Information Retrieval and Language Processing
CA. Montgomery Editor

An Intelligent Analyzer and Understander of English

Yorick Wilks
Artificial Intelligence Project
Stanford University

The paper describes a working analysis and generation program for natural language, which handles paragraph length input. Its core is a system of preferential choice between deep semantic patterns, based on what we call "semantic density." The system is contrasted:
(1) with syntax oriented linguistic approaches, and
(2) with theorem proving approaches to the under standing problem.

Key Words and Phrases: artificial intelligence, computational linguistics, template, paraplate, stereo-• type, machine translation, understanding, natural language processing, semantic preference, semantic density
CR Categories: 3. 36, 3.42, 3.63

Introduction

After the unhappy conclusions of most early attempts at machine translation, some justification is required for presenting it again as a reasonable computational task. Minsky [4], among others, argued that there could be no machine translation without a system that, in an adequate sense, understood what it was trying to translate. The meaning structures and inference forms that constitute the present system are intended as an understanding system in the required sense, and as such, justify a new attack on an old but important problem.

Machine translation is an important practical task; furthermore, it has a certain theoretical significance for a model of language understanding. For it provides a clear test of the rightness or wrongness of a proposed system for representing meaning, since the output in a second language can be assessed by people unfamiliar with the internal formalism and methods employed. Few other settings for a theory of language analysis leave room for such objective tests. Dialog systems are . notoriously difficult to assess; and command systems are restricted to worlds in which commands are relevant, e.g. those of physical objects and the directions for picking them up, which domain excludes the world of real nonimperative discourse about such subjects as friendship, the United Nations, and the problems of juvenile delinquency. On the other hand, conventional systems of linguistics produce only complex representations that can be disputed only on internal grounds. They are never used to produce objective, discussable output, like a sentence in another language that would test the adequacy of the whole representation.

It should be added here that although the present system is cast in the role of a machine translation system, the popular forms of example to test "understanding"—i.e. finding the correct reference of a pronoun on the basis of knowledge of and inferences about the real world—can all be reconstructed within it, as will be shown.

Since the early machine translation work there has been a considerable development in formal linguistics, in particular, the creation of the school of transformational grammatical analysis. This form of analysis of natural language has little relation to the work described here, and for three reasons.

Firstly, Transformational Grammar was set up to be quite independent of all considerations of meaning, context, and inference, which constitutes something of a disqualification for the present task, namely understanding language. Consider such an even apparently structural-grammatical matter as the ambiguity of prepositions; "out of," for example, is highly ambiguous, which can be seen from any reflection on such sentences as: I live out of town. I hit her out of anger. I threw the ball out of the window. The statue is made out of marble. An objective measure of the ambiguity is that the occurrences of "out of" in those sentences would be translated into French in three different ways. Yet, even in such a basic structural area, Transformational Gram-mar makes no suggestions whatever as to how the choice should be made. Whereas in the Preference Semantics system, described below, the choice is made in a simple and natural manner. Such defects as this have been to some extent remedied in a recent development of the Transformational Grammar system, Generative

This work has been supported under Advanced Research Projects Agency Contract #457. Author's address: Institute per gli Sttudi Semantici e Cognitivi, 6976 Castagnola, Villa Heieneurn, Switzerland.

Semantics. However, for our purposes Generative Semantics, like Transformational Grammar, suffers from the other two defects below.

Secondly, it is a matter of practical experience, that Transformational Grammar systems have been extremely resistant to computational application. This practical difficulty is in part due to theoretical difficulties concerning the definition and computability of Transformational Grammar systems.

Thirdly, Transformational Grammar and Genera-tive Semantics systems suffer one overwhelming defect, from the point of view of understanding natural lan-guage. Both have a "derivational paradigm," which is to say, both envisage a system which constructs a derivation by running from an initial symbol to a language sentence. Such derivations have the function of either accepting a sentence or rejecting it because no such derivation can "reach" the sentence from the starting sym-bol. Thus all sentences are sorted into two groups by such systems— the acceptable and the unacceptable— and by doing this they claim to define the notion of an "acceptable," "meaningful," or "grammatical" sentencee.

One can see how far such a task is from the one of understanding language, for sorting in this way is exactly what human beings do not do when they hear a sentencc. They endeavor to interpret it, changing their rules if necessary as they do so. Yet, within the Transformational Grammar and Generative Semantics derivational paradigm, it makes no sense to talk of changing the rules and trying another set, even though that is .just what any "intelligent" understanding system must do. For example, most conventional grammatical sys-tems are armed with some rule equivalent to "only animate things perform tasks of a certain class," which compels them to reject such perfectly comprehensible utterances as those which speak of the wind opening doors and cars drinking gas. (It is unimportant here whether any particular system employs such a particular rule. The point here is a general one about behavior in the face of rule failure.) Only an "intelligent" system, outside the derivational paradigm and able to reconsider its own steps, can overcome this defect. The limitations of Transformational Grammar and Generative Semantics systems, from the point of view of this project,have been discussed in detail in [12 and 13].

The proper comparisons for the present work are with systems of analysis originating from within either artificial intelligence or computational linguistics, none of which (except the work of Woods [17]) owes any strong debt to the Transformational Grammar tradition all of which, in differing degrees, make the concept of meaning representation central, such as the work of Simmons [11], Winograd [16], Schank [8], and Sandewell [7]

Some points of difference between these systems and Preference Semantics may be mentioned briefly.

(i) Preference Semantics is very much oriented toward processing realistic text sentences of sense complexity and of up to 20 to 30 words long. This difference of emphasis, and the sentence fragmentation and large-scale conceptual linkages its implementation requires, distinguishes Preference Semantics from all the approaches mentioned.

(ii) Preference Semantics copes with the words of a normal vocabulary, and with many senses of them, rather than with single senses of simple object words and actions. It is not wholly clear that the methods of [16] could, even in principle, be extended in that way.

