics*, vol.11, nos.1 and 2, March and June 1968]

On-Line Semantic Analysis of English Texts*

by Yorick Wilks, Pembroke College, Cambridge

This paper describes the use of an on-line system to do word-sense ambiguity resolution and content analysis of English paragraphs, using a system of semantic analysis programmed in Q32 LISP 1.5. The system of semantic analysis comprises dictionary codings for the text words, coded forms of permitted message, and rules producing message forms in combination on the basis of a criterion of semantic closeness. All these can be expressed as a single system of rules of phrase-structure form. In certain circumstances the system is able to enlarge its own dictionary in a real-time mode on the basis of information gained from the actual texts analyzed.

1. Introduction

In this paper I describe a system for the on-line semantic analysis of texts up to paragraph length. It was programmed and applied in Q32 LISP 1.5 to material of two sorts: newspaper editorials, and passages of philosophical argument. The immediate purpose of the analysis was to resolve the word-sense ambiguity of the texts: to tag each word occurrence in the texts to one and only one of its possible senses or meanings, and to do so in such a way that anyone could judge the output's success or failure without knowing the coding system.

The system analyzes text up to paragraph length, since I follow a working hypothesis that many word-sense ambiguities cannot be resolved within the bounds of the conventional text sentence; there simply isn't enough context available. So, for example, if someone reads, in British English at least, "I'll have to take this post after all," then he does not know, without more context, whether he is reading about an employment situation or one concerned with the purchase of gardening equipment. If that sentence were analyzed, by any ambiguity resolution system, as part of a larger text, we would expect as a report on the word "post" either "post as a job" or "post as a stake," depending on the larger text of which this example sentence was a part.

When I call this process of tagging words "ambiguity resolution," I do not mean that the words of real texts are usually ambiguous, that a reader cannot decide which of their meanings or senses are meant. If a word is genuinely ambiguous in use, that usually indicates a fault on the part of the writer or speaker. What I am referring to is a procedure for getting a computer to do what human beings do naturally when they read or listen, namely, to interpret each word in a text in one and (usually) only one of its possible senses. So, and again in British English, anyone reading "I must take these letters to the post" just *knows* that the sense of "post" in question is "post as a place for depositing mail" and not either of the two other senses distinguished earlier.

An ambiguity-resolution system would be of some interest within computational linguistics even if it worked on a purely ad hoc basis, since word ambiguity is probably the problem holding up the achievement of reliable mechanical translation. However, the present system is essentially one for the representation of the content of texts. Its use as an ambiguity-resolution procedure, described here, is some test of its ability to represent texts for subsequent interrogation as part of a more general information system since representing content usefully involves disambiguation essentially. Any attempt to represent the content of "I suppose I'll have to take this post" must be prepared to store different representations for the two major interpretations of that sentence I distinguished earlier. Once a representation has been assigned by any method, then an ambiguity resolution for the words of the text can be read from it, and the correctness or otherwise of the resolution is some test of the adequacy of the original representation. That is what the present system does at this stage: it simply outputs a tagging of each text word to one and only one of its senses, as they are distinguished by a semantic dictionary. In the experiment to be described, texts were initially segmented into fragments (see below) for the purposes of the analysis, and in the final output each fragment is given with a list of sense explanations for all the words in it which are resolved (or which had only a single-sense entry initially and so are trivially resolved). A list is also given of words not resolved, if any (see fig. 1). The original English form of the sentence to which the two fragments correspond is "Britain's transport system and with it the traveling public's habits are

[*] Presented at the Second International Congress of Applied Linguistics, Cambridge, September 1969. This work has been supported by contract AFOSR F44620-67-COO46 from the Air Force Office of Scientific Research, monitored by Mrs. Rowena Swanson and administered by the Institute for Formal Studies, Los Angeles. The computation described was done on the time-shared on-line system at System Development Corporation, Santa Monica, Calif. This work is at present supported by contract N00014-67-A-00112-0049 from the Office of Naval Research.

```
(((BRITAIN'S TRANSPORT SYSTEM ARE CHANGING)
    ( WORDS RESOLVED IN FRAGMENT)
      (TRANSPORT AS PERTAINING TO MOVING THINGS ABOUT)
      (BRITAIN'S AS HAVING THE CHARACTERISTIC OF A
      PARTICULAR PART OF THE WORLD)
      (SYSTEM AS AN ORGANIZATION)
 (ARE AS HAVE THE PROPERTY) (CHANGING AS ALTERING)))
 ((WORDS NOT RESOLVED IN FRAGMENT) NIL))
 ((WITH IT THE TRAVELING PUBLICS HABITS)
 ((WORDS RESOLVED IN FRAGMENT)
 ((TRAVELING AS MOVING FROM PLACE TO PLACE)
 (IT AS INANIMATE PRONOUN)
 (HABITS AS REPEATED ACTIVITIES)))
 ((WORDS NOT RESOLVED IN FRAGMENT) NIL)))
```

FIG. 1.—Resolution output from the LISP 1.5 program

changing." The way in which the sentence was broken up into fragments and the significance of the LISP "NIL" symbols will appear later on.

This sort of decision making assumes that it is useful, even though not completely perspicuous, to speak of "senses of words," and that ordinary speakers of English can agree that, in "I won a round of golf today" and "One round of sandwiches, please," the word "round" is being used in two different senses. Not all linguists would agree with this common sense intuition, and they have a case in that it is very difficult to assign word occurrences to "sense classes" in any manner that is both general and determinate. Even the common sense intuition cannot be pushed very far. In the sentences "I have a stake in this country" and "My stake on the last race was a pound," is "stake" being used in the same sense? If "stake" can be interpreted to mean something as vague as "stake as any kind of investment in any enterprise," then the answer is yes. So if a semantic dictionary contained only two senses for "stake," that vague sense together with "stake as a post," then one would expect the word "stake" to be tagged to the vague sense in both the sentences above. But if, on the other hand, the dictionary distinguished "stake as an investment" and "stake as the initial payment in a game or race" then the answer would be expected to be different. Thus, word disambiguation is relative to the dictionary of sense choices available, and can have no absolute quality about it.

The first requirement for any semantic system of this sort is a coding scheme that can distinguish the different senses of words in a dictionary. Let us assume, by way of example, that we want to distinguish two senses of "salt," namely, "salt as an old sailor" and "salt as the substance sodium chloride." Two natural markers to use for this purpose would be one meaning any substance, let us say STUFF, and one meaning any human being, let us say MAN. These markers represent the highest useful level of classification for each word sense. That is to say, for example, that the class of men includes the class of sailors, and so of old sailors. So MAN will be the main marker, or *head,* in the coding for that sense of "salt." Let us suppose, then, that these two senses of "salt" can be expressed by semantic formulas made up from such markers nested, or otherwise combined, to any degree of complexity needed to distinguish the senses. The head of any formula will be its main category marker; so it will be MAN for "salt as an old sailor" and STUFF for "salt as the substance sodium chloride." If then we analyze a text containing the word "salt," and by any formal method select for that word token the formula whose head is STUFF, we will, by that process, have selected the "salt as the sodium chloride" sense for that occurrence of "salt."

The marker names used here are Anglo-saxon monosyllables for purely mnemonic reasons. Marker names more familiar to linguists (such as "human," etc.) will do just as well except that they take longer to read and type.

