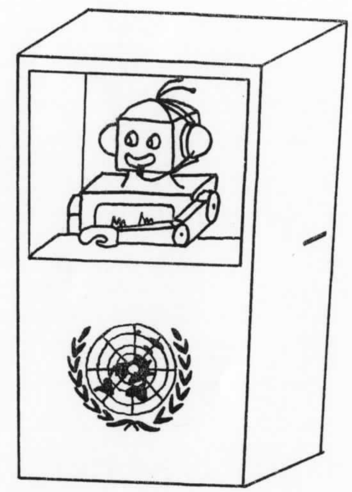
One could multiply such problems several times over, but let us conclude with a problem which affects all of the systems mentioned in this chapter — that of content. It probably has not escaped the reader's notice that each time in our analysis by means of our five questions, when we came to "content" the answer was always the same. In none of the systems had the author attempted to formalize an appreciable amount of the knowledge we need to understand even the simplest stories. Many reasons can be given for this, but two stand out. First, the problem of content is tough. (Charniak 1974) attempts to analyze the information needed to understand the last line of story (16) in Chapter 1 ("He will make you take it back"). In something like 100 pages he comes up with an admittedly incomplete analysis. Secondly, one cannot really get very far on the problem of content until one has at least some idea of what one's representation looks like, else one is left writing a list of English sentences, which hardly solves the problem. Hence it is not until one has some faith in one's representation that one can even try formalizing a sizable chunk of world knowledge.

On the other hand, the picture is not as bleak as the preceding paragraphs might make it seem. Our state of knowledge is steadily improving, even if our goal is still far off. Compare the frame analysis to, say, Raphael's SIR (Chapter 1). Raphael's system was quite limited, as was inevitable at that time. SIR handled "A man has two hands. A hand has five fingers. How many fingers does a man have?", but it is not clear how to extend that to understanding the point of getting a basket in a supermarket. The new systems understand language to a degree unimaginable in these earlier systems, and that is some sort of progress.

E. Charniak and Y. Wilks (eds.), Computational Semantics



PARSING ENGLISH II

Yorick Wilks

This chapter continues the discussion of systems that aim to represent the structure of natural language, and to parse those structures onto natural language. At the end of the survey, the systems discussed, in this and other chapters, will be contrasted and compared along several dimensions.

One important thing to bear in mind in what follows is that "parsing" is being used not only in its standard sense in mathematical and computational linguistics: a sense which is completely neutral about the content, value, or interest of the structure which is parsed onto natural language. So, for example, the general notion of parsing was illustrated in Chapter 7 by means of a phrase structure grammar, but the reader will have gathered already that the whole argument of this book is that far richer meaning and knowledge structures than those produced by phrase structure grammars are needed if we are to do anything interesting. So then, "parsing" in what follows is not neutral about what Charniak in Chapter 1 called the content of representations, but rather assumes that to be worth parsing they should have something to say about what, in Chapter 7, were called the intractables of systematic ambiguity in natural language: namely, word sense ambiguity, structural case ambiguity, and ambiguity of pronoun reference.

Second Generation Systems

To understand what Winograd meant when he contrasted his own with what he called second generation systems, we have to remember, as often in this subject, that the generations are of fashion, not chronology or inheritance of ideas. He described the work of Simmons, Schank and Wilks among others in his survey of new approaches, even though the foundations and terminology of those approaches were set out in print in 1966, 1968 and 1967 respectively. What those approaches, and others, have in common is the belief that understanding systems must be able to manipulate very complex linguistic objects, or semantic structures, and that no simplistic approaches to understanding language with computers will work.

It was suggested in Chapter 7 that, as a first approximation only, such second generation approaches could be seen as within what Minsky has called the "frame approach", as described in Chapter 8.

The key point about any structures that are to be called frame-like is that they attempt to specify in advance what is going to be said, and how the world encountered is going to be organized. The structures, and the inference rules that apply to them, are also expressions of something not present in first generation systems, namely "partial information" (in McCarthy's phrase, and meaning rules like "people try to get what they want": rules which express inductive generalizations that are useful but not always true). As was shown earlier in Chapter 7, with the "women and soldiers" example, such loose inductive information, seeking confirmation from the surrounding context, is required for very simple sentences. In psychological and visual terms, frame approaches envisage an understander as at least as much a looker as a seer.

The reader should notice that the characterization of frame-like systems just given is far more general than that set out by Charniak in Chapter 8, and is intended to serve only to introduce three more systems: Colby's, Riesbeck's implementation of Schank, and Wilks'. After that the more general and more specific characterizations of "frame" will be drawn together.

Colby

Colby (Colby et al 1971), (Enea and Colby 1973), (Colby and Parkison 1974) has developed over a period of several years an on-line dialogue system that simulates a paranoid patient called PARRY. A measure of its success is that it is perhaps the most used AI program in existence; used, that is, by those outside the research project that developed it. Here is a discontinuous sample of its output, again with the system's responses in the upper case:

Have you been hospitalised before?

THIS IS THE FIRST TIME.

How long have you been there?

ABOUT TWO WEEKS.

Any headaches?

MY HEALTH IS FINE.

Are you having any difficulty with your memory?

NO.

Then perhaps you can tell me who is the President of the United States?

NIXON IS PRESIDENT.

What day is it today?

TODAY IS THURSDAY.

What season is it?

LET'S GET ON WITH THE INTERVIEW.

Colby's aim was to produce a system that was not, in his word, "fragile" in the way that most AI systems are: one that did not break down, or give up, if the input was not exactly what it expected, but one that carried on somehow in all circumstances, in the way that conversing humans do.

Fragility is avoided by having no conventional syntax analyser, and by not even attempting to take account of all the words in the input. This is a considerable aid, since any parser that begins to parse a more than usually polite request such as "Would you be so kind as to..." is going to be in trouble. British English speakers arriving in the U.S. quickly learn not to use such phrases, since they cause great confusion to human listeners in stores.

The input text is segmented by a heuristic that breaks it at any occurrence of a range of key words. Patterns are then matched with each segment. There are at present about 1,700 patterns on a list (Colby and Parkison, in press) that is stored and matched, not against any syntactic or semantic representations of words (except to deal with contractions and misspellings), but against the input word string direct, and by a process of sequential deletion. So, for example, "What is your main problem?" has a root verb "BE" substituted, and a generic "YOU" for "YOUR", to become

WHAT BE YOU MAIN PROBLEM

It is then matched successively against the stored pattern list in the following forms, each of which is formed by deleting one word from the original string:

BE YOU MAIN PROBLEM

WHAT YOU MAIN PROBLEM

WHAT BE MAIN PROBLEM

WHAT BE YOU PROBLEM

WHAT BE YOU MAIN

Only the penultimate line exists as one of the stored patterns, and it is therefore matched by what we might call this minimal parsing procedure.

Stored in the same format as the patterns are rules expressing the consequences for the "patient" of detecting aggression and over-friendliness in the interviewer's questions and remarks. The matched patterns found are then tied directly, or via these inference rules, to response patterns which are generated.

Enormous ingenuity has gone into the heuristics of this system, as its popularity testifies. The system has also changed considerably: it is now called PARRY2 and contains the above pattern matching, rather than earlier key word, heuristics. It has the partial, or what some would call "pragmatic", rules about expectation and intention, and these alone might qualify it as "second generation" on some interpretations of the phrase. A generator is also being installed to avoid the production of only "canned" responses.

Colby and his associates have put considerable energy into actually trying to find out whether or not psychiatrists can distinguish PARRY's responses from those of a patient (Colby and Hilf 1973). This is probably the first attempt actually to apply Turing's suggested (1951) test for machine-person distinguishability by comparing outputs. There are statistical difficulties about interpreting the results but, by and large, the result is that the sample of psychiatrists questioned cannot distinguish the two. Even so, there are many who still, on principle, believe that PARRY is not a simulation because, in their terms, it "does not understand". People who argue that way sometimes implicitly use very high standards for what it is to understand. One could extend their case ironically by pointing out that very few people understand the content of sentences in the depth and detail that an analytic philosopher does, and a very good thing too. But there can be no doubt that many people on many occasions DO seem to understand in the way that PARRY does.

When it was said earlier that PARRY was "robust" as regards the way it dealt with unusual or unexpected input, that did not mean that one would get a very sensible reply if one typed in "***", but, of course, doing things like that is not a very sinsible way to test a language program. What is clear is that, because of its very simple construction, PARRY has been much easier to extend to cover new input than most programs.

The real defect of PARRY is that it is "fragile" in a more general way, namely that much of its "understanding" is due to the particular environment of the imitation of paranoid behavior. The user of the program tends to tolerate evident "refusals to co-operate" and abrupt "changes of subject" by the program, on the ground that a paranoid might well behave like that, although, in language understanding terms, this behavior is the result of the system being unable to make any sense (in terms of effective pattern matching) of the input it is receiving.