(iii) Preference Semantics contains no conventional grammar for analysis or generation: its task is performed by a strong semantics. This contrasts with Winograd's use of a linguistic grammar and simple marker system, and to some extent with Simmons' use of case grammar,

(iv) Preference Semantics does not take theorem proving techniques, of whichever major type, to be the core manipulations for an understanding system, but rather sees them as techniques to be brought in where appropriate. In this respect it differs most strongly from Sandewall, whose work assumes some form of theorem prover of a resolution type, into which his predicate calculus representations of natural language sentences can be plugged. Preference Semantics also differs here from Winograd, whose PLANNER-based system is far more oriented to the proving of truths than the Preference Semantics system described below. Another major difference between Preference Semantics and these two other systems is that Preference Semantics inference rules operate on higher level items, structures of semantic concepts and cases representing whole sentences and paragraphs of text, rather than on items at the level of text words and facts (or predicates and features that replace such items one to one in grammatically parsed structure). The latter approach leads to an enormous multiplication of axioms/inference rules, with all the subsequent difficulty of searching among them.

Nothing here, of course, denies the need for knowledge of the physical world, and inferences based upon it, for understanding and translation. What is being argued for here is nondeductive, common sense inference expressed in a formalism that is a natural extension of the meaning representation itself.

A simple case will establish the need for such inference: consider the sentence "The soldiers fired at the women, and we saw several of them fall." That sentence will be taken to mean that the women fell, so that when, in analyzing the sentence, the question arises of whether "them" refers to "soldiers" or "women" (a choice which will result in a differently gendered pronoun in French), we will have to be able to infer that things fired at often fall, or at least are much more likely to fall than things doing the firing. Hence there must be access to inferential information here, above and beyond the meanings of the constituent words, from which we could infer that hurt beings tend to fall down.

The deductive approaches mentioned claim to tackle just such examples, of course, but in this paper we will argue for a different approach to them, which we shall call common sense inference rules. These are expressions of "partial information" (in McCarthy's phrase): generalizations, like the one above about hurt things tending to fall down, which (a) are not invariably true and (b) tend to be of a very high degree of generality indeed. It is part of the case being made here that the importance of such apparently obvious truths in natural language understanding is considerable, but also easy to overlook.

A System of Semantics Based Language Analysis

A fragmented text is to be represented by an interlingual structure, called a *Semantic Block,* which consists of *templates* bound together by *paraplates* and *common sense inferences.* These three items consist *of formulas* (and predicates and functions ranging over them and subformulas), which in turn consist *of elements.*

Some of these semantic items represent text items in a fairly straightforward way as follows:

Items in semantic representation	*Corresponding text items*
formula	English word sense
template	English clause or simple sentence
semantic block	English paragraph or text

Paraplates and common sense inferences, as we shall see, serve to bind templates together in the semantic block. Semantic elements correspond to nothing in a text, but are the primitives out of which *all* the above complex items are made up.

Semantic **Elements**

Elements are 70 primitive semantic units used to express the semantic entities, states, qualities, and actions , about which humans speak and write. The elements fall into five classes, which can be illustrated as follows (elements in uppercase, and the approximate concept expressed in lowercase):

(a) Entities: MAN (human being), STUFF (substances), SIGN (verbal and written symbols), THING (physical ob ject), PART (parts of things), FOLK (human groups), ACT (acts), STATE (states of existence), BEAST (animals), etc.
(b) Actions: FORCE (compels), CAUSE (causes to happen), FLOW (moving as liquids do), PICK (choosing), BE (exists), etc.
(c) Type indicators: KIND (being a quality), HOW (being a type of action), etc.
(d) Sorts: CONT (being a container), GOOD (being morally acceptable), THRU (being an aperture), etc.
(e) Cases: TO (direction), SOUR (source), GOAL (goal or end), LOCA (location), SUBJ (actor or agent), OBJE (pa tient of action), IN (containment), POSS (possessed by), etc.

In addition to these primitive elements, there are *class* elements whose names begin with an asterisk, such as *ANI for the class of animate elements MAN, BEAST. and FOLK; *HUM for human elements MAN and FOLK; *PHYSOB, which denotes the class of elements containing MAN, THING, etc., but not, of course, STUFF. There are also action class elements such as *DO.

The elements are not to be thought of as denotative, even of intensional entities, but as the elements of a micro-language in which more complex concepts are expressed. Thus their justification is wholly in terms of their use to construct semantic *formulas.*

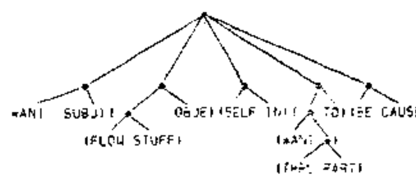
Semantic Formulas

Formulas are constructed from elements and right and left brackets. They express the senses of English words; one formula to each sense. The formulas are binarily bracketed lists of whatever depth is necessary to express the word sense. Their most important element is always their rightmost, "which is called the *head* of the formula, and it expresses the most general category under which the word sense in question falls. However, an element that is used as a head can function within formulas as well. So, for example, CAUSE is the head of the formula for the action sense of "drink" and it may be thought of as a "causing action," but CAUSE can also occur within the formula for a word sense, as it does, for example, within the formula for the action sense of "box," which can be paraphrased in English as "striking a human with the goal *of causing* him pain."

It will help in understanding the formulas to realize that there are conventional two-element subformulas, such as (FLOW STUFF) for liquidity, to avoid the introduction of new primitives. Another such is (THRU PART) to indicate an aperture. Formulas can be thought of, and written out, as binary trees of semantic primitives. In that form they are not unlike the lexical decomposition trees of Lakoff and McCawley. Here is a selection of formulas that will be needed in later examples. In each case I give the formulas as a tree of subformulas, with the head as the rightmost element, then as a. table of subformulas, and lastly as a paraphrase in English. The formulas are for the English words "drink" (as an action), "grasp" (as a *physical* action), "fire at." I also give, in a less extended range of forms, the formulas for "policeman," "big," "interrogates," "crook" as a human being and as a physical object, and "singing" as an activity.

Nothing at all depends on these particular codings. What is at issue here is the claim that codings of this degree of complexity, and containing at least this much semantic information, are necessary for doing any interesting degree of linguistic analysis.