But we also need to express more complex structures than senses of words, such as the meanings of sentences (and so of texts of any length) in order to provide a representation from which an ambiguity resolution can be read off in the way described earlier. Anyone who has ever tried to understand a sentence, in a language he does not know, with the aid of only a dictionary and grammar book, will have probably realized that the meaning structure of a sentence *cannot* be simply a list of word senses, nor even a list of word senses together with a grammatical structure. If that is so, then a device worth trying as a way of representing meaning structure is that of message forms, or *templates.* These are semantic patterns which pick up only certain permitted structurings of word senses from coded texts. Templates are not simply lists of senses but can be interpreted directly as the content of utterances. So, for example, if we were analyzing a left-right sequence of formulas, each representing some sense of some word, and the heads of these formulas in left-right order were MAN BE KIND, then we could say that we had attached to that sequence of formulas the template MAN + BE + KIND, which can be interpreted directly as "a human being is a certain kind of human being." We would expect to detect that template in the analysis of utterances like "My father is over-bearing," "The Pope is Italian," and "The postman is happy in his work," because in each case the message expressed could be said to be "a human being is a certain kind of human being." The use of templates, or message forms, does not require any support from psychological speculations as to how human brains actually process language (even though there is some evidence that people operate not so much with single words as with the "gists" of longer pieces of text). Templates are used here only as experimental devices in their own right.

Matching templates onto lengths of text can resolve some word-sense ambiguity even without further processing, for it can eliminate certain unacceptable combinations of senses. Consider, for example, the sentence, "The local policeman is a good sport really." Whatever is meant by that sentence, it is *not* the message that "a certain kind of human being is a certain kind of recreational organization." Therefore, if in an inventory of templates there was none that could be interpreted as "a human being is a recreational organization," then that particular combination of senses could never be picked up, even though it is a possible combination on the basis of a sense dictionary alone. This sort of restriction on sense combination produces effects similar to Katz and Postal's [1] "projection rule" method.

As expected, short lengths of text, in isolation from more text, remain ambiguous with respect to templates. Consider a sentence like "The old salt is damp." In British English that sentence allows two quite different interpretations: "a certain kind of human being is in a certain state," and "a certain kind of chemical substance is in a certain state." If we suppose that all semantic formulas corresponding to senses about sorts, types, and states have KIND as their head marker, then the two interpretations of the sentence can express interpretations of the templates MAN + BE + KIND and STUFF + BE + KIND, respectively. And until we know whether this sentence is part of, say, a sea story or a laboratory story we cannot decide which template to assign to it.

However, further ambiguity resolution is possible within the compass of a single template, provided that the formulas containing the template markers as their heads can be related to the formulas for certain other words within the sentence (or part of a sentence) under examination. So, to go back to "The old salt is damp" example, one would expect a generally applicable rule eliminating from further consideration the formula for the "collective noun" sense of "old"; as in "The old must be given increased welfare payments." For "old" in the example sentence has its qualifier, or adjectival, sense which might well have KIND as the head of its formula, just as the qualifier formula for "damp" does. Now suppose the other sense of "old" under discussion is coded by a formula with FOLK as its head, where FOLK is a marker used to code words meaning human collectives of any sort. Thus, having matched both MAN + BE + KIND and STUFF + BE + KIND onto "The old salt is damp," we look to see if either template can be expanded to pick up the correct sense of any other words in the sentence. And the natural rule would select a formula with head KIND (as a qualifier for *either* sense of "salt") in preference to one with head FOLK. By "expanding a template" I mean not only the recognition of the appropriate neighboring formula but also the stringing together of such formulas with those of the bare template to form a larger entity, called a *full template,* that represents more words of the text. I shall describe this process of expansion in more detail below. In this case "old" is resolved by the expansion of either template distinguished above, though this resolution does not also select the correct template for the whole sentence, which is still coded by two representations.

It will already be clear that the method of analysis I am describing is not based essentially on a grammatical analysis, as are a number of other systems of semantic analysis [1]. The present system takes the notion of meaningful, rather than grammatical, language as the basic one, and it attempts to attach semantic frames, the templates, directly to text. I shall describe below (Section 4) a method of *fragmenting* input texts at the start of an analysis, so as to have a unit of text to which to attach the templates. This procedure is not far removed from a simple syntax in the conventional linguistic sense, but it is an essentially dispensable procedure. Moreover, there is a sense in which the present system tries to do some of the work of a conventional syntax directly by semantic means, not only by the restrictions on sense combination imposed by the structure of the template itself, but also by procedures like the one I described above where the "plural noun" sense of "old" was rejected in favor of the "qualifier, or adjectival" sense. After all, if we can decide that a piece of text expresses the message "a human being is a certain sort of human being," then we already know, from that alone, that it contains the part of speech sequence Noun + Copula + Adjective (should we want to know such a grammatical fact for any other purpose).

Nor do I want to draw parallels between the templates and what are usually called "deep structures"; largely because any linguistic structure, deep or otherwise, must in the end be assigned to a piece of text on the basis of the actual superficial word-shapes it contains. It is not easy to see why some structures assigned on that basis are "deeper" than others. The only useful connection between templates and deep structures is that they share a common intellectual origin in the old notion of common "logical forms" underlying different forms of words. The present system in fact grew out of coding systems for mechanical translation developed at the Cambridge Language Research Unit by Masterman [2], and the contemporary work it is closest to is that of Simmons and Burger [3] and Quillian [4].

The task of ambiguity resolution is by no means finished when templates have been assigned to the fragments of a text. More than one template may still be attached to some text fragment, and the remaining problem is to reduce this so that one and only one template attaches to each text fragment. A whole text is then represented by a string of templates, and the desired representation for the purpose of ambiguity resolution has been achieved.

The solution to this problem, naturally enough, is to specify rules that relate templates together to correspond to a "proper sequence" of text fragments (though not necessarily a contiguous one). Suppose we consider the text "The old salt is damp, but the cake is still dry," where one would naturally assume that the correct sense of "salt" is in the "salt as sodium chloride" sense. So, if the two templates discussed earlier were both possible message forms for "The old salt is damp"; and, let us suppose, STUFF + BE + KIND is the only one matching with "the cake is still dry," then for the whole sentence there would be two possible template sequences:

MAN + BE + KIND
STUFF + BE + KIND and STUFF + BE + KIND
STUFF + BE + KIND.

In the absence of any overriding considerations, a rule of template sequence could take the second (and correct) sequence in preference to the first on the basis of the repetition of the marker STUFF. This example is, of course, an absurdly oversimplified case of the sort of coherence and repetition of ideas that almost certainly has to be present in written and spoken language in order for it to be understood. By "proper sequence of text fragments," I mean a sequence that allows a single interpretation to be imposed by rules of this sort. It is easy to construct examples of fragment sequences for which it would be very difficult to impose a single reasoned interpretation on the whole, because the constituent fragments lack this coherence: "I stepped on a train, and won a case yesterday," for example.

This coherence between text fragments need not always be expressed by simple *repetition* of markers, nor does it involve only the *heads* of the formulas, as does the last example. One would expect the same resolution of "salt" as in the last example in the sentence "The old salt is damp but the biscuits are still dry." Yet here, biscuits are not a substance, or stuff, like cake; they are things, or individuals. So one would expect the formula for the appropriate sense of "biscuit" to reflect that fact by having, say, the marker THING as its head. In that case the correct sequence of templates would be

STUFF + BE + KIND
THING + BE + KIND,

which could not be selected by mere repetition of heads alone, since the heads that are repeated, BE and KIND, are not those relevant to the resolution of "salt." At this point the selection rules operate with the notion of the "negation classes" of the semantic markers. Roughly speaking, that notion relates each marker to a class of other markers that are "semantically close" to it in some way. So STUFF and THING would be more alike (each would occur in the negation class of the other) than would be MAN and THING. So, working with this form of preference, the correct sequence above would be selected.