Wilks

This system constructs a semantic representation for small natural language texts: the basic representation is applied directly to the text and can then be "massaged" by various forms of inference to become as deep as is necessary for tasks intended to demonstrate understanding. It is a uniform representation, in that information that might conventionally be considered as syntactic, semantic, factual or inferential is all expressed within a single type of structure. The fundamental unit of this meaning representation is the template, which corresponds to an intuitive notion of a basic message of agent-action-object form. Templates are constructed from more basic building blocks called formulas, which correspond to senses of individual words. In order to construct a complete text representation (called a semantic block) templates are bound together by two kinds of higher level structures called paraplates and inference rules. The templates themselves are built up as the construction of the representation proceeds, but the formulas, paraplates and inference rules are all present in the system at the outset, and each of these three types of pre-stored structure is ultimately constructed from an inventory of eighty semantic primitive elements, and from functions and predicates ranging over those elements.

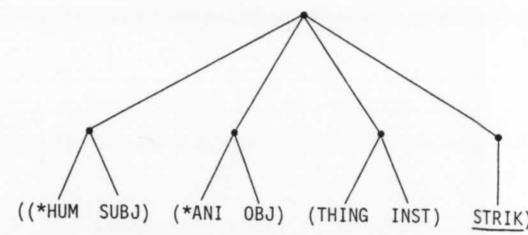
The system runs on-line as a package of LISP (see Chapter 12), MLISP and MLISP2 programs, the two latter languages being expanded LISP languages that have a command structure and pattern matching capacities. It takes as input small paragraphs of English, that can be made up by the user from a vocabulary of about 600 word-senses, and produces a good French translation as output for a considerable range of input texts. This environment provides a pretty clear test of language understanding, because French translations for everyday prose are in general either right or wrong, and can be seen to be so while, at the same time, the major difficulties of understanding programs — word sense ambiguity, case ambiguity, difficult pronoun reference, etc. — can all be represented within a machine translation environment by, for example, choosing the words of an input sentence containing a difficult pronoun reference in such a way that the possible alternative references have different genders in French. In that way the French output makes quite clear whether or not the program has made the correct inferences in order to understand what it is translating. The program is reasonably robust in

actual performance, and will even tolerate a certain amount of bad grammar in the input, since it does not perform a syntax analysis in the conventional sense, but seeks message forms representable in the semantic structures employed.

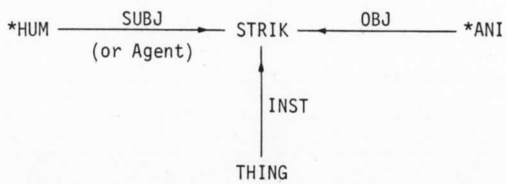
Typical input would be a sentence such as "John lives out of town and drinks his wine out of a bottle. He then throws the bottles out of the window." The program will produce French sentences with different output for each of the three occurrences of "out of", since it realizes that they function quite differently on the three occasions of use, and that the difference must be reflected in the French. A sentence such as "Give the monkeys bananas although they are not ripe because they are very hungry" produces a translation with different equivalents for the two occurrences of "they", because the system correctly realizes, from what will be described below as preference considerations, that the most sensible interpretation is one in which the first "they" refers to the bananas and the second to the monkeys, and bananas and monkeys have different genders in French. These two examples are dealt with in the "basic mode" of the system (Wilks 1973a). In many cases it cannot resolve pronoun ambiguities by the sort of straightforward "preference considerations" used in the last example, where, roughly speaking, "ripeness" prefers to be predicated of plant-like things, and hunger of animate things. Even in a sentence as simple as "John drank the wine on the table and it was good", such considerations are inadequate to resolve the ambiguity of "it", between wine and table, since both may be good things. In such cases of inability to resolve within its basic mode, the program deepens the representation of the text so as to try and set up chains of inference that will reach, and so prefer, only one of the possible referents.

The system contains no explicitly syntactic information at all: what it knows about any English word sense is its formula. This is a tree structure of semantic primitives, and is to be interpreted formally using dependency relations. The main element in any formula is the rightmost, called its head, and that is the fundamental category to which the formula belongs. In the formulas for actions, for example, the head will always be one of the primitives PICK, CAUSE, CHANGE, FEEL, HAVE, PLEASE, PAIR, SENSE, USE, WANT, TELL, BE, DO, FORCE, MOVE, WRAP, THINK, FLOW, MAKE, DROP, STRIK, FUNC or HAPN.

Thus:

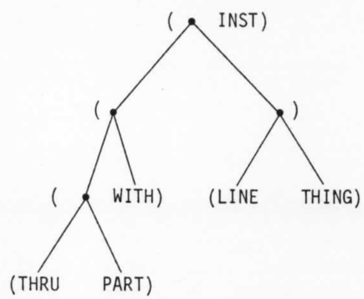


is the formula for the action sense of "beat", and is an action preferable of STRIKing, normally done with an INSTrument (see Chapter 5 on case elements) that is a THING, preferably to an ANImate OBJect, and preferably by a HUMan agent. Asterisks in front of elements (like *HUM) indicate that a class of elements is in question; in this case *HUM covers MAN and FOLK (for human groups). The tree structure is largely superficial so as to get a particular LISP representation and the same semantic entity could be written as Chapter 5 showed Simmons writing his Fillmorian case dependencies, thus:



This form emphasizes that the three case parts at the "top level" of the formula are in some fundamental relation to the basic action, or head of the formula (always its right-most member and underlined here). However there are limitations to this re-picturing, as with all picturings. For example, it makes little sense for formula trees representing noun senses, since the head of such a tree (say, THING for a tree for "needle", or STUFF for a tree for "water") will not have case subparts depending on it in the straightforward way that the head of an action tree, like the above one for "beat", does.

The case subparts of a formula can themselves contain case-subparts and even action elements, down to whatever level is thought necessary by the dictionary maker. So, for example, the formula for "sew" would contain an instrument subpart which would specify that the preferred instrument was a linear object (LINE THING) accompanied by an aperture (THRU PART) — that is, of course, a needle. This sub-tree (of the whole tree for "sew") might look like:

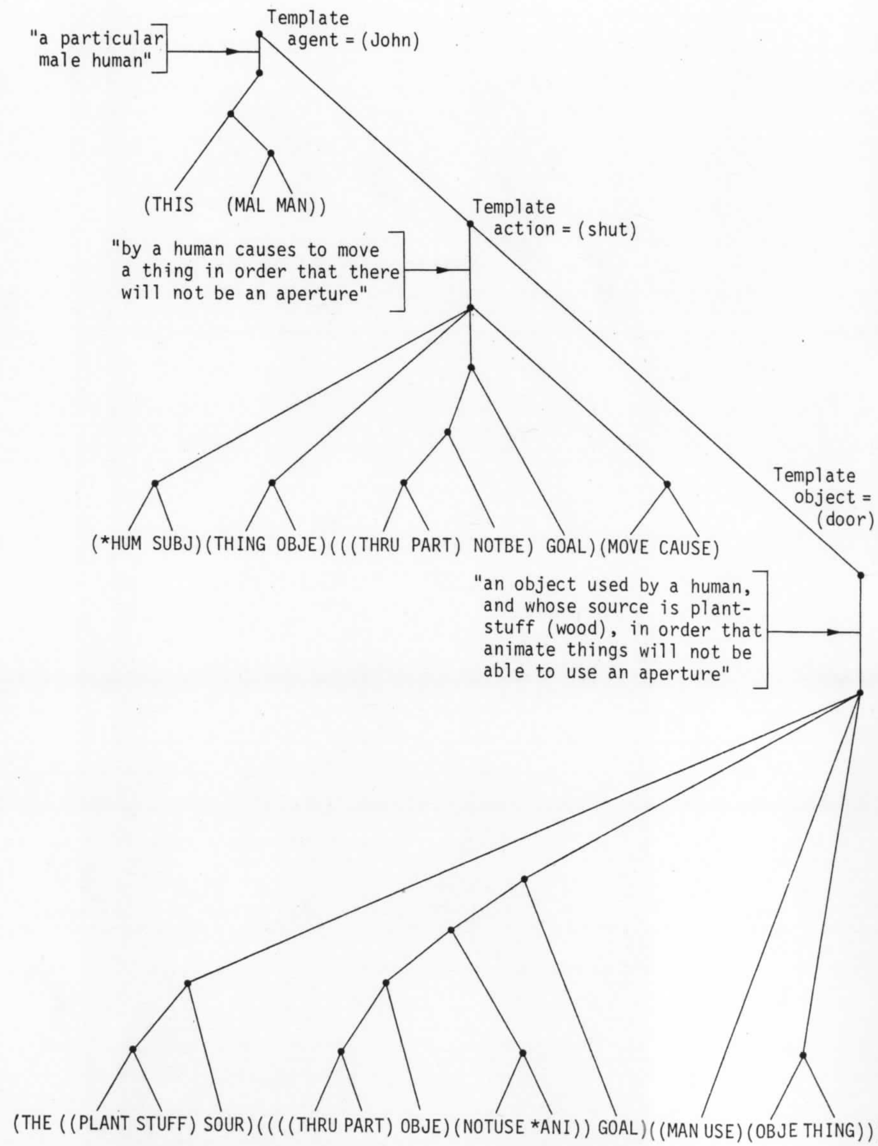


where we have an accompaniment case specification inside an instrumental specification. This form of representation allows <u>order</u> to be of critical importance, which it cannot be in the pictorial-graph style, and it allows far more flexibility in expressing the meaning of noun-senses than can be achieved by the Fodor-Katz method (see Chapter 3) which has no case or action elements (Schank's system is a variant of that method, see Weber (1972)).

