Preprint No. 8 Classification: IR 2.4

Interactive Semantic Analysis of English Paragraphs

- Yorick Wilks -

INTERNATIONAL CONFERENCE

ON

COMPUTATIONAL LINGUISTICS

⇓

COLING

1969

RESEARCH GROUP FOR QUANTITATIVE LINGUISTICS

Address: Fack Stockholm 40, SWEDEN

Preprint No. 8 Classification: IR 2.4

INTERACTIVE SEMANTIC ANALYSIS OF ENGLISH PARAGRAPHS

Yorick Wilks
1353, N. Fuller
Hollywood
California 90046
U. S. A.

Abstract.

This paper describes the use of an on-line system to do word-sense ambiguity resolution and content analysis of English text paragraphs, using a system of semantic analysis programmed in Q32 LISP 1.5. The system of semantic analysis comprised 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 within 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 analysed. An interpretation of the system in terms of "meaningfulness" is suggested.

Key words and phrases: semantics, language analysis, interpretation, template, CR Categories: 3.36, 3.62, 3.63.

Acknowledgements.

This work has been supported by grants from H.M. Government Office of Scientific and Technical Information and A.F.O.S.R. (O.A.R.), Washington D.C., U.S.A., administered by the Trustees of the Cambridge Language Research Unit; and also by grant AFOSR F44620 - 67 - C0046 from A.F.O.S.R., monitored by Mrs. Rowena Swanson and administered by the Institute for Formal Studies, Los Angeles, California, U.S.A. The computation was done on the Q32 time-shared on-line system at Systems Development Corporation, Santa Monica, California, U.S.A.

1. INTRODUCTION.

In this paper I describe a system for the on-line semantic analysis of texts of up to paragraph length. It was programmed and applied in Q32 LISP 1.5 to material of two sorts: newspaper editorials and passages of classical philosophical argument. The immediate purpose of the analysis was to resolve the word-sense ambiguity of the texts: to tag each word of 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 tackles texts of up to paragraph length because I take it as 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.

The system attempts to detect semantic forms (which I call templates) directly in coded text, and not by means of a conventional syntax analysis. This restriction sets the present approach apart from the better-known ones. However, an approach like the present one still has to show how to obtain the information contained in a conventional syntax analysis, and I shall do that below. For each paragraph of text examined the system derives a nested structure of the semantic templates, which can be thought of as its semantic representation. As I shall show, it may be necessary for the system to enlarge its own dictionary in an on-line mode in order to obtain such a representation. From a representation, a word-sense resolution of the text is read off and printed out, since the representation contains one and only one sense representation for each constituent word of the text.

The basic item, the template, is intended to express, in coded form, the message content of an elementary clause or sentence. Thus, if we had to analyse the sentence "The old postman is angry", I would expect to match with it a template that could be interpreted as "A certain kind of man is in a certain state". Similarly, if analysing the clause "The wicked wizard", I would expect to match with it a template that could be interpreted "a man is of a certain kind". The main hypothesis of the system of sense analysis is that one can build up a 'proper semantic sequence' of such templates as a representation of "semantically compatible" fragments of text. At the end of the paper I shall discuss the possibility of explicating the difficult notion of "meaningful language". But at the beginning I am assuming that, if a text is meaningful then its parts must cohere together in some structured way, and that "semantic compatibility" might express that way. This working hypothesis will also mean that the word-senses that can participate in such a proper sequence will be the appropriate ones. By "appropriate senses" I mean simply the dictionary word-senses that a translator of the text would wish to distinguish from the inappropriate ones.

By way of example, I shall consider the semantic compatibilities of the fragments of a paragraph to be found in a Times editorial in December 1966. As given below it has been fragmented by functions whose operations I shall describe [illegible], but I shall assume that it is comprehensible as a sequence of twelve items:

*note

```
1    ((BRITAINS TRANSPORT SYSTEMS ARE CHANGING)
2     (AND WITH IT THE TRAVELLING PUBLICS HABITS)
3     (IT IS THE OLD PERMANENT WAY)
4     (WHICH ONCE MORE IS EMERGING)
5     (AS THE PACEMAKER)
6     (AIRLINES LATELY HAVE BEEN LOSING TRAFFIC)
7     (TO MODERNIZED RAILWAYS)
8     (RAILWAYS AT LAST ARE BEGINNING)
9     (TO TAKE SOME CARS
10    (OFF THE CONGESTED SYSTEMS TO TAKE THE WEIGHT)
11    (IF THE NEW IDEAS ARE FORWARD PRESSED)
12    (COMMA THE OF. COMMUTER MOVEMENT AND DORMITORY
       .REA CONGESTION FLOW P.TTERN COULD BE CHANGED))
```

Fig.1. A paragraph in fragment form and it's semantic compatibilities.