Very little of interest could be done with the heads of formulas alone, as the examples so far have been. The analysis actually works almost entirely with the whole formula picked up by the template pattern. By matching the *bare template* MAN + BE + KIND, say, onto a text fragment, what is actually picked up from the text in the process is a formula whose head is MAN, followed by a formula whose head is BE, followed by a formula whose head is KIND.

Now consider "The old salt is damp though the bed was properly prepared." The most plausible interpretation contains the "salt as an old sailor" sense, which requires, let us suppose, the template sequence

MAN + BE + KIND
THING + BE + KIND.

But from what has been said about negation classes one would not expect rules using them to select this pair of templates in preference to the other pair corresponding to the "salt as sodium chloride" sense (which would contain the head STUFF in place of MAN); since MAN is not as "semantically close" to THING as STUFF is, Hence the whole of the semantic formulas for the senses of "salt" and "bed" would have to be examined at this point; in particular we would expect some indication in the formulas for "bed as an object for sleeping on" that it is *for human beings,* and so there would be some repetition of the marker MAN, in the "bed" formula and as the head of the formula for "salt." Thus, a rule picking up this overlap would be expected to override the one using the weaker negation classes.

I said earlier that the above interpretation might seem to be the more likely one for the sentence, because anyone could conceive of another interpretation, based perhaps on a dictionary meaning for "bed as part of a garden." There might then be a weak (negation class) overlap between the template matching onto this sense and one matching onto the "salt as sodium chloride" sense earlier in the sentence. Unless we had a rule to prefer the template pair with the overlap of MAN markers, we would then have two alternative template pairs for the sentence, and it would remain ambiguous in isolation from more text (with one interpretation corresponding to sailors at rest and one to gardening activity). The latter pair might eventually be selected if the sentence were embedded in a longer narrative about the soil, and we had a technique for reapplying the rules connecting templates together in a recursive manner, so as to end up with only a single string of templates matching a whole text. In the present system this is done using the Cocke Algorithm: the rules relating templates are applied first to pairs of contiguous templates (those matching fragments adjacent in the original text) and then to noncontiguous pairs. Rules are provided for constructing a single composite item for any pair of templates related in this way, and that item can then participate in rewritten strings. This is all precisely analogous to the rewriting of NP + VP as S in a conventional phrase structure grammar.

It is to be expected intuitively that a coherent text can be matched to a single representation in some way like this, for writers who are not poets or philosophers by profession usually go on writing until their meaning is clear, until there can only be one generally acceptable interpretation of what they are saying.

If a pair of fragments of text are such that each has some template representation—and there is some pair of templates, one matching with each of the fragments, related together by overlap of content in some way like those I have described—then I shall call the fragments *semantically compatible.* So, for example, "The old salt is damp but the cake is still dry" would consist of two semantically compatible fragments. The system to be described in this paper generates templates for text fragments and then seeks to apply the rules of semantic connection between the possible chains of templates that can be formed for the whole text. It seeks to apply the rules first to pairs of contiguous fragments and then to noncontiguous pairs. Replacements are constructed for pairs with sufficient overlap, and the rules are then applied recursively using the Cocke algorithm to try and rewrite the strings of templates down to a string with one member, which will be P, the "paragraph symbol," or left-hand side of the "topmost phrase structure rule" in the system of analysis. If this can be done for a given string of templates, the string is considered to be a *proper sequence of templates* and a semantic representation for the text in question. An ambiguity resolution can then be read off from the string in the way described, and, if there is only one such string for the text, the text will be resolved. In representing the system of analysis as a set of phrase-structure rules, the objects of the rules will not be syntactic categories but objects like templates, semantic formulas, paragraph symbols, and so on. However, the operation of the system is exactly like that of a phrase structure parser, and the resulting interpretation can be thought of as a parsing of the fragments of a paragraph, just as the grammatical analysis of a sentence can be thought of as a parsing of the words constituting the sentence.

A word of warning is necessary about the odd nature of examples in the field of ambiguity resolution. It is an important fact about a natural language like English that there are *no* examples of ambiguity resolution that are *beyond question.* Consider, for example, "The bar was shut," which is clearly ambiguous as it stands; it is not clear whether the sentence concerns a barrier or a drinking place. If that sentence is now embedded in "The bar was shut because the barman was sick," then most speakers of English would agree that the sentence was about a bar to drink in. But, even so, that unanimity would be a matter of luck. It could never be put beyond question, for it would always be possible for someone to embed that sentence in some odd larger story text; possibly one about a man who tended a bar for a living but who also had some kind of apparatus which he opened and shut across his driveway whenever he went in and out. There is no solution to the general difficulty raised by this example, and I mention it only to try and keep the discussion of what follows away from carping about examples. It should be possible to assess the output from any ambiguity-resolution program without any knowledge of the system used, but agreement among the assessors will always depend upon common sense and goodwill, however vague those notions may be. For absurd stories can be conceived to refute any suggested resolution.

This fact, if it is one, has important philosophical implications about language, though this is not the place to discuss them [5] One practical implication for the construction of a system of semantic analysis is that there must be some provision for the situation where a given body of rules *fails* to assign any interpretation to some text. This failure cannot be taken to imply that the text is therefore meaningless. No semantic dictionary, even if it contains all the senses specified in the Oxford English Dictionary, can be said to exhaust the possible ways of using the words in the language. It would always be possible to make up a story of the sort described above, which would have the effect of forcing some new sense onto a word, and yet the whole utterance would still be comprehensible to a reader. We all know of poetry that is perfectly comprehensible yet containing words used in senses not specified in any dictionary. Nor is this a phenomenon limited to poets and perhaps philosophers. I have no doubt that I am using "ambiguity" in a nonstandard sense in this paper, yet that need not confuse a reader at all.

One implication for a computable system of analysis is that it should contain some facility for dealing with this situation. As Bolinger puts it, "A semantic theory must account for the process of metaphorical invention. . . . It is a characteristic of natural languages that no word is ever limited to its enumerable senses" [6].

The present system contains an attempt to provide such a facility, albeit a sketchy and tentative one. It is called a *sense constructer* and is an interactive procedure brought into operation whenever the system cannot produce a resolution. It works in an on-line mode under the control of a human operator at a teletype. The system makes suggestions to the operator as to how the dictionary could be augmented, with an additional sense representation for a word, in such a way that a resolution might be produced. The operator can reject the proposed extension of sense on the grounds that it is unthinkable that such-and-such a word could ever be used to mean so-and-so, but if he does not, the text analysis is tried again with that possible sense explanation added into the sense dictionary. In making the suggestions the sense constructer assumes that there is sufficient coherence, in a broad sense, present in the text under examination to force a sense onto a word—either a new original sense, or simply one that the dictionary maker has forgotten to put in. In certain cases its use has been very successful, as I shall describe in more detail below.

2. The Semantic Dictionary

The dictionary consists of a set of *sense pairs,* each one corresponding to some sense of some natural language word. The dictionary items can be thought of as being tied by many-one relations to natural language words outside the dictionary, and at present most of the words considered are tied to only two or three of their main senses. A sense pair is a list of two members. The left member is a semantic *formula,* which is itself a list of semantic markers nested to any level and whose last (rightmost) marker is its *head.* An example would be

(((THIS POINT)TO)SIGN)THING).

The right member of a sense-pair is a *sense-description* which serves only to explain to an operator, in ordinary language print-out, which sense of which word is being operated upon. For the above formula the corresponding right-hand member would be

(COMPASS AS INSTRUMENT POINTING NORTH).

The sense-descriptions are not used as data for computation, except for looking at the first item to get the name of the word in question.