Template structures, which actually represent sentences and their parts are built up as networks of formulas like the one above. Templates always consist of an agent node, and action node and an object node, and other nodes that may depend on these. So, in building a template for "John beat the carpet", the whole of the above tree-formula for "beat" would be placed at the action node, another tree structure for "John" at the agent node and so on. The complexity of the system comes from the way in which the formulas, considered as active entities, dictate how other nodes in the same template should be filled.

Thus, a full template is a complex structure. Here, for example, is one for "John shut the door":

(1)



The article "the" is not given special semantic representation, but is simply attached, as shown, to the corresponding formula (as would be a PAST element to the "shut" formula). Although all templates have this basic form of agent-action -object formulas, other formulas can be attached dependently to any of the nodes. So, if the sentence were, "John shut the green door", a formula for "green" would be attached to the object node above.

How are these template structures actually parsed onto text? The process begins with fragmentation: the program goes through a sentence breaking it at key words like prepositions, subjunctions, etc. This is done only on the basis of the list of key words plus the range of semantic formulas for each (non-key) word. A sentence like "The sort of man | that dogs need | is kind" would be fragmented at the two strokes as shown. Even this requires use of the formula semantics, so as to see that the "that" is a relative that, and not a qualifier as it would be in "The dog likes that man".

Normally each fragment of a sentence, produced in this way, has one or more templates matched onto it — though in the case above "The sort of man is kind" would be immediately rejoined into a single fragment.

Next come the procedures for matching templates onto fragments. First the parser goes through the formula combinations (for there will in general be more than one formula for each word, as formulas correspond to word-senses) for a fragment from left to right looking only at the head elements of the formulas. It looks for agent, action and object formulas, in that order initially. So, since all qualifier (adjective) formulas have KIND as their head, this procedure would never take a KIND-headed formula as a possible agent. This process is bare template matching, and is more complex than I have suggested here because fragments of text do not always have their main items in convenient agent-action-object order. So, there is a scale of preferred head sequences. Again, many fragments do not have all three formula types: "John left" has no object, naturally enough, and so that node is filled out by a dummy so as to retain a canonical form of template. Finally, prepositions in English are always assigned to the action node, and so will always have corresponding dummy agent nodes (a point that will be dealt with in detail later).

The matter of matching is also more complicated than suggested initially when it was said that templates were built up from whole formulas during analysis. The whole template for, say, "John shut the door" given above is built up in that way, but the system already knows that a template based on the three ordered heads MAN CAUSE THING is a possible template in a way that KIND CAUSE THING would not be. Thus bare templates (or template types) consisting of just three ordered elements do pre-exist in the system and are indeed matched onto text fragments.

Then comes a more interesting matching process, let us call it preferential expansion. Templates so far have been constructed by assigning formulas to nodes only on the basis of their head elements. Thus in "John shut the case", we would have the "correct" template similar to (1) above, but we would also have one containing a "legal case" formula at its object node, with a different head, GRAIN — meaning a structure, of arguments in this example.

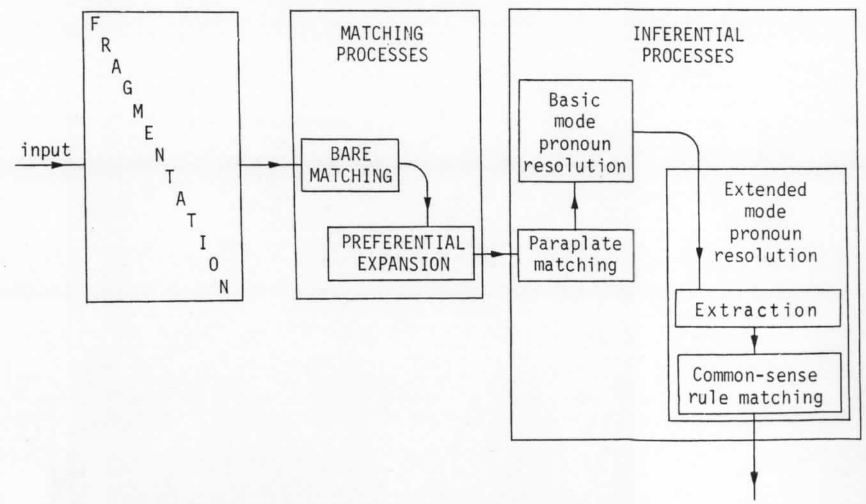
Expansion requires that each possible template for a fragment is taken in turn, and its formulas are examined to see which of their preferences are satisfied in the template in which they find themselves.

Formula subparts were explained earlier as preferring certain semantic entities, not requiring them to be present. Thus the formula for "shut" in (1) prefers a human agent and a physical thing as object. Only the template with the physical object sense of "case" (whose formula head is THING) will satisfy both preferences expressed in the formula for "shut", since a GRAIN formula cannot stand for a physical object. Not only the action formula, but all those making up the template, are examined in this way.

This seeking sorts of agents, objects, etc. is only preferential, in that formulas not satisfying that requirement will be accepted, but only if nothing more satisfactory can be found. The template finally chosen for a fragment of text is the one in which the most formulas have their preferences satisfied. There is a general principle at work here: that the right interpretation "says the least" in information-carrying terms. This very simple device of counting preferences is able to do much of the work of a conventional syntax program as well, since, at the same time as resolving word-sense ambiguity as above, it ensures that "adjective senses" are correctly assigned instead of "noun senses" — in "the green door" — and so on.

The formulas are therefore not just static objects expressing the meaning of words, for in processes like the one just described, they actively dictate how templates are to be constructed. In terms of the bottom-up versus top-down distinction developed earlier, preference methods do not fall neatly into either type, but are a bottom-up selection of a preferred structure which is then applied in a top-down manner.

So far, we have encountered the fragmentation process, followed by two stages of the template matching process. A two stage inferential process then follows, by the end of which a semantic text representation has been constructed. The whole process can be illustrated by a flow chart:



Let us now turn to the inferential processes that tie templates together, so as to produce a semantic block as a text representation — a semantic block is an ordered sequence of templates connected by labelled ties yet to be described. If it can be constructed for a text, then, as far as the system is concerned, all semantic and referential ambiguity has been resolved and the present program will begin to generate French by unwrapping the block again. The generation aspects of this work have been described in (Herskovits 1973). Thus, with the present program, the flow chart above would go on to a French generation program, though it could equally well go to a question-answerer or something of that sort.

From now on, to avoid expressing templates in the detail of (1), they will be written in square brackets containing the English words or dummies for at least the three node-senses. However, it should be remembered that objects in [ ] are complex semantic objects and not the English words they contain.

Paraplates are another type of structure that serves to connect two templates, normally a template representing a main clause and one representing a prepositional phrase. Thus, if we were representing

John left his clothes at the cleaners

which would be represented initially in the system by two templates, the correct paraplate, when applied, would assert a spatial-location case tie (SLOCA) between the two as follows:

```
        [John                 left his + clothes]
SLOCA  ⌐
        →[( □ = Dummy Agent) at the + cleaners]
```

What is the shape of a paraplate in virtue of which it does this? Each paraplate expresses one of the cases (which is the same inventory as used inside formulas) in this way, and there are many paraplates expressing any particular case. Each paraplate is stored under some English preposition, each of which may of course realize many cases.

Thus let us consider "by", as it functions in the following sentences, all of which may be considered to start with "He left Comano by ...", where I have indicated at the right of each line the apparent case of the last clause and thus the type of paraplate that should connect the two clauses on a given line.