Let's now look at possible semantic compatibilities between fragments of the paragraph (marked with braces in the left hand margin of the figure above).

Fragments 1 & 2 are semantically compatible (both essentially assert that a structure is of a certain sort: (1) that a system is changing, (2) that a structure is the public's.) This requires that one takes "to be of a certain sort" in its usual wide logical sense to cover such notions as change and movement:
4 & 5 are semantically compatible (both essentially assert that something is moving in some way).
7 & 8 are semantically compatible (both essentially assert that the railways are near to us in time in some way).
9 & 10 are semantically compatible (both essentially assert that something is taking or removing something).

11 & 12 are semantically compatible (both essentially assert that some structure is changing or about to change).

Notice that semantic parallelisms of this sort between fragments are sufficient to resolve at least one ambiguity in each of the pairs of fragments: for example the correct sense of "habits" for fragment 2 is "structure of behaviour", rather than the less-common "articles of dress". Thus pointing out this parallelism is also selecting the appropriate sense of "habits".

2. THE TEXTS AND SEMANTIC DICTIONARY

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.

The information stored for each dictionary entry word is a list of pairs, each member of which consists of a left-hand member, which is a semantic formula such as (((THIS POINT) TO) SIGN) THING), and a right-hand member, which is a sense description of the meaning of the corresponding formula, such as (COMPASS AS INSTRUMENT POINTING NORTH). Each such pair (called a sense-pair) corresponds to one sense of the dictionary entry word. The sense description (right-hand member of pair) serves only to explain to the operator, in ordinary language print-out, which particular sense of the word is being operated on at any given stage of the procedure. The sense

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

The purpose of the formulae is to encode, and so distinguish the different senses of natural language words: one would expect to assign a different formula to each major sense of a word that a good dictionary distinguishes. Formulae consists of left and right parentheses and elements, where an element is one of the following 53 primitive semantic classifiers, or markers;

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.

These elements constitute the major categories of the classification of word-senses. The whole class of elements is not chosen at random; though as with any system of semantic markers it is difficult to justify its membership in detail on theoretical grounds (though see 4). I shall assume here only that one has to choose some set of markers to work with, and anyone's set of markers is always open to detailed objection. The markers are the basic elements in terms of which all the others in this system (templates, formulae etc.) are defined. So they cannot themselves be further defined, except by means of a table of 'scope notes' which gives the dictionary maker some indication of the marker elements. 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 types of bracket groups used by the dictionary maker in constructing formulae. 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 follows:

I. Adverbial Group.

((TRUE MUCH) HOW)--equivalent for "enough" used as an adverb; same function as "rather nicely" in English; can end with element HOW.

II. Adverbial Clause

(MAN FROM)--same function as "out of sight" in English; cannot end with any of the elements of D4 below, and hence a II type cannot be a well-formed formula (see below) by itself.

All these six types of sub-parts of formulae can themselves be interpreted (as can the formulae) so that each left-part is dependent on the corresponding right-part. This is a non-intuitive order in LISP but is an aid to reading the formulae for English speakers. This is best explained by means of an example. Thus, to take a sense-pair at random, say (COLOURLESS((((((WHERE SPREAD)(SENSE SIGN))NOT HAVE) KIND)(COLOURLESS AS NOT HAVING THE PROPERTY OF COLOUR)))). An explanation would be; "Colourless" 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

It is not difficult to see that that is what (in right-left order) the formula conveys.

Formulae are defined recursively as follows:

D.1. A formula is a binarily bracketted string of formulae and atoms.

D.2. An atom is an element, or an element immediately preceded by "NOT".

It follows from this that an element is not a formula. Not all formulae can be assigned to sense-pairs, but only well-formed formulae:

D.3. The head of a formula is its last atom. (and so is the opposite of the usual notion of 'head' in LISP 1.5).

D.4. A well-formed formula (wff) is (a) a formula, and (b) such that its head is one of the following elements:

HOW KIND FOLK GAIN MAN PART SIGN STUFF THING WHOLE WORLD BE CAUSE CHANGE DO FEEL HAVE PLEASE PAIR SENSE WANT USE THIS.