The formulas are constructed by a dictionary maker and their purpose is to encode, and so distinguish, the different senses of natural language words. Formulas consist of left and right brackets, and markers, drawn from the following list: *BE* BEAST CAN *CAUSE CHANGE* COUNT *DO* DONE *FEEL FOLK* FOR FORCE FROM *GRAIN HAVE HOW* IN *KIND* LET LIFE LIKE LINE *MAN* MAY MORE MUCH MOST ONE *PAIR PART* PLANT *PLEASE* POINT SAME SELF *SENSE SIGN* SPREAD *STUFF THING* THINK *THIS* TO TRUE UP *USE WANT* WHEN WHERE *WHOLE* WILL WORLD WRAP, or any of those markers immediately preceded by NOT.

It is very difficult to justify such an inventory on theoretical grounds, and if anyone asks for a discovery procedure for either the markers or the detailed semantic codings, then he is making a conceptual mistake. There cannot be such a thing, and no worker in the field has even offered one. The interesting question is, given some systematic semantic coding, what can then be done with it? I shall assume here that one has to *choose* some set of markers to work with, and anyone's set of markers is always open to detailed objection [7]. The markers are the basic elements in terms of which the others in this system (templates, formulas, etc.) are defined, so they cannot themselves be further defined, except by means of a table of notes which gives the dictionary maker some indication of the intended scope of the markers. The table contains entries like:

GRAIN:	(II, IV, VI)	any kind of structure or pattern
	(III)	structural or pattern-like.

The Roman numerals refer to the six *bracket types* used by the dictionary maker in constructing formulas. They are, in order, Adverbial Group, Adverbial Clause, Adjunctive Group, Nominal Group, Operative Group, Operative Clause. The first two, for example, can be illustrated as shown below:

I. Adverbial Group:
((TRUE MUCH) HOW)-equivalent for "enough" used as an adverb; same function as "rather nicely" in English; can end only with marker HOW.

II. Adverbial Clause:
(MAN FROM)—same function as "to the end" in English; cannot be a well-formed formula (see below) by itself.

Every bracket pair, whether of a pair of markers alone or one with nested subparts, can be assigned to one of these six types. Thus, in the formula exemplifying bracket type I above, ((TRUE MUCH) HOW), both the inner and outer bracket pairs are of that type. Every bracket pair, however complex, is a binary bracketing with a left-hand member that is dependent on the corresponding right-hand member. This is the less intuitive order in LISP but is a more natural way of reading formulas for English speakers; the usual dependence relation being "leftmost on rightmost" in English.

The interpretation of this dependence relation varies with the bracket type. In type IV, the Nominal Group, it is in effect the straightforward attribute-value relation [4]; as in (WHERE POINT) used to mean "a spatial point." However, in the Adverbial Clause illustrated above as type II, the dependence of MAN on FROM is more like that of the object of a preposition on the preposition. Whatever the interpretation of the relation, the related parts can both be nested to any depth. To take a sense pair at random, say, (COLORLESS ((((((WHERE SPREAD) (SENSE SIGN)) NOT HAVE) KIND) (COLORLESS AS NOT HAVING THE PROPERTY OF COLOR)))). An explanation of the formula would be: "colorless" is a sort; a sort indicating that something does not possess some property; the property is an abstract sensuous property of a certain sort; that certain sort has to do with spatial distribution. And it is not difficult to see that that is what (in right-left order) the formula conveys. Inside that formula ((WHERE SPREAD) (SENSE SIGN)) is itself of type IV, (Nominal Group), as are both of its subparts. So a type IV bracket can be made up of two type IV brackets; just as a noun phrase in English, such as "corn stalk" or "power tool," can be made up of two nouns.

The table of notes therefore contains not only restrictions on which markers can participate in which bracket types but also restrictions on which bracket types can

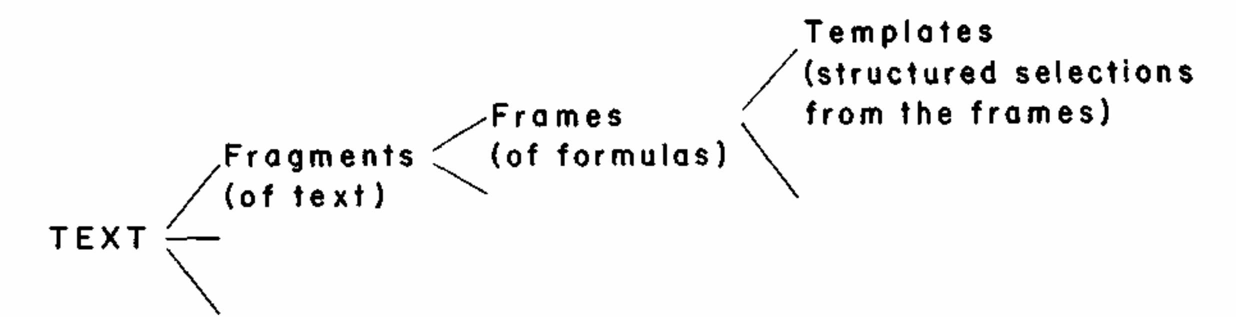

FIG. 2.—Attachment of text to templates

participate in which other bracket types. From what has been said so far it follows, for example, that type IV can occur inside itself. Type II, however, cannot occur inside itself. It will also be clear, from the example of the table format given above for the marker GRAIN, that the markers cannot be exclusively assigned as either items or properties of items. GRAIN can occur in type III as a property, "structural," and also in type IV to stand for the item "structure." In all bracket types the rightmost markers is its head. However, only certain markers can be the heads of *well-formed formulas;* that is, formulas that can be the left member of sense pairs encoding the senses of words. The possible heads of well-formed formulas are those markers italicized in the original list of markers given above. They indicate the major categories of word-sense classification; though this list, too, can only be justified intuitively. Since HOW is not italicized, and since type II can have only HOW as its head, it follows that a type II bracket can never express a word sense. I can summarize with recursive definitions of formula and well-formed formula:

1. A *formula* is a binarily bracketed string of formulas and atoms.

2. *An atom* is a marker, or a marker immediately preceded by "NOT." It follows that a single marker is not a formula.

3. *A well-formed formula* (wff) is *(a)* a formula, and *(b)* such that its head is one of the following markers: HOW KIND FOLK GAIN MAN PART SIGN STUFF THING WHOLE WORLD BE CAUSE CHANGE DO FEEL HAVE PLEASE PAIR SENSE WANT USE THIS.

3. The System of Semantic Analysis

The present system starts an analysis by replacing each fragment of a text by all possible strings of formulas *(frames)* constructed from the formulas for the words of the fragment. It then searches each frame and replaces it by a number of matching templates, or meaning structures. One can display these initial procedures schematically (see fig. 2). In the course of these procedures each fragment of text is tagged to a number of templates, and so each such template is tagged to *some particular* selection of the word-senses for the words of a fragment. The purpose of the subsequent procedures is to reduce this "fragment ambiguity" by specifying a set of strings of these templates, one template corresponding to each text fragment, and so specifying resolutions for the words of the whole text. The intuitive goal is that there should be just *one* string of templates in that set, and hence a unique ambiguity resolution of the text. However, the possibility of a number of independent resolutions cannot be excluded a priori.

The procedures of resolution can be expressed as a set of phrase-structure rules which produce a nesting of frames of formulas from an initial paragraph symbol P. There are rules producing bare templates, the simple concatenated triples of head markers described in the introduction above; others expanding these bare templates to full templates containing formulas; and yet others producing pairs of related full templates from single full templates. The dictionary of sense pairs can also be put in the form of rules like W → fn, where W is a word name and fn a formula for some sense of that word. Taken together, these rules could theoretically generate a text from a nesting of full templates, which was itself generated from the paragraph symbol P.