(2)	He left Comano by courtesy of the police,	SOUR
(3)	by the autostrada,	DIRE (direction case)
(4)	by car,	INST
(5)	by stealth,	WAY (manner case)
(6)	by Monday night,	TLOCA (time location case)
(7)	by following the arrows,	DIRE
(8)	by stealing a boat	INST

Paraplates are in principle six-place entities that are the "skeletons" of a pair of templates, not all of whose places need be filled, corresponding to agent-of-first-template, action-of-first-template, object-of-first-template, and so on for the second template (except that the agent and object of the second are rarely filled). Paraplates, unlike templates, have filtering functions at their nodes that any template pair "matching" a given paraplate must satisfy at every node. Here are four paraplates that should match onto the templates for the sentences with corresponding numbers. Like the sentences above, the paraplates will all have the same left-hand side, which is written only once.

(3')	(*ANI) (MOVE) (WHERE POINT)	→(DIRE) □ □	(WHERE LINE)
(4')		→(INST) □ □	(*REAL)
(7')		→(DIRE) □(*DO)	(WHERE SIGN)
(8')		→(INST) □(*DO)	(*REAL)

where *DO covers a wide class of action heads, and the brackets containing formula parts are all to be interpreted as: matches onto corresponding part of a template if the latter has the mentioned subparts of formula. Thus (3') matches (3) because:

the formula for "he"	falls under (has as appropriate subpart)	*ANI
"left"	"	MOVE
"Comano"	"	(WHERE POINT)
"autostrada"	"	(WHERE LINE)

and so on for the other correspondences of sentences and paraplates. The result of the paraplate (3') matching the two templates for (3) is that the case label DIRE on the arrow of (3') is asserted as holding between the two templates that represent (3). In more complex cases there may be two or more templates available for a sentence fragment, and the template that matches with a "more preferred paraplate" will ultimately represent that fragment. As we shall now see, paraplates, for a given preposition, are ordered and "more preferred paraplate" simply means "the paraplate applied earlier".

The above are some of the paraplates attached to "by" and brought in whenever a "by" template is to be tied to others. They are organized under the action element on the left of the paraplate, and the paraplate stack for a given preposition and primitive action is ordered, though not simply ordered, as we shall see.

So INST case is largely a default case for MOVE as it is cued in by "by", in the sense that almost any entity can be an instrument here if we have no reason to believe it is something else. Thus the more specific (3') must be applied before (4') in order to match direction case for (3) since, if the order were reversed (4') might match with what "ought to" match with (3'). We could imagine some very specific requirement being expressed by a function at the last (sixth) node of (4') — say, one equivalent to "entity for transporting humans" instead of the very wide function (*REAL), meaning any real object or substance, at present in (4'). If that were done then (4') could be put above (3') in the stack and would catch (i.e. the sixth node function would be satisfied by) sentence (4). But if the sentence were "He left Comano by cattle truck" it would fail (4') — since cattle trucks are not normally used for transporting humans — then fail (3'), which is now below (4'), and now there would be no appropriate default paraplate to catch the sentence and assign the default case INST to its internal tie.

This is the sort of reasoning that establishes the need for a preferential order of application paraplates. But it is not a simple (linear) order since, for example, there is no reason why (7') should be ordered with respect to (3'). An order that takes account of that indifference is called a partial ordering (which has no connexion with partial information as defined earlier).

Thus paraplates have a general structure <template form> → <template form>, and so it is reasonable to call them inferential structures in a wide sense, and this idea will not be strange to readers who will have encountered Schank's system in Chapters 4 and 7 where his case expansions of conceptualizations are called "inferences".

As described at the beginning of the Wilks section, the system has also tied templates together by means of simple preferential substitutions for problem pronouns, as with the "monkeys and bananas" example. Solving that example ties templates in the sense that the process can be thought of as imposing a tie from the "pronoun node" in one template to the "noun node" (the noun that the pronoun refers to, that is) in another template.

A general point should be made here about the treatment of the problem of pronoun reference by the systems described in this book, that is: Winograd's (Chapter 6), Rieger's (Chapter 8, which did a considerable amount of indirect pronoun referencing via Riesbeck's analyzer — see below), Charniak's (Chapter 8, which was originally designed to refer pronouns in the general process of analyzing children's stories) and Wilks', in the present discussion. The point is that none of these systems has a single uniform method for resolving pronouns to the nouns or phrases they refer to, because pronoun reference is not a simply structured problem, and so not open to solution by a uniform method.

For example, it is a convenient fiction, encouraged by almost all systems, that a pronoun refers to some item mentioned earlier (or occasionally later). But this is not so in general, as can be seen from such sentences as "It is hot in here" or "In Italy he drank mineral water in his wine and it tasted good" where, in the last example, the "it" refers to a combination not actually mentioned in the sentence. Again, selection restrictions, preferentially applied or otherwise are sometimes sufficient to resolve pronouns, as in the "bananas and monkeys" example, but it is easy to construct other examples where they are not enough — and we shall turn to several in a moment. There is the well known rule, often half remembered from school, that a pronoun refers to the last mentioned object "of the right sort", and sometimes, usually if we have nothing else to go on, that rule can explain how we normally refer a pronoun, as in "John took the wheel off his bike and then the saddle and left it in the garage", where "it" would normally be taken to be the last thing mentioned, the saddle. But, of course, if we read "The log floated past the tree and then the fish swam under it" we know fairly surely that it was the log the fish swam under and not the, last mentioned, tree.

In this last example we might well want to say that we refer the pronoun as we do because of what we know about floating and swimming, and trees are not normally floating.

What we have, in common sense terms, is a hotch-potch of different ways of referring pronouns in different circumstances. If any of them is offered as the principal rule, it is not too hard to find exceptions to it. The danger is then of going to the other extreme and saying that these various general rules are no use because they don't work in all cases. That too is wrong: norms are not less valuable because they are sometimes violated.

There are roughly two main sorts of approach to the pronoun reference problem to be found in AI systems. On the one hand are those who concentrate on very general manipulations of their systems and hope that much pronoun reference will be dealt with correctly if those manipulations are properly performed. This style emphasizes the central examples rather than the exceptions, and also tends to classify different types of pronoun reference in terms of the operations of the systems themselves, and Wilks' and Schank's systems emphasize this approach.

On the other hand are approaches which emphasize special routines to deal with special sorts of circumstance, and pronoun reference problems arising in connexion with description of those types of circumstance: Winograd's "specialists" (Chapter 6) and Charniak's "demons" (Chapter 8) are examples of such routines. This approach tends not to emphasize the role of the system of representation, and tends to concentrate on difficult, rather than central, examples in that it is interested in examples which violate the norms and seldom in those which conform.

In practice, all working systems are a combination of these two approaches, and none could yet be said to have done more than scratched the surface of this enormous problem.

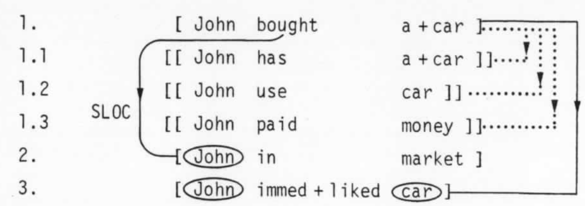
After this general digression on pronoun reference, let us return to Wilks' system. Many examples are not resolved by the simple preferential methods of his "basic mode" and, in particular, if a word sense ambiguity, or pronoun reference, is still unresolved, then a unique semantic block of templates cannot be constructed and the "extended inferential mode" is entered. In this mode, new template-like forms (called extractions) are inferred from existing ones, and then added to the template pool on which common sense inference rules then operate.

Let us illustrate these final parsing processes by looking at the sentence:

(9) John bought a car in the market and liked it immediately.

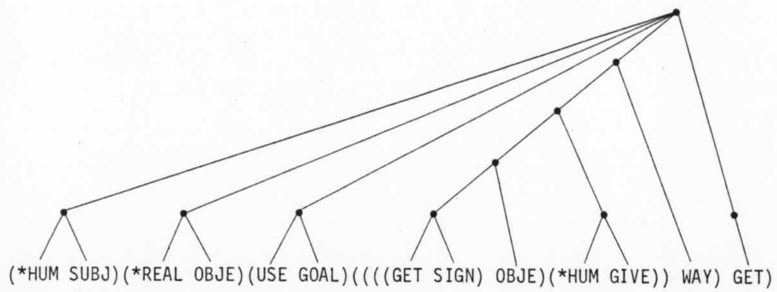
The final semantic block for this will be (schematically):

(10)



where "car" and "John" are ringed to show that they are the final resolved values of the formulas at those nodes. The "John" nodes have been replaced by simple preferential procedures in the basic mode, and we shall concentrate on the "car" node that replaces "it". This representation is the result of three separate processes operating on the original three templates (# 1, 2, 3) representing the text. First, the SLOC tie has been inserted by paraplates in the way described. Secondly, 1.1, 1.2 and 1.3, in double brackets, are extractions which are added to a representation prior to the application of inference rules. They have been extracted from # 1, as is shown by the dotted arrows connecting them. The main type of extraction is one which comes from the "unpacking" (see below) of each case type (except AGENT and OBJECT) in the formulas of the source template (# 1).