3. INITIAL FRAGMENTATION OF THE TEXTS.

An initial set of functions breaks each sentence of a paragraph up into strings of words, and, in certain circumstances, reforms discontinuous sub-strings 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 marker (such as a preposition).* So for example, the first paragraph of text is returned as on p.2 above 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 sub-strings, thus obliterating the original sentence boundaries.

---

* These markers are largely derived from Earl . (3)

It can be seen from the example paragraph above that the functions described do not simply segment sentences in a linear manner. They also 'take out' certain kinds of clause from within a sentence and append them as separate sub-strings. An example of this 'taking out' and reforming can be seen in the example paragraph reproduced above. The first two fragments read ((BRITAINS TRANSPORT SYSTEMS ARE CHANGING)(AND WITH IT THE TRAVELLING PUBLICS HABITS)).
These are produced from a sentence that originally read "Britains transport system and with it the travelling publics habits are changing". This sort of break-up 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 (see Halliday(4)) 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 re-ordering 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 character '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 re-ordering is that when a template has been matched with a fragment, the subsequent routines seek for 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 shall call "internal" those semantic routines which operate wholly within fragments, and "external" those which scan text outside a particular fragment in order to resolve its word-senses.

4. THE SYSTEM OF SEMANTIC ANALYSIS.

Production of single bare templates

The present system replaces each fragment of text by a number of strings of formulae (frames) constructed from the formulae 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 procedures schematically as follows:

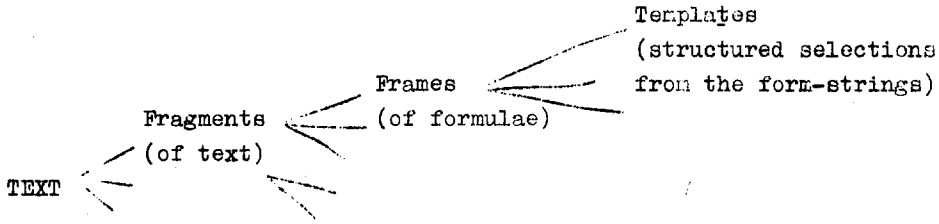

Fig. 2. Attachment of text to templates.

In the course of these procedures, therefore, 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 a particular set of word-senses for the words of the whole text.

The intuitive goal is that there should be just one string of templates in the set, and hence a unique ambiguity resolution of the text. However, the possibility of a number of independent resolutions cannot be excluded a priori.

Thus the outcome of applying these procedures to a text is either nothing, or a string of sense-explanations for the words of the text. In the case where the outcome is nothing, further procedures are defined whereby the system returns, as it were, to the beginning, adjusts one or more dictionary entries in a determinate way and then tries again to resolve the text. Thus the positive outcome described may be achieved after any one of a finite number of tries. As will be seen, there is a limit to the number of possible tries; and after it has been exhausted, the system has to conclude that the text cannot be resolved by this particular method.

The procedures of resolution can be put in the form of a set of phrase-structure rules which produce a nesting of frames of formulae from an initial paragraph symbol P. The rules are given in their generative rather than their analytic form, but I give the "lowest-level" rules first, because they are the ones applied at the first stage of analysis. The presentation will thus end up, rather than start, with highest level rules P→..., where P is a "paragraph symbol" analogous to the sentence marker, S, in conventional grammar.

Following what has been said above:

D.5. A frame for a fragment is a string of formulae such that each word of the fragment that has a (non-null) dictionary entry is represented by one and only one formula, and that formula has the same linear order in the as the

corresponding word in the fragment. Thus the set of all frames consistent with this definition (and with the dictionary entries for the words of some fragment) constitutes an initial representation of a fragment in the system.

We can now define the fundamental notion of template.

D.6. A bare template is any concatenated triple of elements that can be produced by Rules 1-6 below. (The rules 6. are only a sample).