"drink" (action) ->((*ANI SUBJ) (((FLOW STUFF) OBJE) ((SELF IN)(((*ANI (THRU PART)) TO) (BE CAUSE)))))



. Formulas are best seen as meanings of *subformulas*,each of which is either a case specification or a direct .specification on the head itself. Within any subformula there is a dependence at every level of the left half of a binary pair of the right half. This dependence relation is normally to be understood as type subspecification, .in .the way that *ANI specifies the type of agent in the example above. The mutual relation of the subformulas is not one of dependence, even though all the other sub-formulas be thought of as dependent on the rightmost subformula containing the head. However, the order of the subformulas is significant, since, for example, an ob-ject specification is considered to be the object of *all* actions to its right in the formula, whether they are the head or at some other level in the formula.

Susubformula	*Case/Act*	*Value*	*Explanation*
.*ASI SL BU	SUBJ	'*ANI	the preferred agent is animate
FLOW STUFF OBJE	OBJE	(FLOW STUFF)	preferred objeci is liquid
SELF IN.	IN	SELF	the container is the self, the subject
*ANI THRU PART))TO)	TO	(* ANI(THRU PART))	the direction of the action is a human aperture(the mouth)
*CASE	CAUSE	BE	the action is of causing to be (somewhere else)

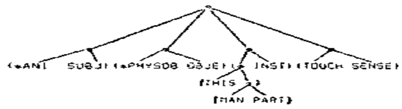
Let us now decompose the formula for "drink." It is to be read as an action, preferably done by animate things (*ANI SUB); to liquids, or to substances that flow FLOW STUFF)OBJE); causing the liquid to be in the ani-mate thing ('SELF IN); and via (TO indicating the direction case) a particular aperture of the animate thing, the mouth, of course. It is hard to indicate a notion as specific as "mouth" with such general concepts. But we think that it would be simply irresponsible to suggest adding MOUTH as a semantic primitive, as do semantic systems that simply add an awkward lexeme as a new "primitive."

This notion of "preferring" is important: SUBJ case displays the preferred agents of actions, and OBJE case the preferred objects, or patients. We cannot enter such preferences as stipulations, as many linguistic systems do, such as Fodor and Katz's "selection restrictions." For can be said to drink gall and wormwood, and cars are said to drink gasoline. It is proper to prefer the nor-mal quite different from probabilistically expecting it, we shall argue), but it would be absurd, in an intelligent understanding system, not to accept the abnormal if it is described. Not only everyday metaphor but the descrip-tion of the simplest fiction require it.

A formula expresses the meaning of the word senses to which it is attached. This claim assumes a common sense distinction between explaining the meaning of a word and knowing facts about the thing the word indi-cates. The formulas are intended only to express the former. to express what we might find in a reasonable dictionary, though in a formal manner.

Now let us consider:

"grasp" (physical action) -> ((*ANI SUBJ)((*PHYSOB OBJE) (((THIS(MAN PART))INST)(TOUCH SENSE)))))

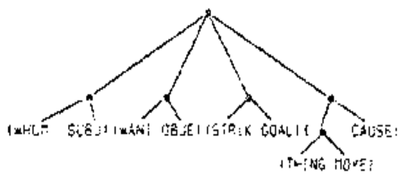


Subformula	*Case/Act*	*Value*	*Explanation*
(*ANI SUBJ)	SUBJ	*ANI	the preferred agent is animate
(*PHYSOBOBJE!	OBJE	.PHYSOB	the preferred agent is a physical object ((THIS(MAN
PART))INST)	INST	(!THIS(MAN PART))	ihe instrument is a human part, the hand
(TOUCH SENSE)	SENSE	TOUCH	the action is of physical contact

So, grasping in this sense is something preferably done by an animate thing to a physical object, done with the hand as instrument: an action of physical contact with the object. The mental sense of "grasp" is a THINK, ac-tion.

Now consider:

"fire at" (action) ->((*HUM SUBJ)((.*AN1 OBJE) ((STRIK GOAL)((THING MOVE)CAUSE))))



Sub/ormula	*Case/Act*	*Value*	*Explanation*
(*HUM SUBJi .	SUB]	.HUM	preferably done by a human
(*ANI OBJE;.	OBJE	*ANI	preferably done to an animate thing :!v™
<;STR'.K GOAL;	GOAL	STRIK	the aim being to strike the the animals; thing ((THiNG MOVE;CAUSEi
CAUSE	(THING MOVE)		the action is of causing an object (the bullet) to move

The fact that the bullet is the agent of the moving is im-plicit, and agents are unmarked except at the top level of the formula, although objects are marked at every level. So then, "firing at" is causing a thing to move so as to strike an animate target.

Let me now give the remaining formulas, with only an explanation, if the principles of the tree and table representation are now clear.

"policeman" → ((FOLK SOUR)((((NOTGOOD MAN)OBJE)PICK.)(SUBJ MAN)))

i.e. a person who selects bad persons out of the body of people (FOLK). The case marker SUBJ is the dependent in the last element pair, indicating that the normal "top first" order for subject-entities in formulas has been violated, and necessarily so if the head is also to be the last element in linear order.

"big" → ((*PHYSOB POSS)(MUCH KIND))

i.e. a property preferably possessed by physical objects (substances are not big).

"interrogates" → ((MAN SUBJ)((MAN OBJE)(TELL FORCE)))

i.e. forcing to tell something, done preferably by humans. to humans.

"crook→ ((((NOTGOOD ACT)OBJE)DO)(SUBJ MAN))

i.e. a man who does bad acts.

"crook" →-» ((((((THIS BEAST)OBJE)FORCE)(SUBJ MAN))POSS)(LINE THING))

i.e. a straight object possessed by a man who controls a particular kind of animal.

"singing" → ((*ANI SUBJ)((SIGN OBJE)((MAN SENSE)CAUSE)))))

which is to say, an act by an animate agent of causing a person to experience a sign, the song.

Semantic Templates

Just as the semantic elements have been explained by seeing how they functioned within formulas, so formulas, one level higher, are to be explained by describing how they function within *templates,* the third kind of semantic item in the system. The notion of a template is intended to correspond to an intuitive one of message: one not reducible merely to unstructured associations of word-senses.

A template consists of a network of whole formulas, and its connectivity is between an agent- , action- , and object-formula, such that from any one of these members of the basic triple a list of other formulas may depend. In any particular example, one or more of the formulas may be replaced by a dummy. We shall discuss such cases further.