Thus, we have in the formula for "bought" (see below) a GOAL subpart which says that the purpose of buying something is to use it, hence from the template # 1, 1.2 has been extracted by "unpacking" the GOAL subpart. Similarly 1.3 follows by an extraction of the WAY (or manner) case in # 1. 1.1 intuitively is a causal consequence extracted from # 1, since "buy" is basically a GET action, its formula being:



That is to say, in semi-English, buying is a getting, done by humans to real objects in order to use them, the manner being the human giving money = (GET SIGN). Thus, the three extraction arrows in the representation above could in fact be labelled CAUSE, GOAL and WAY respectively.

Extractions, then, are new template-like entities added to a representation to deepen it. They represent information not explicitly present in the original text, information that may not even be necessarily true. After all John may have bought the car (# 1) but may not have it (# 1.1), because he re-sold it immediately. Extractions are produced by "unpacking" the case subparts of formulas and allowing them to produce template-like items as if all the case preferences in the formula had in fact been satisfied. Since the formula is installed in a template information can be found, usually in the rest of the template, that shows what exactly such satisfaction would be. That is, if the GOAL of buying is to use something, the system can see from the object node of template # 1 what the something is. Extractions, then, are generally local, from within a template.

Thirdly we have common sense inferences (square "template → template" links in (10)). #3 has been tied to #1 in the representation above by an inference arrow (shown as a square line) on the basis of which the ?it in #3 has been correctly replaced by the formula for "car". Inference rules have the form <template> → <template> but always contain restricted variables. In this case the rule tying #1 to #3 would have been:

[animate 1 cause + self + have realobject 2] → [1 *judges 2]

where the variable 1 is restricted to be something animate: "*judges" represents a general function satisfied by actions of liking, as in the present example; and the square brackets show that we are dealing with only the "readable shorthand" form of the rule, for its "full form" would be:

((*ANI 1)((SELF HAVE) CAUSE)(*REAL 2)) → (1 *JUDG 2)

The left-hand side matches onto template #1 and the right-hand side onto #3. Thus resolving ?it at the object node of #3 as the same entity as that filling the variable #2 on the left-hand side, i.e. the car.

It should be noticed that these inference rules are only partial: they are not in general consequences that must follow, in any sense. The assumption is that if they can match and so fill "gaps" in this way, then the solution they yield is the correct interpretation of the text. Moreover, the rules can be chained so as to reach from a template (or extraction) containing a problem pronoun to one containing a possible solution, and the "preferential assumption" is that the shortest possible chain of such inferences will be the right one, which is not to deny that some other longer chain of rules would reach the other, less preferred, solution ("market" in the example above). A similar hypothesis of "bridging inferential gaps" by the shortest determinate chain has recently been explored, though from a psychological point of view, by Clark (1975).

There is also a considerable overall similarity of approach to that of the "demon strategy" of Charniak described in Chapter 8, though not as regards the shortest chain preference. In that chapter Charniak's work was described purely in inferential terms, but one of its original purposes in (Charniak 1972) was to resolve pronoun reference in children's stories. Both are "gap filling" approaches, as contrasted with, say, Rieger's (1974) approach where inferences are made without, as here, continually looking for confirmation of them at every stage. Again, both Charniak's and Wilks' systems have rules equivalent to

A IMPLIES B

A IMPLIES NOT B

where an inference rule seeks either an assertion or its negation. In (3) above, the same matching would have taken place if we had been told that John did not like it. These approaches are therefore far more dedicated to relevance and cohesion than to logical form, for the A/B inference pair above would render any system logically trivial, but from a language understanding point of view, no worse for it!

The chief drawback of Wilks' system is that codings consisting entirely of primitives have a considerable amount of both vagueness and redundancy. For example, no reasonable coding in terms of structured primitives could be expected to distinguish, say, "hammer" and "mallet". That may not matter provided the codings can distinguish importantly different senses of words. Again, a template for the sentence "The shepherd tended his flock" would contain considerable repetition, each node of the template trying, as it were, to tell the whole story by itself. Whether or not such a system can remain stable with a considerable vocabulary, of say several thousand words, has yet to be tested.

Again, it remains to be shown that a large body of common sense inference rules can be controlled, and that solutions can be found to pronoun problems without the chains becoming inordinately long. There may also be something inherently implausible about always preferring a rule chain of length, say, 4 to one of length 3. (Though preferring length 4 to length 1 might be clearer and more plausible.) This difficulty may mean that preferences in the system will have to be weighted, in a way they are not at present.

Riesbeck's Implementation of Schank's System

Schank's system of conceptual representation was introduced in Chapters 4 and 7. It was implemented in the MARGIE system (Schank et al. 1973) with a parser by Riesbeck (1974), an inferential memory by Rieger (1974, and see Chapter 8) and a generation system by Goldman (1974). It is with the parser for English that we will be concerned here. The construction of a parser for conceptual dependency notation was certainly no trivial task: the reader will remember from Chapter 6 that, although Schank gives procedures for extending the conceptualizations — case inferences, for example — he does not, in any sense, give procedures for attaching the conceptual structures to natural language, nor does he consider it his task to provide the information to enable this to be done. That does not mean that he ignores problems like word-sense and case ambiguity in sentences, only that he assumes that the application of a correct representation will, necessarily, solve these problems. As we saw in the last section, on Wilks, the semantic information and criteria required to parse case structures, for example, are considerable and Riesbeck has attempted also to provide these. His first step in making Schank's system computable was, of course, to convert the Schank diagrams to explicit linear symbolic form, since diagrams necessarily contain their information in an implicit form.

Riesbeck's parser is, in the terms developed earlier, depth-first. It becomes top-down almost immediately in the parsing process. Not immediately, because the conceptual structures to be parsed onto the text are "cued in" by the appearance of the words of the sentence, examined from left to right. Riesbeck describes these structures as expectations: expectations, that is, about what else will be said, or has already been said, in the text; and these expectations are concretely expressed as requests associated with the appearance of text words. The strongest argument for this approach is that the most appropriate conceptual structures for what appears in a text are utterly unlike the superficial structure, and that these most appropriate structures are best applied as soon as possible in the parsing — rather than, say, after a later process of deepening the representation as in the Wilks system. The best example for showing this is Riesbeck's "John gave Mary a beating" which, of course, has little to do with giving, and we shall work through this example in detail.

In Riesbeck's system the proper structure for this sentence is actually attached to "beating", so that when "beating" is reached, in going through the sentence, that proper conceptualization (based on the primitive action PROPEL, and expressing the movement of John's hand) will, as it were, take over and supplant the request structures already satisfied during the analysis of the first four words of the sentence.

Below is Riesbeck's sequential table of requests associated with the sentence words, as it is parsed from left to right, and expressed here as sequence down the page.

NP here is the familiar grammar category noun-phrase. The table is quite easy to follow if we remember that the numbered requests in the third column remain active (and are thus passed down the third column) if they are not satisfied. The requests 5 and 6 called "true" are simply default requests (of the type found in LISP, see Chapter 12) that will be satisfied automatically if nothing above them is.

What happens with the parsing of this sentence, put very roughly, is that the satisfaction of request 2 makes the parser think, as it were, that it is dealing with a sentence like "John gave Mary a dollar"; hence its expectation about "to" at step 2. It goes on believing this until requests 3 and 4 are reinstated at step 7 and request 4 is satisfied and not request 3, as would have been the case if Mary had been given a dollar.

The action taken at step 7 is to completely restructure the representation and to base it, not on the Schankian primitive act PTRANS (for "give"), but on PROPEL (for "beat").

STEP	WORD READ	REQUESTS WAITING	REQUESTS TRIGGERED	ACTIONS TAKEN
Ø	none	1 - is there an NP?	none	none
1	John	1 - is there an NP?	1	assume "John" is subject of the verb to follow
2	gave	2 - is the current NP a human? 3 - is the current NP an object? 4 - is the current NP an action? 5 - true	5	assume the word "to", if it appears, introduces the recipient of the "giving"
3	Mary	2 - is the current NP a human? 3 - is the current NP an object? 4 - is the current NP an action?	2	assume Mary is the recipient of the "giving"
4	a	3 - is the current NP an object? 4 - is the current NP an action? 6 - true	6	save the current list of requests and replace it with: 7 - does the current word end an NP?
5	beating	7 - does the current word end an NP?	none	none
6	period	7 - does the current word end an NP?	7	build the NP "a beating" and reset the list of requests
7	none	3 - is the current NP an object? 4 - is the current NP an action?	4	assume the NP action is the main action of the clause, the subject (John) is the actor and the recipient (Mary) is the object

This same method is applied to more complex sentences, such as those containing prepositional phrases.