However, the generative forms are no real guide to the analysis algorithms; all they do is ensure in advance that the system is computable (the rules are set out in full in [8]). In this section I shall describe the procedures as they are applied in the process of semantic analysis.

MATCHING BARE TEMPLATES ONTO FRAGMENTS

I shall assume that a text under analysis has been fragmented in some determinate manner and that from it and the semantic dictionary a number of *frames* of formulas have been constructed. Each frame is a string of formulas such that each word in the fragment that has a nonnull dictionary entry is represented in the frame by one and only one formula, which has the same linear order in the frame as the corresponding word has in the fragment. There will, therefore, be a frame for every possible combination of word senses for a fragment of text and a dictionary.

The possible triples of markers that constitute bare templates are defined in a *standard order:*

Substantive (or noun) type marker from a class N1 + Active (or verb) type marker from a class V + Substantive marker from a class N2.

The rules also produce nonstandard orders of templates such as V + N1 + N2 and N1 + N2 + V as well as *debilitated templates* such as N1 + N2, KIND + N1, N1 + V, and N1 by deletion rules. A fragment is said to *match* with templates if a frame for it contains a concatenation of heads corresponding to any bare template, whether standard, nonstandard, or debilitated.

The templates actually produced by the rules are certainly motivated by psychological and related considerations about what people can possibly say, for example, MAN + HAVE + PART can be produced by the rules, but MAN + B + WORLD cannot. But here they should be considered simply as analytic devices in their own right. Now, in order to produce matches with templates that can plausibly be interpreted as meaning structures for fragments—in that they correspond to heads and frames for the appropriate word senses in a fragment—it is necessary that classes of templates be preferred in a rank order. There are four such ranks. The standard order N1 + V + N2 occurs in the first rank along with some nonstandard and debilitated orders such as KIND + N1. The lower ranks contain progressively more debilitated forms. If the matching algorithm finds a rank I template form in a frame it does not look for lower ranks, and so on down the order of ranks.

The rank choice enables much of the work of a conventional grammar to be done by template matching. An example should make this clear as well as explain the presence in the first rank of a debilitated form of template like KIND + N1. Consider the fragment "The old transport system," and for simplicity let us consider only two frames of formulas for it: (1) the frame consisting of the formulas for the *appropriate* senses of the words in that fragment, and (2) the frame identical with the first except that it contains representations of "old" as substantive (noun = "the old people") as well as the active (verb) form of "transport." So, by the semantic coding system described above, those two frames will contain the following heads in order for the words "old," "transport," "system," respectively: (1) KIND, KIND, GRAIN, and (2) FOLK, DO, GRAIN. Now the rules of template production permit both FOLK + DO + GRAIN and KIND + GRAIN in rank I, the latter by transposition and deletion from N1 + BE + KIND and KIND + N1. If the form KIND + N1 were *not* in the first rank, along with the forms like N1 + V + N2, which yields FOLK + DO + GRAIN, then a phrase like this one would never get the correct interpretation, which must contain both the sense of "transport" whose formula head is KIND ("transport" being an adjective in this fragment), and the sense of "old" whose formula head is KIND ("old" also being an adjective in this fragment). If KIND + N1 were *not* in rank I, then the matching routine would match FOLK + DO + GRAIN onto the fragment via the second frame and never look any further for debilitated forms; and in doing so it would have got the wrong senses of "transport" and "old."

In the LISP implementation, the matching of bare templates is done by a function named TEMPO, which takes as its argument a frame of formulas, one for each word of a fragment. TEMPO scans each such combination in turn, starting with the frame containing all the main senses of the words. TEMPO searches for triples of heads in the order of preference given by the rank table, and each type of template is collected on a list which is the value of a different free LISP variable. If TEMPO finds nothing till it reaches the debilitated N1 + N2 or KIND + N1 form, it replaces N1 + N2, by N1 + BE + N2 (BE being the "dummy verb") and transposes KIND + N1 as N1 + BE + KIND. Similarly V + N1 and N1 + V are replaced by THIS + V + N1 and N1 + V + THIS, respectively (THIS being the "dummy substantive"). The function of these dummy features is to give a general form of template for subsequent processing, even when it is not wholly present in the text. Consider another fragment that is not in an assertion form, but is again a noun phrase, say, "the black wizard." The heads of the appropriate formulas for "black" and "wizard" would be KIND and MAN, respectively. As there is no verb, a debilitated template of the KIND + N1 form would match onto these two heads, and that would then be converted into MAN + BE + KIND, which is the intuitively correct interpretation. The dummy verb is added in the way described; and in cases where the first head is the predicate KIND, the order of the two heads is reversed to give the MAN + BE + KIND form. In the "old transport system" case discussed earlier, the debilitated form KIND + GRAIN will match onto both "old + system" *and* "transport + system." It will be converted twice with the dummy verb to the standard form GRAIN + BE + KIND. That template can be interpreted as "a structure is of a certain sort," and is a very general representation of both "a system is old" and "a system is for transport." So far, then, the fragment "the old transport system" has been matched with two different bare template types, GRAIN + BE + KIND and FOLK + DO + GRAIN, since they were both in rank I, and there is no reason to prefer one to the other at this stage. But the fragment has matched with three bare template *tokens.* This can be represented schematically as follows, with the matched fragment words under the appropriate formula heads that make up the three template tokens:

FOLK +	DO	+ GRAIN
old	transport	system
GRAIN +	BE	+ KIND
system	(is)	transport
GRAIN +	BE	+ KIND
system	(is)	old

As I noted in the introduction, what has actually been picked up from the frame by the bare template matching

```
((THE OLD TRANSPORT SYSTEM)
  ((FOLK DO GRAIN)
   ((((MUCH WHEN)FOLK) (OLD AS OLD PEOPLE))
  ((((THING FOR) (WHERE CHANGE))DO) (TRANSPORT AS MOVE ABOUT))
  ((WHOLE GRAIN) (SYSTEM AS AN ORGANIZATION))))
  ((GRAIN BE KIND)
    ((WHOLE GRAIN) (SYSTEM AS AN ORGANIZATION))
    ((BE BE) (DUMMY))
    (((MUCH WHEN)KIND) (OLD AS HAVING BEEN THROUGH MUCH TIME))))
  ((GRAIN BE KIND)
    (((WHOLE GRAIN) (SYSTEM AS ORGANIZATION))
    ((BE BE) (DUMMY))
    (((THING FOR) ((WHERE CHANGE)KIND)) (TRANSPORT AS PERTAINING TO
                          MOVING THINGS ABOUT)))))
```

FIG. 3.—Bare template output for a fragment

procedure is a triple of *formulas,* whose heads correspond in left-right order to some permissible bare template. If the bare template matching is output in LISP, it looks as shown in figure 3 for that fragment.

This list of three bare templates is only part of the value of the LISP function TEMPO with the fragment name as its argument, because for the purposes of this example certain word senses and combinations of them have been ignored. Each major item in the above list is a bare template tied to the three formulas which have heads corresponding to its member markers.

MATCHING FULL TEMPLATES ONTO FRAGMENTS

The full templates are the items with which the system really operates, and they are derived from bare templates by looking at the *remaining* formulas in the frame, that is, more than the three in the bare template output above. A full template is not a triple of formulas but a sextuple; it is the three formulas associated with the bare template plus the formulas which precede those bare template formulas in the frame. Any of these latter may be absent and will then be represented by LISP NILs.