The program sees each clause, phrase, or primitive sentence of text (called its *fragments}* as strings of formulas, drawn, one for each text word, from a dictionary. The program attempts to locate one or more templates in each string of formulas by first looking only at their head elements and seeking for acceptable sequences of heads.

A *bare template* is such an acceptable, or intuitively interpretable, sequence of an agent head, an action head, and an object head (subject again to the proviso about dummies). If there is a sequence of formulas whose heads are identical to such a bare template of elements, then the sequence of *formulas* is a template for that fragment, taken together with any other formulas that may be found to depend on those three main formulas.

For example: "Small men sometimes father big sons," when represented by a string of formulas, will contain the two sequences of head elements (where the heads of formulas are written under the corresponding word):

	small	men	sometimes	father	big	sons
	KIND	MAN	HOW	MAN	KIND	MAN
and	KIND	MAN	HOW	CAUSE	KIND	MAN

(CAUSE is the head of the verbal sense of "father"; "to father" is analyzed as "to cause to have life.")

The first sequence has no underlying bare template because there is no intuitively interpretable element triple there, in the sense in which MAN CAUSE MAN in the second sequence is intuitively interpretable as "a human causes another human to exist." Thus we have already disambiguated "father," at the same time as picking up a sequence of three formulas, which is the core of the template for the sentence. It must be emphasized here that the template is the sequence of formulas (which are trees or structured lists) and is be confused with the bare template, or triple of elements (heads) used to locate it.

It is a hypothesis of this work that we can build up a finite but useful inventory of bare templates adequate for the analysis of ordinary language: a list that can be interpreted as the messages that people want to convey at some fairly high level of generality (for template matching is not in any sense phrase-matching at the surface level). The bare templates are an attempt to explicate a notion of a nonatomistic linguistic pattern: to be located whole in sentences in the way that human beings appear to when they read or listen.

We would not wish to defend, item by item, the particular template list in use at any given moment. Such' lists are always subject to modification by experience, as are the formulas and even the inventory of basic elements. The only defense is that the system using them actually works; and if anyone replies that its working depends on mere inductive generalization, we can only remind them of Garvin's obvious but invaluable remark that all linguistic generalizations are, and must be, inductive.

Let us now illustrate the central processes of expansion and preference, in which the formulas become active items guiding the extension of the template network from a triple of formulas to a full template with preference bonds and dependent formulas. Let us consider the sentence "The big policeman interrogated the crook," for which we already have the appropriate formulas set out above.

The template matching algorithm will see this sentence as a string of formulas, one for each of its words, and will look only at the heads of the formulas. ~ I shall now write [crook(man)] to denote not the English words in the square brackets but *the formula for the word or words.* Then, since MAN FORCE MAN is in the inventory of bare templates, one scan of the string of formulas containing [crook(man)] will pick up the sequence of formulas [policeman][interrogated][crook-(man)], in that order. Again, when a string containing the formula [crook(thing)], the shepherd's sense of "crook," is scanned, since MAN FORCE THING is also a proper bare template, the sequence of formulas [policeman] [interrogated][crook (thing)] will also be selected as a possible initial structure for the sentence. I should add here that the formula for both *tenses* of "interrogates" is the same, the tense difference being indicated by a tense element put into the formula during the process of expansion now being described.

We now have two possible template representations for the sentence after the initial match; both are triples of formulas in actor-action-object form. Next, the templates are expanded, if possible This process consists of extending the simple networks we have so far, both by attaching other formulas into the network and by strengthening the bonds between those already in the template, if possible. Qualifier formulas can be attached where appropriate, and so the formula [big] is tied to that for "policeman" in both templates. But now comes a crucial difference between the two representations: one which will resolve the sense of "crook."

The expansion algorithm looks into the subparts of •he formulas that express preferences to see if any of the preferences are satisfied: as we saw, the formula [big] prefers to qualify physical objects. A policeman is such, and that additional dependency is marked in both templates: similarly for the preference of "interrogate" for human actors in both representations. The difference comes with preferred objects: only the formula [crook-man)] for human crooks can satisfy that preference for human objects, since the formula [crook(thing)] for shepherd's crooks, cannot. Hence the former template network is denser by one dependency, and is preferred over the latter in all subsequent processing: its connectivity is (ignoring the "the's"): [big] → [policeman] ↔ interogates] → [crook(man)] and so that becomes the template for this sentence. The other possible template (one arrow for each preferential dependency established, and a double arrow to mark the standard, nonpreferential, link between the three major formulas of the template) was connected as follows: [big] → poliiceman] ↔↔ [interrogates] ↔ [crook(thing)] and . is now discarded.

Thus the parts of the formulas that express preferences of various sorts are not only used to express the meaning of the corresponding word sense, but they can also. be interpreted as implicit procedures for the construction of correct templates. This preference for the greatest semantic density works well, and can be seen as an expression of what Joos calls "semantic axiom number one" [2], that the right meaning is the least meaning, or what Scriven [10] has called "the trick [in meaning analysis] of creating redundancies in the in-put." As we shall see, this uniform principle works over both the areas that are conventionally distinguished in linguistics as syntax and semantics. There is no such distinction in this system, since all manipulations are of formulas and templates, and these are all constructed out of elements of a single type.

Templates and Linguistic Syntax

As a further example of linguistic syntax done by preference, let us take the sentence "John gave Mary the book," onto which the matching routine will have matched two bare templates, since it has no reason so far to prefer one to the other, as follows:

John gave Mary the book
MAN GIVE THING
MAN GIVE MAN

The expansion routine now seeks for dependencies between formulas, in addition to those between the three formulas constituting the template itself. In the case of the first bare template, a GIVE action can be expanded by any substantive formula to its immediate right which is not already part of the bare template (which is to say that indirect object formulas can depend on the corresponding action formula). Again "book" is qualified by an article, which fact is not noticed by the second bare template. So then, by expanding the first bare template we have established in the following dependencies at the surface level, where the dependency arrows "→" correspond to *preferential* relations established between formulas for the words linked.

John ↔ gave ↔book
↑ ↑
Mary the

But if we try to expand the second bare template by the same method, we find we cannot, because the formula for "Mary" cannot be made dependent on the one for "give," since in that template "Mary" has already been seen, wrongly of course, as a direct object of giving, hence it cannot be an indirect object as well. So then, the template MAN GIVE MAN cannot be expanded to yield any dependency arcs connecting formulas to the template; whereas the template MAN GIVE THING two dependency arcs on expansion, and so gives the preferred representation.