Let us look at the parsing of a prepositional phrase as in "John prevented Mary from leaving by locking her in".

There are three phases in the parsing:

(i) "prevent" generates requests (expectations) for prepositions FROM and BY1. Let me call these objects preposition-forms for the moment. BY1 is what we can think of as a "sense" of the preposition "by". If Riesbeck were using case description, we could say that BY1 is the "manner case" manifestation of "by", as opposed to its other case manifestations.

(ii) FROM itself has very complex requests associated with it such as FROMØ:

```
(T(IMBED ⟸ ((MOD QUOTE((CANNOT)))
    (TIME CHOICE ⟸ TIME)
    (SUBJ CHOICE OBJ))
  ((OR(EQ WORD(QUOTE BY))
    (BREAK_POINT))
 (RESET_ALL)NIL))
 NIL
```

This is clearly a complex object and need not be fully explained to define its role, namely that of tying together two conceptualizations: one for the preventing and one for the leaving, and in that sense its role is analogous to the Wilks paraplates described earlier. It is written out here so that we can see that this request is seeking the preposition "by" to follow it (fourth line from bottom).

(iii) the requests of BY1 are an even more formidable object than that above, but Riesbeck makes clear that it is seeking forms of primitive act to fit a following "by" phrase should it occur, as it does in fact in the example sentence.

One immediate problem here is what is a prepositional-form like FROMØ? Is it a case or not? At one point Riesbeck suggests that it is 'specific to "prevent"' which suggests that it cannot be a case in any general sense, but is some other classification of preposition appearance. Secondly, it is not really made clear what happens if an expectation is not fulfilled, because they always seem to be. So it is never clear where, say, other FROMn's would come into play if FROMØ failed to do the trick.

This last point is part of a general question as to whether there is anything corresponding to Wilks' paraplate preference order in Riesbeck or not. Wilks has argued, perhaps not convincingly, that there has to be some order imposed in case parsing, but Riesbeck says specifically that the requests are not ordered. On the other hand, Riesbeck's strategy as described may imply an ordering: "prevent" calls FROMØ and if that fails and there is a FROM1, say, available, then there is some ordering like that of application of the paraplates. But that might seem inconsistent with his claim that the requests are NOT ordered.

Riesbeck may unconsciously accept a form of what one might call the phenomenological fallacy, which is that people say what is expected of them, and so a preference order of structures is not necessary. But, alas, they all too often don't, and one has actually to attend to what they say.

Again, depending on the exact interpretation of prepositional-forms, the system is highly superficial in a strict sense of the word: it is its verbs that seek prepositions (if FROMØ is a preposition), rather than basic actions seeking cases. This surface method naturally makes it hard to state "significant semantic generalizations", and is particularly odd in a parser that claims to be based on Schank who abhors all processing based on surface correlations. So, in that sense, Riesbeck's parser may be more independent of Schank's system than appears initially.

Another general, and related, problem is that Riesbeck wishes to avoid any intermediate structure (such as Wilks' templates), whether syntactic or semantic, between text words and the final conceptualization. To do this he may, in the end, have to associate different requests with a word corresponding not only to different senses of the word, but to its syntactic positions. Riesbeck's request table shows how closely related requests are to syntactic order. That need not be fatal, even if so, but it would clearly make it hard to state any semantic generalizations that should be the heart of the system, and very hard to specify the word-by-word requests for a sentence of any length or complexity.

Part of this comes from general difficulties about uncontrolled expectation as a parsing mechanism, and the question of whether one can have a wholly depth-first parser, like Riesbeck's, if one has no back up. One might say that depth-first parsing, whether semantic or syntactic, requires a notion of failure and backup: that is to say, a notion of when a current structure has become inapplicable, and then of where, in the previous choices taken, to go back and try again. But almost all contemporary approaches are weak on failure and backup: Winograd's, Wilks' and Riesbeck's. Indeed, part of the point of top-down, frames-like, approach is to make backup much less needed. The tension comes in trying to combine this with a wholly depth-first parsing technique as Riesbeck has done. Or, to put this in slogan form, structures of expectation may have to be explored more than one at a time so as to simulate attention. Systems that do not attend are like people that do not attend but only expect what to hear. In neither case do they communicate or understand.

Exaggerated expectation involves what was earlier called the phenomenological fallacy: when understanding language as human beings we are never conscious of alternative interpretations, the fact that a word we read in context has many senses out of context, etc., therefore a semantic parser should not consider such alternatives either, for, if it has the right conceptual/semantic/preferential/frame structures in it, it will go directly in a depth-first manner to the correct reading and never consider any other.

The premise of this position is of course true, but the conclusion is totally false, and it is perhaps worth setting out why.

Where it is wrong is in its assumption that the correct interpretation fits and the other possible interpretations do not fit at all. Hence the first path that can be followed will be the right one. The truth of the matter in semantics is that the right interpretation fits better than the others, but to see that we necessarily have to see how well the other possibilities fit.

One detailed example Riesbeck discusses is the parsing of:

"John gave Mary the book"

in which the action "give" generates requests, one of which is that "if what follows the verb immediately is human then assume it is RECIPient case, otherwise assume it is OBJect case." There is nothing in his description that allows any backup to an alternative request should one fail after appearing to succeed and this is consistent with the depth-first no backup position. So, for example, both the following perfectly natural sentences would be wrongly analyzed on such a method:

"John gave Mary to the Imam of Oudh"

"John gave his city his stamp collection"

There is nothing the least tricky or bizarre about such sentences, and the examples show that simple unfettered expectation is not enough unless one can be sure one has one's criteria right, or one has some breadth-first way of considering alternatives or one has complex backup.

PARSING AND "LARGER CHUNKS OF KNOWLEDGE"

In Chapter 8, Charniak introduced a supermarket frame as a representation of a "larger chunk" of our knowledge about the world than the knowledge representations discussed so far in this chapter, even though the three authors surveyed here were described earlier as being within the frames approach too. What we have already then are two "scales" of frame: a smaller and a larger. Wilks' template structures and Schank's conceptualizations would be frame structures on the smaller but not the larger scale, whereas Charniak's frames, as described in Chapter 8, would be larger scale frames. Schank has recently made a transition to the larger scale with structures he calls (after Abelson 1973) "scripts". These will now be described briefly, and their possible relation to parsing will be discussed later.

Schank began the transition, together with Rieger, by developing a new class of causal inferences that deepen the conceptualization diagrams still further. So, in the analysis of "John's cold improved because I gave him an apple" (Schank 1973a) the extended diagram contains at least four yet lower levels of causal arrowing, including one corresponding to the notion of John constructing the idea (MBUILD) that he wants to eat an apple. So we can see that the underlying explication of meaning here is not only in the sense of linguistic primitives, but in terms of a theory of mental acts as well.

A contrast became apparent here between Schank's work and Rieger's: in Rieger's thesis (Rieger 1974) the rules of inference create separate and new subgraphs (see Chapter 8) which may stand in an inferential relation to each other so as to produce conclusions about problems of, say, pronoun reference, etc. But in Schank's corresponding papers the same inferences are not applied to actual problems of language analysis (Schank 1973a) but they only produce more complex conceptual graphs. Then Schank reversed this trend towards ever more complex diagrams (for individual sentences) by considering the representation of text. He concluded that the representations of parts of the text must themselves be interconnected by causal arrows, and that, in order to preserve lucidity, the conceptual diagrams for individual sentences and their parts must be abbreviated, as by triples such as PEOPLE PTRANS PEOPLE. Here, some of the surface structure has to survive in the representation unless one is prepared to commit oneself to the extreme view that the ordering of sentences in a text is a purely superficial and arbitrary matter. The sense in which this is a welcome reversal of a trend

should be clear, because in the "causation inference" development all the consequences and effects of a conceptualization had to be drawn within itself. Thus, in the extreme case, each sentence of a text should have been represented by a diagram containing most or all of the text of which it was a part. Thus the representation of a text would have been impossible on such principles.

Most recently (1975a) Schank has produced a "script" for knowledge about eating in a restaurant. This is no longer a causal sequence developed during the processing of a text, but is, like Charniak's frames (Chapter 8) a structure which precedes any encounter with text. The script begins:

"Scene 1 entering

PTRANS - go into restaurant

MBUILD - find table

PTRANS - go to table

MOVE - sit down

Scene 2 ordering

ATRANS - receive menu

ATTEND - look at it

MBUILD - decide on order

MTRANS - tell order to waitress."

and so on for scenes 3, eating, and 4, exiting. Schank also has a program which will take a paragraph length restaurant story and produce a slightly longer story with the "missing bits" filled in from the script above. So that, if it reads a story like "John went to the restaurant, got angry and left", it can reply with what it assumes to have happened in more detail, such as "John went to the restaurant. He found a table and sat down. He got the menu and ordered the food. When it came he didn't like it, so he got angry with the waitress and left."