R1. T → N1 + V + N2
R2. V → BE
R3. N2 → KIND, THIS, GRAIN, THING, SIGN.
R4. N1 → GRAIN, THIS, THING, PART, SIGN, MAN, FOLK, STUFF, WHOLE, WORLD.
R5i. (N1 → THIS) +...+ N2 → PART, MAN, FOLK, STUFF, WHOLE, WORLD
ii. (N1 → THING) +...+ N2 → PART, STUFF, WHOLE, WORLD
iii. (N1 → PART) +...+ N2 → PART, STUFF, WHOLE, WORLD
iv. (N1 → SIGN) +...+ N2 → PART, STUFF
v. (N1 → MAN) +...+ N2 → PART, FOLK, STUFF, MAN
vi. (N1 → FOLK) +...+ N2 → PART, MAN, FOLK, STUFF
vii. (N1 → STUFF)+...+ N2 → PART, STUFF, WHOLE, WORLD
viii. (N1 → WHOLE)+...+ N2 → PART, STUFF, WHOLE, WORLD
ix. (N1 → WORLD)+...+ N2 → PART, STUFF, WHOLE, WORLD
x. (N1 → GRAIN)+...+ N2 → PART
R6i. (N1 → GRAIN)+...+ V → PAIR, DO, CAUSE, CHANGE, HAVE
ii. (N1 → THIS) +...+ V → PAIR, DO, CAUSE, CHANGE, HAVE

The form of rules 5 and 6 is simply a convenient abbreviation of a more conventional form. For example:

R5 iv. (N1 → SIGN) +...+ N2 → PART STUFF

is simply an abbreviated expression of the two context-dependent phrase-structure rules:

SIGN+...+ N2 → SIGN+...+ PART, and
SIGN+...+ N2 → SIGN+...+ STUFF.

These rules produce bare templates in the form:
Substantive (or noun) type element +
Active (or verb) type element +
Substantive (or noun) type element.

Thus MAN+HAVE+PART can be produced in this way, but MAN+BE+WORLD cannot. This order we call the standard order, and templates are always considered and compared in this order even if located in fragments in other (nonstandard) orders, or in "debilitated forms."

D.7 & 8. If N1+VN2 represents the standard order, then
V+N1+V2 and N1+N2+V are nonstandard orders, and
N1+N2
N1+V
N1 N1
V are debilitated forms.

D.9. A fragment matches with templates if a frame for it contains concatenations of heads (in left-right order) corresponding to any template produced by Rules 1-11.

Where: (* indicates a blank item).

R7: THIS → *
R8: B2 → *
R9: KIND → *
R10: V → *

R11.i N1 +...+ (KIND → *) → KIND +...+ N1
ii. (V → *) +...+ KIND → KIND +...+V
iii. N1 +...+ (V → *) → V+...+N1
iv. (v → *) +...+ N2 → N2 +...V

Rules 1-6 produce standard forms of bare template, and Rules 7-11 produce (by means of deletions and reorderings) the permitted debilitated and nonstandard forms. The latter rules produce actual text-items, in the sense of heads (of formulae) to be located in the frames that represent fragments of text directly.

In order to produce templates that can plausibly be interpreted as meaning structures for fragments - in that they correspond to the heads and frames for the correct word-senses of the fragments - it is necessary that classes of templates be produced in a given order. There are four such ranks of classes, as shown by the following table:

RANK	TEXT-ITEMS	STANDARD FORM
I	N1+V	N1+V+THIS
	V+N1	THIS+V+N1
	N1+V+N2	N1+V+N2
	V+N1+N2	N1+V+N2
	N1+N2+V	N1+V+N2
	KIND+N1	N1+BE+KIND
	N1+V+KIND	N1+V+KIND
II	N1+KIND+V	N1+V+KIND
	V+N1+KIND	N1+V+KIND
	N1+KIND	N1+BE+KIND
	N1+N2	N1+BE+KIND
III	V+KIND	THIS+V+KIND
IV	V	THIS+V+THIS
	NI	THIS+BE+N1
	KIND	THIS+V+KIND

Fig 3. Preference table for bare templates.

Since Rules 1-11 are nonrecursive, there is no problem about ordering the productions in this way. Apart from the forms given in the table, there are only vacuous cases such as *+*+*.

The above table is intended to make clear the relation between the various standard forms (in the rightmost column) and the corresponding "items in frames" produced or recognized (middle column). Thus in the generative mode, text items are produced from the standard forms by transposition and deletion. In the analytic mode the text-items are recognized in the rank order shown, and then transposed and augmented with dummy BE and THIS elements so as to be in standard form for further computation.

The actual function of the rank choice is best explained by example, particularly as regards the composition of Rank I, since the ranks lower than I clearly consist of "debilitated forms" and it is intuitively plausible to produce fuller forms first. This ordering is one example of the general rule which enables template matching to do (at least) the work of a conventional grammar; namely, pack the frames as tightly as possible, or, in other words, produce the fullest possible template.