This general method can yield virtually all the results of a conventional grammar covering the same range of expressions, while using only relations between semantic elements.

Case Ambiguity

In the actual implementation of the system, an input text is initially fragmented, and templates are matched with each fragment of the text. As we shall see, there are then complex routines for establishing contextual ties between these templates separated by fragmentation. However, it is claimed here that, for dealing with text containing realistically long and complicated sentences, some such initial fragmentation is both psychologically and computationally important.

The input routine fragments paragraphs at the occurrence of any of an extensive list of *key* words. The list contains all punctuation marks, subjunctions, conjunctions, and prepositions. In difficult cases, described in detail in [14], fragmentations are made even though a key word is not present, as at the slash in "John knows / Mary loves him," while in other cases a fragmentation is not made in the presence of a key word, such as "that" in "John loves that woman."

Let us consider the sentence "John is / in the country," fragmented as shown. It should be clear that the standard agent-act-object form of template cannot be matched onto the fragment "John is." In such a case, a degenerate template MAN BE DTHIS is matched onto the two items of this sentence; the last item DTHIS bein g a dummy object, indicated by the D.

With the second fragment "in the country," a dummy subject DTHIS fills out the form to give a degenerate template DTHIS PBE POINT. The PBE is the same as the head of the formula for "in," since formulas for prepositions are assimilated to those for actions and have the head PDO or PBE. The fact that they originate in a preposition is indicated by the P, so distinguishing them from straightforward action formulas with heads DO and BE. POINT (indicates a spatial location that is not a movable physical object) is the head of the formula for "country," so this bare template triple for the fragment only tells us that "something is at a point in space." At a later stage, after the preliminary assignment of template structures to individual fragments, *TIE* routines attach the structures for separated fragments back together. In that process the dummies are tied back to their antecedents. So, in "John is in the country," the DTHIS in the MAN BE DTHIS template for the first fragment of the sentence ties to the whole template for the second fragment, expressing where John is.

It is very important to note that a preference is between alternatives. If the only structure derivable does *not* satisfy a declared preference, then it is accepted anyway. Only in that way can we deal naturally with metaphor.

So, in examples like "I heard an earthquake singing / in the shower" (fragmentation as indicated by slashes), as contrasted with "'I heard / an earthquake sing / in the shower," we shall expect, in the first case, to derive the correct representation because of the preference of notions like singing for animate agents. This is done by a Simple extension of the density techniques to relations between structures for different fragments by considering, in this case, alternative connectivities for dummy parts of templates.

Thus, there will be a dummy subject and object template for /singing/, DTHIS CAUSE DTHIS, based on the formula for "singing" given earlier.

Now the overall density will be greater when the agent DTHIS, in the template for "singing," is tied to a formula for "I" in a preceding template, than when it is tied to one for "earthquake," since only the former satisfies the preference for an animate agent, and so the correct interpretation of the whole utterance is made.

But, and here we come to the point of this example, in the second sentence, with "sing" no such exercise of preference is possible, and the system must accept an interpretation in which the earthquake sings, since only that can be meant.

In order to give a rough outline of the system, I have centered our description on the stages of analysis within the individual fragment. After what has been described so far, *TIE* routines are applied to the expanded templates in a context of templates for other fragments of the same sentence or paragraph. The same techniques of dependency and preference are applied between full templates for different fragments of a sentence or paragraph. At that stage, (1) case ties are established between templates (using the same cases as occur within formulas at a lower level); (2) dummies are attached to what they stand for as we indicated with the earthquake example; (3) remaining ambiguities are resolved; and (4) anaphoric ties are settled.

Paraplates and Case Ambiguity

The first of these tasks is done by applying *paraplates* to the template codings, using the same density techniques one level further up, as it were. Paraplates are complex items having the general form:

(list of predicates on mark-template)(case)
(list of predicates on case-template)
(generation stereotype)

A stereotype is a context sensitive generation pattern which will be described in the section on generation below, and in what follows here I shall give the paraplates *without* the attached stereotypes. The paraplates are essentially patterns that span two templates, which I call the mark and case templates, where the mark template generally precedes, though not necessarily immediately, the case template. If the predicates are all satisfied by the contents of the two templates, then that paraplate is considered to match onto the two templates and the case ambiguity of the preposition that functions as the pseudo-action in the second template is solved. Thus if we were analyzing "He ran the mile in four minutes" and we considered the template for the second fragment "in four minutes," we would find that all the predicates in some paraplate for TIMEILOCATION case matched onto the appropriate parts of the templates for the two template fragments, and we would then know that the case of the second template was indeed TIMELOCATION and not, say, CONTAINMENT, as it would be in "He ran the mile in a plastic bag."

The paraplates are attached, as left-right ordered lists, to key words in English, generally prepositions and subjunctions. Consider the following three schematic paraplates for senses of "in" written out in order of preference below. These are presented without generation stereotypes for ease of explanation, but with a description in lowercase of which sense of "in" is in question in each line. The notion of *mark* is the standard intuitive one of the point of dependence of a phrase or clause. Thus, in "He ran the mile in four minutes" the second clause may be said to depend on the action "ran," which is then its mark. Whereas, in "He liked the old man in the corner," the mark of the second fragment is "man."

I will write the three paraplates out, first in linear order as they really are, and then in tabular form for ease of comprehension. The linear order is to be understood as corresponding to that of the six major formulas of the mark and case templates. The predicates in the paraplates may refer to any or all of these, The paraplates are called in on encountering the ambiguous subjunction, or most usually, ambiguous preposition that always functions as the pseudo-action of the second template—the one in hand, as it were. I have put a slash in the paraplate to indicate where the shift is

plates are called in on encountering the ambiguous subjunction, or most usually ambiguous preposition thar always functions as the pseudo-action of the second template – the one in hand, as it were. I have put a dash in the paraplate to indicate where the shift is, from predicates on the mark template to those on the case template. Also, where predicates have atomic arguments, like 2OCAS below, it indicates that those elements are separate arguments of the predicate Where a predicate, like PRMARK below, has an argu-ment that is a list, that list is a subformula that has to be located whole in the appropriate template formula so as to satisfy the predicate in question.