As so often in our subject it is hard to see just what claims are being made by such a program, or how to evaluate such output. The sort of consideration that makes it more plausible can be seen from the following genuine newspaper story:

> "Mr. Justice Forbes, at Winchester Crown Court yesterday, criticized authorities who charge people with attempted murder without "a scrap of evidence".
>
> Philip Slater, aged 25, was charged with carrying an offensive weapon, attempted murder, and wounding with intent.
>
> After he had pleaded guilty to wounding with intent Mr. Justice Forbes said: "Why was this charge of attempted murder put? There was never the slightest chance that any jury would convict this man on this charge."

The reader should now ask himself whether the man charged was carrying a gun. If the answer is yes, the question arises as to why one thinks so, because the story does not say so. Schank might well argue that we are filling out the story from a "script" of some sort.

Exactly the same considerations are present in Charniak's frames, discussed in Chapter 8, and these two systems with "large scale frames" will be discussed in the next part of the chapter.

We shall now turn to some comparisons and contrasts, under ten connected headings, between the systems discussed in the last four chapters from the point of view of parsing.

Level of Representation

One important line of current dispute among recent approaches concerns the appropriate level of representation for natural language. On the one hand are those like Colby, and in some ways Charniak, who hold that language can be, in effect, self-representing, while on the other hand there are those like Schank and Wilks who hold that the appropriate level of computation for inferences about natural language is in some reduced, or primitive, representation. Charniak holds that his structures are independent of any particular level of representation, or rather, that they could be realized at a number of levels of representation, depending on the subject area. However, there is no doubt that the representation in terms of predicates that he offers in his work appears to be in one-to-one correspondence with English words.

The strongest low-level approach is undoubtedly that of Colby, who straightforwardly faces the enormous mapping problems involved if the structures are at the English word level. It is important to realize that this dispute is ultimately one of degree, since no one would claim that every locution recognized by an intelligent analyzer must be mapped into a "deep" representation. To take an extreme case, any system that mapped "Good Morning" into a deep semantic representation before deciding that the correct response was also "Good Morning" would be making a serious theoretical mistake.

But these disputes are at all times questions of degree: Schank has argued most strongly for the use of primitives but has only produced them for verbs, although work has continued independently on noun primitives for his system (Weber 1972). While on the other hand Wilks has produced primitives for all parts of speech, yet retains access to surface items in his system, unlike Schank, and now (1975c) seems in the process of putting other, less primitive, more superficial, items into his semantic formulas.

However, the most serious argument for a non-superficial representation is not in terms of the avoidance of mapping difficulties, but in terms of theoretical perspicuity of the primitive structures, and this argument is closely tied to the defence of semantic primitives in general, which is a large subject that can only briefly be touched on here. One of the troubles about semantic primitives is that they are open to bad defenses, which decrease rather than increase their plausibility. For example, some users of them for linguistic representation have declared them to have some sort of objective existence, and have implied that there is a "right set" of primitives open to empirical discovery. On that view the essentially linguistic character of structures of primitives is lost, because it is an essential feature of a language that we can change its vocabulary, and be understood via alternative vocabularies. If there was a right primitive vocabulary, that was a list of names of brain-items, that essential feature would be lost. What is the case is that there is a considerable amount of psychological evidence that people are not able to recall either the actual words or the syntactic structure used. There is large literature on this subject, from which two sample references would be (Wettler 1973) and (Johnson-Laird 1974).

These results are, of course, no proof of the existence of semantic primitives but they are undoubtedly supporting evidence of their plausibility, as is, on a different plane, the result from the encoding of the whole Webster's Third International Dictionary at Systems Development Corporation. It was found that a rank-ordered frequency count of the words used to define other words in that vast dictionary was a list (omitting "the" and "a") which corresponded almost item-for-item to the sort of list of semantic primitives used by, and derived a priori by of course, those researchers described here who construct semantic representations.

It is important to distinguish the dispute about level from the, closely connected, topic of the centrality of the knowledge required by a language understanding system.

Centrality

The centrality of certain kinds of information concerns not its level of representation but its non-specificity: again a contrast can be drawn between the sorts of information required by Charniak's system, on the one hand, and that required by Schank's and Wilks', on the other. Charniak's examples suggest that the fundamental form of information is highly specific to particular situations, like parties and the giving of presents, while the sorts of information central to Schank's (before scripts) and Wilks' systems are partial assertions about human wants, expectations, and so on, many of which are so general as to be almost vacuous which, one might argue, is why their role in understanding has been ignored for so long.

If one was a reasonably fluent speaker of, say, German, one might well not understand a German conversation about birthday presents unless one had detailed factual information about how Germans organize the giving of presents, which might be considerably different from the way we do it. Conversely, of course, one might understand much of a technical article about a subject in which one was an expert, even though knowing very little of the language in which it was written.

In the end this difference may again turn out to be one of emphasis, and of what is most appropriate to different subject areas, though there may be a very general issue lurking somewhere here. It may not be a foolish question to ask whether much of what appears to be about natural language in AI research is in fact about language at all. Even if it is not that may in no way detract from its value. Newell (Moore, Newell 1973) has argued that AI work is in fact "theoretical psychology", in which case it could hardly be research on natural language. When describing Winograd's work earlier in the chapter, this question was raised in a weak form by asking whether his definition of "pickup" had anything to do with the natural language use of the word, or whether it was rather a description of how his system picked something up, a quite different matter.

Suppose we generalize this query somewhat, by asking the apparently absurd question of what would be wrong with calling, say, Charniak's work an essay on the "Socio-Economic Behavior of American Children under Stress?" In the case of Charniak's work this is a facetious question, asked only in order to make a point, but with an increasing number of systems in AI being designed not essentially to do research on natural language, but in order to have a natural language "front end" to a system that is essentially intended to predict chemical spectra, or play snakes and ladders or whatever, the question becomes a serious one. It seems to be a good time to ask whether we should expect advance in understanding natural language from those tackling the problems head on, or those concerned to build a "front end". It is clearly the case that any piece of knowledge whatever could be essential to the understanding of some story. The question is, does it follow that the specification, organization and formulization of that knowledge is the study of language, because if it is then all human enquiry from physics and history to medicine is a linguistic enterprise. And, of course, that possibility has actually been entertained within cert. strains of modern philosophy.

This is not an attempt to breathe fresh life into a philosophical distinction between being about language and not being about language (though see Chapter 11 below), but rather introducing a practical distinction between specific knowledge and central knowledge without which a system could not be said to understand language at all. For example, one might know nothing of the arrangement of American birthday parties, but could not be accused of not understanding English even though one failed to understand some particular children's story. Yet, if one did not have available some very general partial inference such as the one about people being hurt and falling, or one about people endeavoring to possess things that they want, then it is quite possible that one's lack of understanding of quite simple sentences would cause observers to think that one did not understand English. An interesting and difficult question that then arises is whether those who concentrate on central and less central areas of discourse could, in principle, weld their bodies of inferences together in such a way as to create a wider system; whether, to put the matter another way, natural language is a whole that can be built up from parts?

On the other hand, it must be realized that there are undoubtedly linguistic problems that do seem to rest on even more specific real world knowledge than that represented in current systems, as can be seen by contrasting the sentences:

The deer came out of the wood

The grub came out of the wood

where we might safely assume that readers would assign quite different senses to "wood" in the two cases simply on the basis of the two different agents, though no one has yet suggested any general method for tackling such elementary examples.

Phenomenological Level

Another distinction that can be confused with the central-specific one is that of the "phenomenological levels" of inferences in an understanding system. There is nothing daunting in that phrase: consider the action eating which is, as a matter of anatomical fact, quite often an act of bringing the bones of one's ulna and radius (in the arm) close to that of the lower mandible (or jaw). Yet clearly, any system of common sense inferences that considered such a truth when reasoning about eating would be making a mistake. One might say that the phenomenological level of the analysis was wrong even though all the inferences it made were true ones. The same would be true of any AI system that made everyday inferences about physical objects by considering their quantum structure.

Schank's analysis of eating contains the information that it is done by moving the hands to the mouth, and it might be argued that even this is going too far from the "meaning" of eating, whatever that may be, towards generally true information about the act which, if always inferred about all acts of eating, will carry the system unmanageably far.