The function which matches full templates is called PICKUP; it takes as its argument a fragment name and immediately derives a list of possible bare templates like the one above. It then looks back at the frame of formulas for each bare template to see if the formula preceding each formula in the bare template can be a proper qualifier for it. A discussion of why preceding formulas should be expected to be qualifiers must be delayed until the description of the initial fragmentation procedure in Section 4 below.

So PICKUP looks first at FOLK + DO + GRAIN, which are the heads of formulas for "old," "transport," and "system," respectively. In no case is there any qualifier formula in the frame that is not already in the bare template, except one for the vacuous "The." In the frame for the first GRAIN + BE + KIND form, there is the qualifier formula for "transport" whose head is KIND, but no other qualifier not already in the bare template. I say *qualifier* because that sense of "transport" has head KIND and precedes a nounlike formula (for those who like to think in conventional grammatical terms) whose head is GRAIN. This is a *form-closeness,* and PICKUP keeps a score of these as it turns each bare template into a full one. It also counts verblike formulas preceded by adverblike ones, adjectivelike formulas preceded by adverblike ones, and so on. It also scores one for the form N + BE + KIND where N is a nounlike head, as GRAIN is. So then, PICKUP can score from 0 to 4 for any template; up to 3 for the predecessors of the heads, and 1 for the N + BE + KIND form. In this case it will score 0 for FOLK + DO + GRAIN; 2 for the first GRAIN + BE + KIND; and only 1 for the second GRAIN + BE + KIND, since the KIND sense of "old" is not a proper qualifier for the KIND sense of "transport" (i.e., adjectives do not qualify adjectives in English).

As well as keeping this score, PICKUP builds up a full template form by adding on to the bare template those formulas that are qualifiers in the required sense. The full templates for the first and third of the above bare ones will be just the same as the corresponding bare ones except for three NILs inserted to mark the absence of any of the three possible preceding qualifiers. In the case of the second bare template, PICKUP will build up the item

```
((GRAIN BE KIND)
 (((WHOLE GRAIN) (SYSTEM AS AN ORGANIZATION))
  ((BE BE) (DUMMY))
  (((MUCH WHEN)KIND) (OLD AS HAVING BEEN THROUGH MUCH TIME))
  (((THING FOR) ((WHERE CHANGE) KIND)) (TRANSPORT AS PERTAINING
  TO MOVING THINGS ABOUT)) NIL NIL)).
```

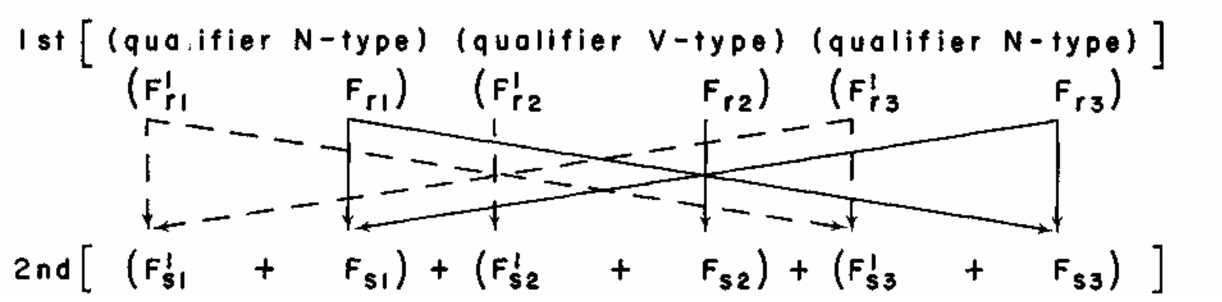
1st [(qua.ifier N-type) (qualifier V-type) (qualifier N-type)]
(F'r1 Fr1) (F'r2 Fr2) (F'r3 Fr3)
2nd [(F's1 + Fs1) + (F's2 + Fs2) + (F's3 + Fs3)]

FIG. 4.—Connecting pattern between full templates

The fourth formula is the proper qualifier for the first, and, if such had been found for the second and third, they would have appeared in place of the NILs in the fifth and sixth places, respectively.

Inside PICKUP the function REFINE returns as its value a list of five sublists of full templates. Its first sublist contains those form-close internally in *four* ways, down to the last sublist containing those with no such closeness. PICKUP takes the first nonempty sublist of REFINE, and of that list returns as its value the list of full templates that are content-close as well (if any). What is meant by content-close is analogous to form-closeness. Two formulas are said to be *content-close* if (1) they share a common pair of markers; or (2) they have one or more of the following elements in common: ONE, COUNT, WORLD, WHOLE, LIFE, LINE, MUST, SELF, SPREAD, TRUE, WRAP, WHEN, WHERE, THINK; or (3) their cores are such that they are identical, or either is a member of the other in the sense of a list member, or the left- or right-hand member of either core is a member of the other.

Again, there is and can be no *theoretical* rationale for the list in (2). It is simply an empirical observation about the way the markers are used that, if two formulas both contain the marker COUNT, that fact is more likely to locate correct word senses than if they both contain MAN. The *core* of a formula is simply its subpart that depends directly on the head; so it will be a marker in a simple formula, but in a formula like (((WHERE POINT) FROM) SIGN) it is ((WHERE POINT) FROM).

In the example considered earlier, PICKUP will select the full template set out on page 67 in preference to the other two on grounds of its form-closeness score alone. Content-closeness is only examined when there is more than one full template with the highest available form-closeness score.

THE "SEMANTIC PARSER": RESOLVING A PARAGRAPH

The procedures considered so far have rejected possible interpretations for fragments in two ways: first, by matching preferred classes of bare templates onto coded fragments; second, by preferring interpretations that can be expanded to fill the coding frame as fully as possible and with as much content connection as possible. All these I call *internal* rejection procedures, in that they operate over the span of single text fragments and may still leave a fragment tied to more than one full template.

The remaining, *external,* rejection procedure spans texts consisting of a number of fragments. It seeks for closeness relations between the markers of full templates matching onto different fragments. These closeness relations are somewhat weaker than the content-closeness defined within a full template in that they also make use of the weaker *negation-class inclusion* between markers, discussed in the Introduction. Moreover, these relations do not simply establish preferences, as with the full template matching; they are used to provide a criterion of closeness between a pair of full templates, which any actual pair may or may not satisfy.

If we think of a full template reordered more naturally so that each qualifier formula precedes the formula it qualifies, and consider it symbolically as the string of six formulas:

$$S = [F'_{sl} + F_{sl} + F'_{s2} + F_{s2} + F'_{s3} + F_{s3}],$$

then the ten directions of connection between the formulas of the two templates R and S can be illustrated schematically as shown in figure 4. If this form seems unnecessarily abstract, one can refer back to the full template form on page 67. There the six formulas are in the order

$$[F_{sl} + F_{s2} + F_{s3} + F'_{s1} + F'_{s2} + F'_{s3}],$$

with the qualifiers (primed) placed after the main template formulas. Two full templates are considered to be *semantically close* if (with the above notation for full templates) *at least three* of the following pairs of formulas are such that (1) the head of the second is identical with, or in the negation class of, the first:

$$(F_{rl}F_{s1}), (F_{rl}F_{s3}), (F_{r2}F_{s2}), (F_{r3}F_{s1}), (F_{r3}F_{s3})\ ;$$

(2) either they, or their qualifier formulas, are content-close.

If, for any pair of full templates, three or more of these connectivities are present, then a new templatelike item is constructed from the two full templates. This item replaces the pair in the paragraph-length string of full templates under examination. Then the shorter string is reexamined using Cocke's algorithm for other pairs of semantically close templates. Contiguous pairs of templates are examined before noncontiguous pairs.

A successful resolution is reached when a string of templates has been rewritten down to a single item. If that can be done, the original string of full templates is said to be a proper sequence, and it is deemed an ambiguity resolution for the text examined. Each possible string of full templates is examined in turn until all have been examined.