The presence in Rank I of the debilitated form KIND+N1 can be understood by considering, for example, the fragment:

(THE OLD TRANSPORT SYSTEM).

To simplify matters I shall consider only (i) the frame consisting of representations of the <u>appropriate</u> senses of the words in that fragment, and (ii) the frame identical with the first except that it contains representations of OLD as substative (noun = "the old people") and the active (verb)

form of TRANSPORT. Thus, by the semantic coding system described above, those two : will contain the following heads, and in the order shown:

i. ... ...KIND) ...KIND) ...GRAIN) , and
ii. ... ...FOLK) .....DO) ...GRAIN) .

Now the above rules generate both

(FOLK+DO+GRAIN) and (KIND+GRAIN)

as strings of text-items; the latter by deletion from (N1+BE+KIND) and (KIND+N1). It is clear that if the form KIND+N1 were not in Rank I with forms like (N1+V+N2) which yield (FOLK+DO+GRAIN), then a substantive phrase like this one would never receive a proper interpretation, since Rank I (without the form (KIND+N1)) would always look for an active (verb) sense for "transport" and having found one, would be satisfied.

As I have described the process so far both bare template forms (FOLK+DO+GRAIN) and (GRAIN+BE+KIND) would be produced. I shall show in the next section the additional procedures which produce the second of these in preference to the first Production of single full templates.

Further production rules limit the templates actually produced, and these require the notion of full template, defined as follows:

D.10. A full template is two triples of formulae such that the heads of the first triple constitute a bare template, and the second triple can be produced from the first by the rules 12-16.

D.11. The six formulae constituting a full template are called text-values.

The six formulae so defined give content to the corresponding bare template (expressed by the heads of three of the formulae). The rules 12-16 specify the <u>other</u> three formulae in such a way that each of them can be the qualifier of one of the formulae with a head defining part of the bare template. The rules 12-16 (not given here for reasons of space) are, in effect, rules producing an ordered pair of formulae such that the first is an appropriate qualifier for the second. Thus rule 13i produces an adjective type of formula (one ending in KIND) before a noun-type of formula, and so on.

The full templates are the items with which the system really operates. They can be illustrated by contrast with bare templates by considering fragment 3 of the paragraph examined earlier. That fragment was "It is the old permanent way". Among the bare templates produced for it by the system are the following two:

```
((IT IS THE OLD PERMANENT WAY)
  ((THING BE SIGN)
   (((THIS THING) (IT AS INANIMATE PRONOUN))
    ((BE BE) (IS AS HAS THE PROPERTY))
    ((((MAN FOR) ((WHERE POINT) FROM)) (LINE SIGN))
     (WAY AS PATH OR ROUTE))))
((THING BE SIGN)
 (((THIS THING) (IT AS INANIMATE PRONOUN))
  ((BE BE) (IS AS HAS THE PROPERTY))
  ((((THIS THING) (TRUE USE)) SIGN) (WAY AS MEANS))))
```

The fragment here is tied to two items, each of which is a bare template triple followed by the three formulae in the sense frame which locate it (their last elements are the same as those of the template triple A point of

interpretation should be added here for speakers of American English: all speakers of British English interpret "way" in this fragment as having its "path or route" sense in this context.

The two bare templates are now expanded to full templates as follows:

```
((IT IS THE OLD PERMANENT WAY)
  ((THING BE SIGN)
   (((ONE THING) (IT AS INANIMATE PRONOUN))
    ((BE BE) (IS AS HAS THE PROPERTY))
    ((((THIS THING) (TRUE USE)) SIGN) (WAY AS MEANS))
    NIL NIL ((NOTCHANGE KIND) (PERMANENT AS UNCHANGING))))
  ((THING BE SIGN)
   (((ONE THING) (IT AS INANIMATE PRONOUN))
    ((BE BE) (IS AS HAS THE PROPERTY))
    ((((WHERE IN) ((WHERE POINT) FROM)) (LINE SIGN))
     (WAY AS PATH OR ROUTE))
    NIL NIL ((NOTCHANGE KIND) (PERMANENT AS UNCHANGING)))))
```

These two items are the expansions (in frames of sense pairs) of the two bare templates. They consist of the same items as the bare template plus three formulae which are the qualifiers of the first three, (the fourth of the six is the qualifier of the first of the six and so on). In this the 'it' and 'is' have no qualifiers, hence the LISP 'NIL's in those positions. Bare templates other than these two were matched onto the fragment, but only these two could be expanded in this way. Hence these two were the 'survivors' and the others were rejected from further consideration.