1. PR MARK (MOVE CAUSE)) (2OBCAS INST GOAL)/
   (TO into)(PROBJE(CONT THING)) 2 PRMARK
2. *DO)(2OBHEAD)/(LOCA make part)
3 PRMARK(MOVE CAUSE))/
   (TO into) ((PROBJE(CONT THING))

What is not made absolutely clear by that form of the paraplates is where, on the six formulas of the two templates, each of the above predicates matches. Let .us now set out each paraplate vertically in six lines corresponding in turn to agent of first template, action of first template, object of first template, and then the same order for the second, case template.

	FIRST AGENT
PRMARK.(MOVE CAUSE))	FIRST ACTION
2OBCAS INST GOAL)	FIRST OBJECT
	SECOND AGENT
(TO into)	SECONDACTION
PROBJE (CONT THING))	SECOND OBJECT

	FIRST AGENT
PRMARK *DO)	FIRST ACTION
2OBHEAD)	FIRST OBJECT
	SECOND ACTION
LOCA makejpart)	SECOND AGENT
	SECOND OBEJCT

	FIRST AGENT
PRMARK.(MOVE CAUSE))	FIRST ACTION
	FIRST OBJECT
	SECOND AGENT
TO into)	SECOND ACTION
PROBJE (CONT THING))	SECOND OBJECT

•DO is a wide class of action heads, TO and LOCA an case: markers, 2OBCAS and 2OBHEAD are simply predi-cates that look at both the object (third) formulas of of the current template (the second) and of the preceding template, i.e. at two objects. 2OBHEAD is true iff the two have the same head, and 2IOBCAS is true iff they contain the same GOAL or INSTRUMENT subformula. The fact that those two predicates actually apply at two of the six places is a notational weakness in the tabular display above. PRMARK is a predicate on the semantic form of the mark, or a word governing the fragment that the key begins. In all the following eamples, the mark is the action in the first fragment, and the predicate is satisfied iff it is a (MOVE CAUSE) action: an action that causes something to move. Similarly, PROBJE is a predicate on the semantic form of the object (third formula) of the current template, and is satisfied if the predicate's ,argument is found in the formula.

: Now consider the sentence "I put the key / in the lock," fragmented at the slash as shown. Let us consider that two templates have been set up for the second fragment: one for "lock" as a fastener, and one for the raising lock on a canal. Both formulas may be expected to refer to the containment case, and so to satisfy(PROBJ E CONT). We apply the first paraplate and find that it fits only for the template with the correct (fastener) sense of "lock," since only there will 2OBCAS be satisfied, i.e. where the formulas for "lock" and "key" both have a subformula under GOAL indicating that their purpose is to close something. The third paraplate will fit with the template for the canal sense of "lock," but the first is a more extensive fit (indicated by the order of the paraplates, since the higher up the paraplate list, the more nontrivial template functions a paraplate contains) and is preferred. This preference has simultaneously selected both the right template for the second fragment and the correct paraplate linking the two templates for further generation tasks.

If we now take the sentence "He put the number in the table," with two different templates for the second fragment (corresponding to the list and flat object senses of "table" respectively) we shall find that the intuitively correct template (the list sense) fails the first paraplate but fits the second, thus giving us the "make part of" sense of "in," and the right (list) sense of table," since formulas for "number" and (list) table" have the same head SIGN, though the formula for (flat, wooden) "table" does not.

Conversely, in the case of "He put the fork / in the table," fitting the correct template with the third paraplate will yield "into" sense of "in" (case DIRECTION) and the physical object sense of "table"; and this will be the preferred reading. Here we see the fitting of paraplates, and by choosing the densest preferential fit, which is always selecting the highest paraplate on the list that fits, we determine both word sense ambiguity and the case ambiguity of prepositions at once. Paraplate fitting makes use of deeper nested parts (essen-tially the case relations other than SUBJ and OBJE) of the formulas than does the template matching.

Anaphora and References

The TIE routines also deal with simple cases of anaphora on a simple preference basis. In cases such as “I bought the wine, /sat on a rock / and drank it," it is easy to see that the last word should be tied by T*IE* to “wine” and not "rock." This matter is settled by density after considering alternative ties for "it," and seeing which yields the denser representation overall. It will be "wine" in this case since "drink" prefers a liquid obiect.

In more complex cases of anaphora that require access to more information than is contained in formulas, templates, or paraplates, the system brings down what we referred to earlier as common sense inference rules-[1] Cases that require them will be ones like the sentence: "The soldiers fired at the women and we saw several of them fall." Simple semantic density considerations in *TIE* are inadequate here because both soldiers and women can fall equally easily, yet making the choice correctly is vital for a task like translation became the two alternatives lead to differently gendered pronouns in French. In such cases the Preference Semantics system applies a common sense rule, whose form, using variables and subformulas, would be

(1 (THIS STRIK) (*ANI 2)) <-» ((.*ANI 2)(NOTUP BE)DTHIS)

where the variables are restricted as shown, and the final DTHIS is simply a dummy to fill out the canonical form. This rule can be made more perspicuous by extending the informal [ ] notation to denote the *template form* representation of whatever is in the square brackets, thus: [1 strikes animate2]«-» [animate2 falls]. The rules are applied to "extractions" from the situations to form chains of templates and template forms, and a rule only ultimately applies if it can function in the shortest, most-preferred, chain.

The way the .common sense inferences work is roughly as follows: they are called in at present only when *TIE* is unable to resolve outstanding anaphoras, as in the present example. A process of extraction is then done, and it is to these *extractions,* and the relevant templates, that the common sense rules subsequently apply. The extractions are new template forms inferred from the deep case structure of formulas. So for example, if we were extracting from the template for "John drank the water," then going down into the tree structure of primitive elements in the formula for "drink." given earlier, we would extract that some liquid was now inside an animate thing (from the containment case in the formula for "drink"), and that it went in through an aperture of the animate thing (from the directional case)." Moreover, since the extractions are partially confirmed, as it were, by the information about actor and object in the surrounding template, we can, by simple tying of variables, extract new template forms equivalent to, in ordinary language, "the water is in John," etc. These are (when in coded form) the extractions to which the common sense rules apply as the analytical procedure endeavors to build up a chain of extractions and inferences. The preferred chain will, unsurprisingly, be the shortest.