The top level function for this is called PARSPARA, and its main subfunctions are called FIT and JAM. They test for semantic closeness between two full templates, and, if such closeness is found, the two full templates are replaced by a single item with the form of a full template. Or, to put it in terms of the two function names, if the full templates FIT, they are then JAMmed. The function JAM actually builds up a representation of the two templates containing the content-close parts. FIT and JAM work with *message-pairs,* which are to a fragment what a sense pair is to a word.

A message-pair is a two-item list: the right-hand item is some full template, the left-hand item is a list containing the name of some fragment with which the full template matches.

PARSPARA builds up all possible frames of message-pairs for the paragraph from the value of PICKUP. Each frame of message-pairs is now a possible meaning representation for the whole paragraph. PARSPARA then scans each frame in turn to see if it can find a right-left contiguous pair of message-pairs satisfying FIT. If it can, it deletes the first message-pair and replaces the second by a message-pair consisting of (1) the JAM value of the two "parsed" full templates, and (2) a *list* of the names of the fitting fragments. Suppose we consider the sentence "Britain's transport system and with it the traveling public's habits are changing." For reasons discussed below, this sentence is analyzed as the two fragments (BRITAIN'S TRANSPORT SYSTEM ARE CHANGING), (AND WITH IT THE TRAVELING PUBLIC'S HABITS). Now consider the analysis of a text containing this sentence, and so of the *paragraph frame* containing a message-pair for each of these fragments. Let us consider the frame containing the ultimately correct message pairs:

```
((BRITAIN'S TRANSPORT SYSTEM ARE CHANGING)
(((WHOLE GRAIN) (SYSTEM AS AN ORGANIZATION))
((BE BE) (ARE AS HAVE THE PROPERTY))
((CHANGE KIND) (CHANGING AS ALTERING))
(((THING FOR) ((WHERE CHANGE) KIND))
  (TRANSPORT AS PERTAINING TO MOVING THINGS ABOUT)) NIL
                                                NIL)),
```

and

```
((AND WITH IT THE TRAVELING PUBLICS HABITS)
((MUCH ((MAN FOR) (MUCH DO)) GRAIN))
 (HABITS AS REPEATED ACTIVITIES))
 ((BE BE) (DUMMY))
 (((WHOLE FOLK) KIND)
 (PUBLIC'S AS CONNECTED WITH THE WHOLE PEOPLE))
 NIL (((WHERE CHANGE)HOW)
 (TRAVELING AS MOVING FROM PLACE TO PLACE))))).
```

The two full templates in these message-pairs can be seen to be semantically close even without the need of negation-class inclusion. The formulas for "transport" and "traveling" share a common core (WHERE CHANGE), for example; this is the connectivity (F'_{r1} F'_{s3}) in figure 4 above. In addition to this, the three main formulas of the templates have identical heads in corresponding positions.

These two message-pairs are therefore semantically close, and the function JAM builds up a message-pair containing a pseudotemplate like the following, which contains as many of the semantically close items as possible:

```
(((BRITAIN'S TRANSPORT SYSTEM ARE CHANGING)
  (WITH IT THE TRAVELING PUBLICS HABITS)))
  ((((WHOLE GRAIN) (SYSTEM AS AN ORGANIZATION))
  ((BE BE) (ARE AS HAVE THE PROPERTY))
  ((CHANGE KIND) (CHANGING AS ALTERING))
  (((THING FOR) ((WHERE CHANGE) KIND))
  (TRANSPORT AS PERTAINING TO MOVING THINGS ABOUT))
  NIL ((WHERE CHANGE) HOW)
  (TRAVELING AS MOVING FROM PLACE TO PLACE))))).
```

This item then replaces the two message-pairs in the paragraph frame, which thus becomes progressively shorter during the parsing. Other surviving full templates for the fragments in general fail to have sufficient semantic connectivity, and the parsing of their paragraph frames breaks down. This is the *external rejection* procedure. If a parsing finishes so that the paragraph frame has only one pseudotemplate in it, the first item in that message-pair will be a nesting of the *fragment names* which represents a semantic parsing pattern for the whole text. The actual word-sense resolution is read off from the original, unparsed frame and is printed out in the form given in figure 1 above.

4. An Actual Experiment and the Use of the Sense-Constructer

Ten paragraph-length texts were chosen for analysis: five from randomly chosen *Times* editorials (data texts); and five from the works of philosophers—Descartes, Leibniz, Spinoza, Hume, and Wittgenstein. The reason for the choice of this type of material will emerge in the discussion. Each paragraph was stored as a list of sentences on a LISP file, and an alphabetical concordance for the texts was obtained with the aid of standard routines. From this the semantic dictionary was written.

Some of the data texts were assigned a semantic structure and resolution by intuitive methods, so that negation classes for the markers could be derived inductively.

Before the analysis begins, an initial set of functions breaks each sentence of a paragraph into strings of words, and, in certain circumstances, reforms discontinuous substrings into whole strings. The output from this process is a sentence in the form of a list of "sentence fragments," each of which (if it is not a single word) is either an elementary sentence, a complex noun phrase, or a clause introduced by a word from an assigned list, such as a preposition [9]. Each paragraph of text is processed by a function which applies the set of fragmentation functions to each of the sentences of a paragraph in turn, and returns the paragraph as a single list of such substrings, thus obliterating the original sentence boundaries.

It can be seen from the example paragraph above that the functions do not simply segment sentences in a linear manner. They also "take out" certain kinds of clauses from within a sentence and append them as separate substrings. An example of this "taking out" and reforming can be seen in the example sentence ((BRITAIN'S TRANSPORT SYSTEM ARE CHANGING) (AND WITH IT THE TRAVELING PUBLICS HABITS)) discussed above. These are produced from a sentence that originally read "Britain's transport system and with it the traveling public's habits are changing." This sort of breakup leads to an apparent grammatical "howler," namely, a singular subject for a plural verb. But for the purposes of semantic analysis by the present system, that is not a disadvantage; it is more than outweighed by having the text cut into semantically acceptable units [10] for the attachment of templates to them.

The fragmented paragraphs are not passed directly to the template matching procedure but are first processed by a set of reordering functions. These inspect the fragmented output for a paragraph and seek for qualifying phrases beginning with marker words like "of" and "for." These are delimited at their other end by the characters "fo" and are placed as a whole before the word they qualify, as are adjectives before the preceding noun, and so on. Only after this rearrangement are the fragments passed on to the matching functions. The reason for the reordering is that, when a template has been matched with a fragment, the subsequent routines seek the qualifiers of a noun or verb only *to the left of it.* Thus a phrase "a book of rules" goes to the matching routines as "a of rules fo book."

The purpose of the fragment unit is to define a unit of context between the word and the sentence as usually understood. I have not discussed the fragmentation functions in any detail, partly for reasons of space and partly because they are theoretically dispensable. I assume that they represent syntax of the most rudimentary kind and are of no particular interest here.

The five newspaper paragraphs were analyzed without difficulty, as in figure 1 above, but in the case of two of the philosophical paragraphs the system was unable to produce a resolution with the aid of the initial dictionary. In these cases the system returned (NO RESOLUTION ALL PATHS BLOCKED) at the teletype.This situation could arise for a number of reasons: the text fragments did not cohere together sufficiently, a vital word sense had been left out of the dictionary, or a word in the text was being used in a new and original sense. A suggestion for unblocking this is to allow the word dictionary to enlarge itself—to supply an additional sense entry for the word that is holding the procedure up, if it can be found. Such a construction could be thought of, in terms of a system of phrase-structure rules, as adding a new rule, $W \rightarrow F_n$, where F_n is a formula and W a word name, and so shifting to a new extended rule system as the system adjusts to the particular text. So this *sense-constructer* is a rule-changing activity that is itself rule governed, and the system of analysis is not represented by a single set of generative rules but by a constructible series of such sets.