When expanding in this way to produce full templates from bare ones the following meta-rule (M15) is applied

"Produce preferentially those full templates in which as many elements as possible are developed by the rules R12-R14." This means producing if possible those full templates in which each element of the bare template has a formally appropriate predecessor. By means of a further rule (R16) an attempt is made to produce not only full templates with formally appropriate internal relations, but also ones with semantically close internal relations as well. That is to say, full templates such that the triple of qualifying formulae are semantically close to the formulae they respectively precede. Where,

D.12. Two formulae are said to be semantically close if:

i) they share a common pair of elements; or
ii) 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
iii) 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.

Rules producing more than one template

I can now consider the production of concatenations of the full templates described so far.

D.13. A paragraph string is any string of templates produced by the rules 17 & 18 from the paragraph symbol P.

R17. $P \rightarrow T_r + T_s$

if T_r is a full template written as a string of six formulae thus,

$$[\ F^1_{r1} + F_{r1} + F^1_{r2} + F_{r2} + F^1_{r3} + F_{r3}\ ]$$

where F_{r1} is a noun type; F^1_{r1} its qualifier (adjective type); F_{r2} is a verb type, F^1_{r2} its qualifier (adverb type), and so on, then

R18. $$(T_s \rightarrow F^1_{s1} + F_{s1} + F^1_{s2} + F_{s2} + F^1_{s3} + F_{s3})\ T_s$$
$$\rightarrow (F^1_{t1} + F_{t1} + F^1_{t2} + F_{t2} + F^1_{t3} + F_{t3}) + \ldots +$$
$$(F^1_{u1} + F_{u1} + F^1_{u2} + F_{u2} + F^1_{u3} + F_{u3}),$$

where the values of the two template forms produced are semantically close.

D.14. Two full templates T_r T_s are <u>semantically close</u> if (with the above notation for full templates) at <u>least</u> two of the following pairs of formulae are (i) such that the head of the second is identical with, or in the negation class of, the first:

$(F_{r1}\ F_{s1})$, $(F_{r1}\ F_{s3})$, $(F_{r2}\ F_{s2})$, $(F_{r3}\ F_{s1})$, $F_{r3}\ F_{s3})$; and (ii) either they, or their qualifier formulae, are semantically close. These ten possible directions of connection between two full templates can be shown schematically as follows:

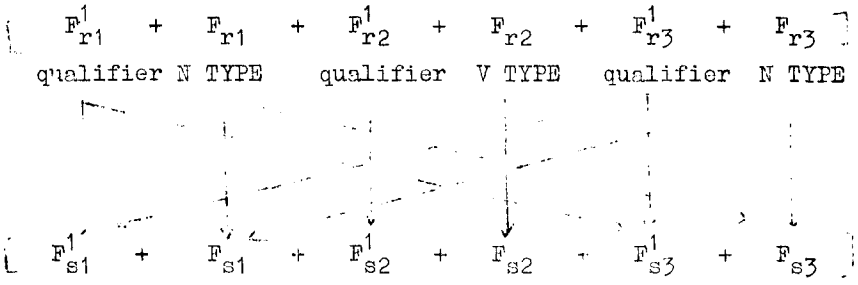

Fig 4. <u>Connecting pattern between full templates</u>.

See note on page 31.

Rule 18 does not, as might appear at first sight, involve self-contradiction. The shorthand form of rule writing is now being extended to mean that when T_s has been rewritten as $F_{s1}+\ldots\ldots+F_{s3}$, then the latter may be rewritten as the right-hand side of the second arrow.

This "expansion-concatenation" rule can be recursively applied to the initial productions from P. Thus at any stage in the process a paragraph string of full templates is produced. At any point the string can be considered terminal and, with the aid of the dictionary of words and sense-pairs, the paragraph string of templates can be converted to a string of frames and so to a text of words. This is analagous to the introduction of the lexicon in any standard phrase-structure grammar. The dictionary entries themselves can be put in phrase structure form. For example, if a word W_n has two sense pairs S1 and S2 in its dictionary entry, then the sense-pairs themselves can be put in the form S1 $\rightarrow W_n$ and S2 $\rightarrow W_n$ respectively. This form of the dictionary entries is useful in representing the self-modification of the system described below.

5. APPLICATION OF THE SYSTEM TO TEXTS.

Matching bare templates onto fragments.