So then in the "'women and soldiers" example we extract a coded form, by variable tying in the templates, equivalent to [soldiers strike women], since we can tell from the formula for "fired at" that it is intended to strike the object of the action. We are seeking for partial confirmation of the assertion [X? fall], and such a chain is completed by the rule given, though not by a rule equivalent to, say [something strike X] → [X die], since there is nothing in the sentence as given to partially confirm that particular rule in a chain, and cause it to fit here. Since we are in fact dealing with subformulas in the statement of the rules, rather than with words, "fitting" means an "adequate match of subformulas."

It is conceivable that there would be another, implausible chain of rules and extractions giving the other result, namely that the soldiers fall: [soldiers fire] A [X fires] → [X fired at] → [X fall], etc. But such a chain would be longer than the one already constructed and would not be preferred.

The most important aspect ,of this procedure is that it gives a rationale for selecting a preferred interpretation rather than simply rejecting one in favor of another, as other systems do. It can never be right to reject another interpretation irrevocably in cases of this sort, since it may turn out later to be correct, as if the "women" sentence above had been followed by "and after ten minutes hardly a soldier was left standing." After inputting that sentence the relevant preferences in the example might be expected to change. Nonetheless, the present approach is not in any way probabilistic. In the case of someone who utters the "soldiers and women" example sentence, what is to be taken as his meaning is that the women fell. It is of no importance in that decision if it later turns out that he intended to say that the soldiers fell. What was meant by that sentence is a clear, and not merely a likelihood, matter.

It must be emphasized that, in the course of this application, the *common sense* rules are not being interpreted at any point as rules of inference making truth claims about the physical world. It is for this reason that we are not contradicting ourselves in this paper by describing the Preference Semantics approach • while arguing implicitly against deductive and theorem proving approaches to language understanding. The clearest way to mark the difference is to see that there is no inconsistency involved in retaining the rule expressed informally as [1 strikes animate2] → [animate2 falls], and at the same time, retaining a description of some situation in which something animate was struck . but did not fall or even stagger. There is a clear difference here from any kind of deductive system which, by definition, could not retain such an inconsistent pair of assertions.

The Generation System for French

Translating into French requires the addition to the system of generation patterns called *stereotypes.* Those patterns are attached to English word senses in the dic-

[1] The present paper describes the linguistic base, or basic mode, of the system. The extended mode, requiring the rules of partial information and their application to the deep structure of formulas, is described in considerable detail in [15].

tionary, both to key and content words, and are carried to the semantic block for the sentence, or paragraph,by the analysis. The block contains all that is necessary for generation, which is then a task of recursively unwrapping the block in the right way. The generation process is described in considerably more detail in [1].

A content word has a list of stereotypes attached to each of its formulas. When a word sense is selected during analysis, this list is carried along with the formula into the block. Thus, for translation purposes, the block is not constructed simply with formulas but with *sense-pairs.* A sense-pair is: (formula for a content word list of stereotypes;. We saw in the last section that each key paraplate contains a stereotype, which gets built into the block if the corresponding paraplate has been selected by the *TIE* routines. This stereotype is the generation rule to be used for the current fragment, and possibly for some of the fragments that follow it. The simplest form of a stereotype is a French word or phrase standing for the translation of an English word in context, plus a gender marker for nouns. For example:

private (a soldier):	(MASC simple soldat)
add (for a number):	(impair)
build:	(construire)
brandy:	(FEMI eau de vie)

Note that, after processing by the analysis routines, all words are already disambiguated. Several stereotypes attached to a formula do not correspond to different senses of the source word but to the different French constructions it can yield.

Complex stereotypes are strings of French words and functions. The functions are of the interlingual .context of the sense-pair and always evaluate either to a string of French words, to a blank, or (for content words only) to NIL. Hence such stereotypes are context-sensitive rules, which check upon, and generate from, the sense-pair and its context, possibly including fragments other than the current one. When a function in a content word stereotype evaluates to NIL, then the whole stereotype fails and the next one in the list is tried.

For example, here are the two stereotypes attached to the formula for the ordinary sense of "advise":

conseiller (PREOB a MAN))
conseiller)

The first stereotype would be for translating "I advised my: children to leave." The analysis routines would have matched the bare template MAN TELL MAN on the words I-advised-children. The function PREOB checks whether the object formula of the template, i.e. the formula for "children" in our example, refers to a human being; if it does, as in this case, the stereotype generates a prepositional group with the French preposition " à ."' using the object sense-pair and its qualifier list. Here this process yields " à mes enfants," and the value of the whole stereotype is "conseiller à mes enfants." For the sentence "I advise patience," however, whose translation might be "je conseille la patience," this stereotype would fail, because the object head in the template, brought in by the concept of patience, is STATE. The second is simply "(conseiller)," because no prescription on how to translate the object needs to be attached to "conseiller" when the semantic object goes into a French direct object. This is done automatically by the higher level function which constructs French clauses.

Thus we see that content words have complex stereotypes prescribing the translation of their context, when they govern an "irregular" construction: one that is irregular by comparison to a set of rules matching the French syntax onto the semantic block.

The general form of the generation program is a recursive evaluation of the functions contained in stereotypes. Thus, depending on its context of occurrence, a particular word of the French output sentence may have its origin in stereotypes of different levels: content word stereotype, or key word stereotype (or stereotypes) that are part of a set of top level basic functions. The system is formally equivalent to an i augmented transition network in the sense of Woods [17]|

Some complexity arises from the fragmented structure of the block and from dealing with the problem of integrating complex (i.e. context-sensitive) stereotypes. The program maintains a cursor which points to the fragment which is being generated from; the purpose of certain functions in a stereotype is to move the cursor up and down the block.

Integration of complex stereotypes in some contexts requires the reordering of the stereotype string. Thus, for "I often advised him to leave" going into "Je lui ai souvent conseille de partir," the stereotype: (conseiller (PREOB a MAN)) needs to be rearranged. This is done by a feature which permits the values of designated functions in a stereotype to be lifted and stored in registers. The values of these registers can be used at a higher level of recursive evaluation to construct a new correct French string.