In practice, this *sense-constructer* consisted of PARSPARA examining the value of a free variable BESTPARS each time it failed to parse a frame completely. It stored as the value of BESTPARS the parsing tree containing the template that had been rewritten least by JAM. It seemed a good first guess at the recalcitrant word that it was in the template that cohered least with its neighbors. If all the paragraph frames blocked, PARSPARA would print (CONSTRUCTER MODE) and evaluate a function of no variables called CONSTRUCTER. This function controls all subsequent operations via the READ and PRINT functions at the teletype. CONSTRUCTER looks at the value of the template in BESTPARS and suggests at the teletype that a word in the corresponding fragment has its dictionary of sense-pairs enlarged by identifying the recalcitrant word with the most "content-close" word in the paragraph. If the operator accepts the system's suggestion at the teletype, the system is rerun with the enlarged dictionary in an effort to get a resolution. In such a case (or if none of the system's suggestions are acceptable to the operator), the

```
  (CONSTRUCTER MODE)
  ((NO RESOLUTION ALL PATHS BLOCKED)
  (BEST PARSING CONTAINS)
    (((((KIND SIGN) (ATTRIBUTE AS A PARTICULAR KIND OP PROPERTY)
      ((BE BE) (DUMMY))
      ((SAME KIND) (SAME AS IDENTICAL))
      ((WHOLE (MUST (KIND SIGN)))
        (NATURE AS ESSENCE OR ESSENTIAL PROPERTIES))
        NIL NIL)
  (THE SAME NATURE OR ATTRIBUTE))))
(RECALCITRANT TEMPLATE IS FOR)
(THE SAME NATURE OR ATTRIBUTE)
(CONTINUE YES OR NO)
YES
  (SUGGEST ATTRIBUTE AS NATURE) (SHALL I TRY IT YES OR NO)
YES
  (((IF THERE WERE TWO OR MORE DISTINCT SUBSTANCES)
    ((WORDS RESOLVED IN FRAGMENT)
    ((THERE AS AT A POINT)
    (WHERE AS EXISTED)
    (OR AS DISJUNCTION)
    (MORE AS IN AN INCREASED MANNER)
    (DISTINCT AS DIFFERENT) (SUBSTANCES AS SORTS OF THING)))
    ((WORDS NOT RESOLVED IN FRAGMENT)
    (TWO (((COUNT SIGN) (TWO AS A NUMBER))
    ((COUNT KIND) (TWO AS HAVING THE PROPERTY OF TWOITY)))))
```

FIG. 5.-Dialogue in CONSTRUCTER MODE together with first part of subsequent resolution

system returns to the normal operating mode. This procedure was not called upon for the newspaper paragraphs, but it produced some interesting suggestions in the case of two of the philosophical paragraphs. In CONSTRUCTER MODE, dialogues like those in figure 5 occurred.

5. Discussion

One of the main difficulties in coding for, and evaluating, a system like this one is the necessary vagueness of some of the sense-entries (especially evident in words like "it" and "is"). Nonetheless, I think the present system could constitute a tentative criterion for meaningfulness: a text is meaningful if and only if some system like the present one can resolve it. It is easy enough to propose a necessary criterion on the ground that one needs to be able to tell in what senses the words of a text are being used in order to call it meaningful, though one could not infer that a text was meaningless because a particular system and dictionary rejected it. I have argued at length elsewhere that it is possible also to justify the corresponding sufficient criterion [5]. The establishment of such a criterion would be of some interest in the cases of the five philosophical paragraphs, since it was texts like these that Carnap [11] and the "logical syntax" school generally said could be shown to be meaningless on the basis of a system of analytic rules, through they never in fact constructed such a system. The criterion suggested here would only be one of degree (in terms of the number of applications of the sense-constructer procedure a text required for its resolution). That is perhaps the only acceptable form that a criterion of meaningfulness could take, as there seems something absurd about an attempt to set an absolute bound to the meaningful.

Another speculative interest of the present system might be its application to the speech patterns of schizophrenics. Schizophrenic discourse seems [12] to be meaningful within the boundaries of units of the same order of length as the clause or phrase. The trouble is that these units do not seem to fit together in a coherent way in the schizophrenic's speech pattern. A system of the present sort, which tries to make such items cohere, might conceivably provide a measure of "semantic disorder" in such cases.

A number of connections can be made also between the semantic structure assigned to a text by the present system and that assigned by formal logic. These connections have been investigated in the cases of the five philosophical paragraphs, which have a form sufficiently like the one required by formal logic. These connections are of some interest in view of the almost total neglect of the sense-ambiguity of natural language words by formal logic.

One can, for example, interpret the present system so as to create a notion of "valid *and useful"* argument. It has long been recognized that an argument can be formally valid (and even have true premises) and yet be completely useless. This is usually due to a genuine ambiguity in the argument. For example, the following is perfectly valid: "All kings wear crowns, all crowns are coins, therefore all kings wear coins." And, within the context of each premise, each premise is true. (In the "numismatic world of discourse," for example, the second is true.)

An argument could be deemed "valid and useful" if it is formally valid *and* if the present system assigns to it a consistent and complete interpretation. I am using the terms "consistent" and "complete" in a way similar to Bobrow's [13] use of them: an interpretation is complete if the system assigns an interpretation to each key term in the argument, and "consistent" if *it assigns the same interpretation (word-sense) to every occurrence of a term.* Thus, the argument above would not pass the "usefulness" criterion, since a proper ambiguity-resolver would assign *different interpretations* to the two occurrences of the key term "crown."

Received October 1969

References

1. Katz, J., and Postal, P. An *Integrated Theory of Linguistic Descriptions.* Cambridge, Mass.: M.I.T. Press, 1964.
2. Masterman, M. "Semantic Message Detection for Machine Translation." Proceedings of the International Conference on Applied Language Analysis, London, 1961.
3. Simmons, R. F., and Burger, J. F. *A Semantic Analyzer for English Sentences, SP-2987.* Santa Monica, Calif.: System Development Corp., 1968.
4. Quillian, R. "The Teachable Language Comprehender." *Communications of the A.C.M.,* vol. 12 (1969).
5. Wilks, Y. *Grammar, Meaning, and the Machine Analysis of Language.* London: Routledge & Kegan Paul (in press).
6. Bolinger, D. "The Atomization of Meaning." *Language,* vol. 41 (1967).
7. Sparck-Jones, K. "Semantic Markers." Cambridge Language Research Unit ML.181. Mimeographed. 1965. Obtainable from author, 20 Millington Rd., Cambridge, England.
8. Wilks, Y. *Computable Semantic Derivations, SP-3017.* Santa Monica, Calif.: System Development Corp., 1968.
9. Earl, L. *An Algorithm for Automatic Clause Delimitation in English Sentences.* Technical Report 5.13.64.5. Palo Alto, Calif.. Lockheed Missiles and Space Co., March 1964.
10. Halliday, M. "Some Aspects of the Thematic Organization of the English Clause." RAND Memorandum 5224, January 1967.
11. Carnap, R. *The Logical Syntax of Language.* London: Routledge & Kegan Paul, 1937.
12. Laing, R. D. *The Divided Self.* London: Tavistock Publications, 1960.
13. Bobrow, D. G. "Natural Input for a Computer Problem-Solving System." Ph.D. dissertation, M.I.T., 1965.