Rules 1-6 above define the matching of bare templates onto a fragmented text, one bare template onto each text fragment. TEMPO is the main (top level) function that does this: it examines in turn all the frames of sense pairs for a fragment, and so on for all the fragments of a paragraph.

It takes as its argument a frame of sense-pairs, one for each word of a given fragment. TEMPO scans each such combination in turn, starting with the frame containing all the main senses of the words in the fragment (the first ones in the dictionary entry for each word). TEMPO searches for triplets of heads in the order of preference given in fig.3, above. For example, if it finds type I templates it doesn't look for any of types II-IV and so on. 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 N+N form, it replaces the N+N by N+BE+N (BE being the "dummy verb"). Similarly V+N and N+V are replaced by THIS+V+N and N+V+THIS respectively (THIS being the "dummy substantive"). The function of these dummy features is to supply a general form of template for subsequent processing, even when it is not wholly present in the text. Suppose, for example, a fragment consisted not of an assertion form, but of a noun phrase like "the black wizard" , where the heads of the appropriate codings for "black" and "wizard" would b, KIND and MAN respectively. As there is no verb, a debilitated template of the N+N form would match onto these two heads, and that would then be converted into MAN+BE+KIND. which is the intuitively correct interpretation (WIZARD is BLACK). The dummy verb is added in the way described; and in cases like this, where the first head is the predicate KIND, the order of the two heads is reversed, so as to give the MAN+BE+ KIND form. This transposition is defined by R11i.

The internal rejection functions (matching full templates)

Earlier I distinguished between internal and external procedures. Internal rejections are those procedures which cast out matching templates by means of the

expansion from bare to full templates. The main function which does this is PICKUP. It takes a fragment name as argument and constructs the TEMPO value for it. PICKUP makes a decision in the case of each template whether or not to reject it from further consideration. Those that survive are then considered further by the external rejection procedures. The survivors from PICKUP represent a stage of ambiguity resolution beyond that given by TEMPO. If, for example, PICKUP examines a template that has been matched onto a fragment containing the words round box, where a template head had been attached to a formula for box, then, hopefully, PICKUP keeps at least the template in which round is coded by its "spatial property sense" and box is coded by its "container" sense.

Inside PICKUP the function REFINE returns as its value a list of five sub-lists of full templates: its first sublist contains those form-close internally in four ways (as defined by rules 12-15), down to the last sub-list containing those with no such closeness. PICKUP takes the first non-empty sub-list of REFINE, and of that returns as its value the full templates that are semantically close as well (if any).

The 'semantic parser'; resolving a paragraph.

The top-level function PARSPARA takes as its argument a list of fragments, produces the PICKUP value for each (in the full template form given on p.14) and then parses these full templates using rules 17 & 18. A nesting of templates that satisfies these rules is an interpretation for the paragraph, and its word-sense content is read off and printed out (since a nesting of full templates is simply a selection of the possible word-sense assignments for a text.) Full templates which cannot be parsed with those for other

fragments are simply rejected .This is the external rejection procedure referred to earlier.

Functions called FIT and JAM express rule 18: 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. If the three main formulae in a full template are related to the three main formulae of another template by any three of the connectivities expressed in fig.4. above, then the two templates FIT (are semantically close). The function JAM builds up a representation of the two templates based on their connectivities. FIT and JAM work with message-pairs, which are to a fragment what a sense pair is to a word.

D.15. A message-pair is a two-item list: one item is a list of the first three sense-pairs of some full template, the other item is a list containing the name of some fragment with which the full template matches.

PARSPARA constructs the PICKUP value (full templates) for its list of fragments, and then builds up all possible frames of message-pairs for the paragraph. 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.

as if we have a paragraph frame containing the two message-pairs:

((BRITAINS 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 TRAVELLING PUBLICS HABITS)
((MUCH((MAN FOR)(MUCH DO))GRAIN))
(HABITS AS REPEATED ACTIVITIES))
((BE BE (DUMMY))
(((WHOLE FOLK) KIND)
(PUBLICS AS CONNECTED WITH THE WHOLE PEOPLE))
NIL (((WHERE CHANGE)HOW)
(TRAVELING AS MOVING FROM PLACE TO PLACE))))),

then the two full templates in those message-pairs are a fitting pair, and we shall expect them to be replaced in the string by the form:

(((BRITAINS TRANSPORT SYSTEM ARE CHANGING)
(WITH IT THE TRAVELLING 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)
TRAVELLING AS MOVING FROM PLACE TO PLACE))))).