Finally, the integration of complex stereotypes requires the implementation of a system of priorities for regulating the choice of generation rules. Since any word or key can dictate the output syntax for a given piece of the block, there may arise conflicts, which are resolved by having carefully settled priorities. The principle, as in the analysis program, is that a more specific rule has priority over a more general one. Thus, when a content word stereotype prescribes the translation of fragments other than its immediate context, it has priority over any key stereotype. This important process of a stereotype controlling the generation of other fragments than the one to which it attaches is also described in detail in [11.

Implementation

The system is programmed in LISP 1.6 and MLISP and runs on line on the PDP 6/10 system at Stanford Artificial Intelligence Laboratory where it is the system dump named MT. It runs at present over a vocabulary of about 600 words and takes texts of up to small paragraph length. There is no morphology in the system at present, every input and output word being treated as a separate LISP atom, since morphology presents no substantial research questions to compare with those of semantics. An English sentence is input and a French output, as it might be as follows to show the ambiguities of the preposition "out of":

I PUT THE WINE ON THE TABLE AND JOHN DRINKS IT OUT OF A GLASS. HE OFTEN DRINKS OUT OF DESPAIR AND THROWS THE GLASSES OUT OF THE WINDOW.

JE METS LE VIN SUR LA TABLE ET JEAN LE BOIT DANS UN VERRE. IL BOIT SOUVENT PAR DESESPOIR ET JETTE LES VERRES PAR LA FENETRE.

After this follows the usual cpu time declaration and the line *(common sense inferences called)* if the extended anaphora procedures using partial information are required. After that comes the whole semantic block for diagnostic purposes.

The format of the block is a list, each item of which, at the top level, is a text fragment tied to a template, the template being a list of pairs (of formulas and generation stereotypes) and of sublists of such pairs that are dependents on the main nodes of the template in the manner described above. In the lists at the same level as the text fragments are the key generation stereotypes for fragments, as well as paraplate and inference nodes that declare satisfactory preferred ties.

The block is clearly not wholly target-language independent because it contains the generative rules; however, it is very largely so. Moreover, the semantic representation it expresses could easily be adapted as a data base for some quite different task, such as question| answering. Indeed, many of the inferences required t set up the block, like those described in detail above, are equivalent to quite sophisticated question-answering.

[2]By e use of nonhierarchical here, I would mean the connected linear structures I have described, each one approximating anotion of nuclear "message."

[3] The common sense reasoning exhibited here is of a quite different sort from other programs in linguistics and artificial intelligence, and the only other systems to use "partial information" of this sort and Schank's and Rieger's [8 and 9].. Their systems and this one share far more similarities than differences. The main points of contrast concern: (a) the fact that the Preference Semantics system emphasizes the notion of choice between alternative competing structures for a piece of language; (b) a more general contrast in that the description of this system is weighted more toward the solution of concrete problems and the application of the system to actual text rather than being the description of a static network of concepts; and (c) the clear differences in the notion of "phenomenoliogical level" the other systems employ in describing common sense reasoning: Preference Semantics tries to avoid imposing highly rationalist analyses of cause and mental phenomena that are very hard to justify in terms of common sense—if that is indeed to be the basis for understanding ordinary language.

Discussion

I have presented and argued for a nonstandard approach to the computational semantics of natural language and, by implication, against the more conventional linguistic approaches, as well as those from artificial intelligence that assume that natural language is approximated by restricted micro-worlds of simple object words, and the use of theorem proving methods.

In particular, I think the onus is on those who believe in strictly linguistic approaches to show the psychological and computational importance of the structures they impose with considerable difficulty upon; even simple sentences. The present work suggests that a well defined semantic structure is the heart of the matter, that the "semi-parsing" of this system may be sufficient to support such structures, and that the heavily hierarchical syntax analyses of yesteryear may not be necessary.[2,3]

Received March 1973; revised. July 1974

References

1. Herskovits, A. The generation of French from a semantic representation. Stanford Artif.. Intell. Proj. Memo # 212, Aug. 1973.
2. Joos, M. Semantic Axiom # !, *Language {*19721. 193-211.
3. McCarthy. J., and Hayes, P. Some philosophical questions from the standpoint of artificial intelligence. !n Melzer and Michie (Eds.) *Machine Intelligence 4,* American Elsevier, New York, 1969.
4. Minsky, M. In Minsky (Ed), *Semantic Information Processing,* MIT Press, Cambridge, Mass., 1*968.*
5. Minsky, M. Frame systems. Unpublished ms., MIT, Feb. 1974.
6. Papert, S. *The Romanes Lectures.* U. of California, Berkeley, 1973.
7. Sandewall, E. Representing natural language information in predicate calculus. In Melzer and .Michie (Eds.) *Machine Intelli gence 6,* American Elsevier, New York, 1971.
8. Schank, R. Conceptual dependency. *Cognitive Psychology* (1972), 82-123.
9. Schank, R., and Rieger, C. Inference and the computer under standing of natural language. Stanford Artif. Intel. Lab. Memo * 197, May 1973.
10. Scriven, M. The concept of comprehension. In Carroll & Freedle (Eds.), *Language Com prehension,* Washington, D.C., 1972.
11. Simmons, R., and Bruce, B.C. Some relations between predicate calculus and semantic net representations of discourse. Proc. Second Internal. Joint Conf. on Artif. Intel., London, 1971.
12. Wilks, Y. Decidability and natural language. *MIND* (1971), 218-239.
13. Wilks, Y. *Grammar, Meaning and the Machine Analysis of Language.* Routledge & Kegan, London, 1971.
14. Wilks, Y. An artificial intelligence approach to machine trans lation. In Schank & Colby (Eds.) *Computer Models of Thought and Language,* San Francisco, 1973.
15.
16. Wilks, Y. A preferential, pattern seeking, semantics for natural language inference. To appear in *Artif. Intel.*
17. Winograd, T. *Understanding Natural Language,* Academic Press, New York, 1972.
18. Woods, W.A., Procedural semantics for a question-aaswer machine, Proc. AFIPS 1968 FJCC Vol. 33, pp. 457-471. AFIPS Press, Montvale, N.J.