This fitting together, or parsing, of message-pairs expresses the semantic compatibility between the corresponding fragments discussed earlier . PARSPARA rewrites such strings

of message-pairs recursively, trying to reach a two item list which (by rule 17) is P the paragraph symbol. If this point is reached the corresponding sense-resolution is read off and printed out for the paragraph in the following form: each fragment is given with the list of sense expressions 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).

```
(((BRITAINS TRANSPORT SYSTEM ARE CHANGING)
   ((WORDS RESOLVED IN FRAGMENT)
    ((TRANSPORT AS PERTAINING TO MOVING THINGS ABOUT)
     (BRITAINS 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 TRAVELLING PUBLICS HABITS)
   ((WORDS RESOLVED IN FRAGMENT)
    ((TRAVELLING AS MOVING FROM PLACE TO PLACE)
     (IT AS INANIMATE PRONOUN )
     (HABITS AS REPEATED ACTIVITIES)))
   ((WORDS NOT RESOLVED IN FRAGMENT) NIL)
```

fig. 5. First two fragments of the resolved output for a text paragraph.

The original English for the first two fragments of that paragraph was "Britain's transport system and with it the travelling public's habits are changing".

The sense constructor procedure.

A procedure was built in to the system to deal with the

cases where 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. An obvious suggestion for tackling 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 thought of as adding a new rule F - a, where F is a formula and a word name, and so expanding to a new rule system as the system adjusts to the particular text.

In practice PARSPARA examined 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. It seemed a good first guess at the recalcitrant word that it was in template that 'cohered' least with its neigbours. If all the frame 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 recalcitrant template in BESTPARS and suggested that a word in the corresponding fragment have its dictionary of sense pairs enlarged by identifying the recalcitrant word with the most 'semantically close' word in the paragraph. If the operator accepts the system's suggestion at the teletype, the system is rerun with the enlarged dictionary to try and get a resolution. In such a case (or if none of the system's suggestions are acceptable to the operator) the 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 the following are possible:

```
(CONSTRUCTER MODE)
((NO RESOLUTION ALL PATHS BLOCKED)
  (BEST PARSING CONTAINS)
  (((((KIND SIGN) (ATTRIBUTE AS A PARTICULAR KIND OF
        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)
      (WERE 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. 6. Dialogue in CONSTRUCTER MODE together with first part of subsequent resolution.

6 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 claim that the present system could constitute a tentative criterion for meaningfulness: a text is meaningful if and only if a system like the present one can resolve it. It is easy enough to get 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. I have argued at length elsewhere that it is possible also to justify the corresponding sufficient one (8). 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 (2), and the 'Logical Syntax' school generally, said could be shown to be meaningless on the basis of a system of analytic rules, though 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 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 (6) 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 don't 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 connexions can be made also between the semantic structure assigned to a text by the present system and that assigned by formal logic. These connexions have been investigated in the cases of the five philosophical paragraphs, which have a form sufficiently like the one required by formal logic. These connexions 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 recognised that an argument can be formally valid (and even have true premisses) 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 premiss, each premiss 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 (1) 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'.

REFERENCES

1. Bobrow, D.G. Natural Input for a Computer Problem-Solving System. Ph.D. Thesis, M.I.T. (1965)

2. Carnap, R. The Logical Syntax of Language, Routledge, London (1937)

3. Earl, L. An algorithm for Automatic Clause delimitation in English sentences. Lockheed Missiles and Space Co., Tech. Rept. 5.13.64. 5. (March, 1964)

4. Greenberg, J.H. (ed) Universals of Language. M.I.T. Press, Cambridge Mass. (1963)

5. Halliday, M. Some aspects of the thematic organization of the English Clause. RAND Memorandum 5224 (January, 1967).

6. Laing, R.D. The Divided Self. Tavistock Publications, London (1960)

7. Katz, J., and Postal, P. An integrated theory of Linguistic Descriptions. M.I.T. Press, Cambridge, Mass., (1964).

8. Wilks. Y. Argument and Proof in Metaphysics, from an Empirical Point of View. Ph.D. Thesis, Cambridge. (1968)

to page 19:
The negation class of elements for each element is derived inductively by a separate procedure. The notion onvolved is like that of logical contrary: an element and any member of its nggation class are partly synonymous and partly exclusive. For example, an entity can be basically a STUFF or basically a THING; it cannot be both so each of these elements is in the negation class of the other.