There is no denying that this sort of information might be useful to have around somewhere; that, in Minsky's terms, the "default" value of the instrument for eating is the hand brought to the mouth, so that, if we have no contrary information, then that is the way to assume that any given act of eating was performed. Nonetheless, there clearly is a danger, and that is all that is being

highlighted here, of taking inferences to a phenomenological level beyond that of common sense. A clearer case would be Schank's analysis (1973a) of mental activity in which all actions, such as kicking a ball, say, are preceded by a mental action of conceiving or deciding to kick a ball. This is clearly a level of analysis untrue to common sense, and which could have only harmful effects in a system intended to mimic common sense reasoning and understanding.

Decoupling

Another general issue in dispute concerns decoupling, which is whether or not the actual parsing of text or dialogue into an "understanding system" is essential. Charniak and Minsky believe that this initial "parsing" can be effectively decoupled from the interesting inferential work and simply assumed. But one could argue that many of the later inferences would actually have to be done already, in order to have achieved the initial parsing. For example, in analyzing "He shot her with a colt", we cannot ascribe any structure at all until we can make the inference that guns rather than horses are instruments for shooting, and so such a sentence cannot be inserted into an inference-but-no-parsing structure without assuming that language does not have one of its essential characteristics, namely systematic ambiguity. The essence of decoupling is allowing representational structures to have significance quite independent of their application, and if one does decouple one may be in a situation not essentially different from that of the logician who simply asserts that such-and-such is the "right structure" of some sentence.

Also, the inferences required to resolve word sense ambiguities, and those required to resolve pronoun reference problems, are not of different types; often the two problems occur in a simple sentence and must be resolved together. But Charniak's decoupling has the effect of completely separating these two closely related linguistic phenomena in what might seem to be an unrealistic manner. His system does inferencing to resolve pronoun ambiguities, while sense ambiguity is presumably to be done in the future by some other, ultimately recoupled, system. (Although Charniak would argue that sense ambiguity could be introduced into his system in its present form.)

Another way of pointing up the difference between the attitudes of the systems discussed to decoupling is by describing the role of syntax analysis in them. As we saw, syntax was the heart of Winograd's system, but both scales of frame approach discount syntax analysis, though for very different reasons: Charniak does so because it is part of the initial parsing from which his inferential work has been decoupled. Schank and Wilks do so because they believe a semantic analysis to be fundamental, and that in an actual implementation the results of syntactic analysis can all be achieved by a sufficiently powerful semantic analyzer. And this last assumption is confirmed by the limited degree of success that the two semantic analyzers have actually achieved in operation.

The thesis behind this chapter (and it differs from the views of Chapters 7 and 8) is that parsing is essential to a system, and so it cannot be ultimately decoupled. This goes not only for "representation of knowledge" systems, but equally for those claiming to represent semantic memory. The argument is not only that parsing provides a test of a proposed structure, for that is secondary, but that the parsing procedures define what the significance of the proposed structure is. It will not have escaped the notice of readers that all the knowledge and memory structures proposed and discussed in this section of the book look very like rearranged English. Schank and Wilks claim, in different ways, that the structures are only short-hand for others expressed in terms of primitives, but those in turn look like rearranged, but more bizarre, English. The argument of this chapter is that it is only parsing — the procedural application of these structures to surface text — that gives them significance and thus stops them being just rearranged English.

Application

This point is a generalization of the last two, and concerns the way in which different systems display, in the structures they manipulate, the actual procedures of application of those structures to input text or dialogue. This is a matter different from computer implementation of the system. In the case of Colby's patterns, for example the form of their application to the input English is clear, even though the matching involved could be achieved by many different implementation algorithms. In Wilks' system the same is true of the template structures, even though by the time the input has reached the canonical template form it is considerably different from the input surface structure. The system at the extreme end of any scale of perspicuity of application is Winograd's, where the procedural notation, by its nature, tries to make clear the way in which the structures are applied. At the other end are the systems of Schank and Charniak, where no application is specified, which means that the representations are not only compatible with many implementation algorithms, which does not matter, but are also compatible with many systems of linguistic rules, whose specification is an essential piece of inquiry, and whose subsequent production may cause the basic system to be fundamentally different.

Application is thus different from decoupling, for Schank's system is clearly coupled to language text by Riesbeck's parser, though his structures do not express their own application to language text, as was pointed out in the discussion of Riesbeck's system.

In some of his more recent writings Winograd has begun to develop a view that is considerably stronger than this "application" one: in his view the control structure of an understanding program is itself of theoretical significance, for only in that way, he believes, can natural language programs of great size and complexity remain perspicuous.

Forward Inference

Another outstanding dispute concerns whether one should make massive forward inferences as one goes through a text, keeping all one's expectations intact, as Charniak, Schank and Rieger hold, or whether, as Wilks argues, one should adopt some "laziness hypothesis" about understanding, and generate deeper inferences only when the system is unable to solve, say, a referential problem by more superficial methods. Or, in computer terms, should a system be problem or data-driven?

Although Schank sometimes writes of a system making "all possible" inferences as it proceeds through a text, this is not in fact the heart of the dispute, since no one would want to defend any strong definition of the term "all possible inferences". Charniak's argument is that, unless certain forward inferences are made during an analysis of, say, a story — forward inferences, that is, that are not problem-driven; not made in response to any particular problem of analysis then known to the system — then, as a matter of empirical fact, the system will not in general be able to solve ambiguity or reference problems that arise later, because it will never in fact be possible to locate (while looking backwards at the text, as it were) the points where those forward inferences ought to have been made. This is, in very crude summary (see Chapter 8), Charniak's case against a purely problem-driven inferencer in a natural language understander.

One difficulty with this argument is the location of an example of text that confirms the point in a non-contentious manner. Another is that it is not always clear whether the argument is about what people are thought to do when they understand, or about how one should construct an understanding system. It would be possible, for example, to agree with Charniak's argument and still construct a purely problem-driven inferencer on the ground that, at the moment, this is the only way one can cope with the vast majority of inferences for understanding, since any system of inferences made in response to no particular problem in the text is too hard to control in practice. Rieger (1974), for example, has to resort to a simple arbitrary numerical cut-off of forward inferencing, which seems unsatisfactory. Indeed, it is noticeable that the more recent papers of Schank and Charniak have been considerably less forward-inference oriented than earlier ones. On the other hand, systems that now run without forward inferencing could easily do some: the process Wilks calls extraction for example could easily run as a purely data-driven process.

This dispute is perhaps only one of degree, and about the possibility of defining a degree of forward inference that aids the solution of later semantic problems without going into unnecessary depth. This might be an area where psychological investigations would be of enormous help to workers in AI.

Modularity

Modularity concerns the decomposability of a program or system into (interacting) parts, and the nature of the relationship between the parts. Winograd's program, as we saw, contains syntactic, semantic and deductive segments, which interact in a way he describes as "heterarchic", as opposed to "hierarchic", which means that different segments can be in control at different times.

On the other hand, Schank and Wilks have argued that it is not necessary to observe either the syntactic-semantic, or the semantic-deductive, distinction in an understanding program, and that it may well not be possible to specify the syntactic-semantic distinction in principle either. On this view there is no particular virtue in integrating syntax and semantic routines, since there was no need to separate them.

Charniak, however, would argue that, in some sense, one should make a syntax-semantics distinction here if one can (see Chapter 2). This is consistent with his view on decoupling, and it is convenient to decouple at a module, as it were, such as syntactic analysis; but decoupling and strong modularity are not the same thing. Winograd's program, for example, is modular but not at all decoupled from surface text.

Scale of Representations

This dispute is related to the ones about application and decoupling and concerns the practical function of the larger scale frame representations of knowledge. The point of view of this chapter has been that these representations must be justified in terms of some concrete problem that they solve, such as word-sense or pronoun reference ambiguity, otherwise they may not be essentially connected with the understanding of natural language. That is to say that representation of knowledge as such, and independent of any purpose or problem, is not a task for AI.

In the light of this let us turn back for a moment to the "larger scale" chunks of knowledge: Charniak's frames, which were created as wholes (Chapter 8) or Schank's scripts which were created by "stacking up", in a predetermined order, primitives for individual actions. This development is open to any representation system with forms standing for individual assertions: Wilks, for example, could "stack up" his common sense inferences rules in advance so that they ran not just

P IMPLIES Q

but

P

IMPLIES Q

IMPLIES R

IMPLIES S

where each of the schematic letters stands for a complex object like a template, or one half of a common sense inference rule.

One way of putting a doubt about the role of any structures of so large a scale is to imagine one for a less everyday activity.

Suppose we were considering sentences about a puberty rite in a distant tribe. We might make up a frame as follows in a combination of Schank's and Charniak's (Chapter 8) notation:

Script: male puberty rite

Roles: male child, village elder, helpers, crowd

Reason: placing ritual incisions on back of child

(11) a) Goal: CHILD is tattooed

b) HELPERS hold CHILD (by both arms)

c) ELDER obtain TOOLS

d) ELDER exhorts CROWD (on proper behavior)

e) (general condition)

§bad behavior by CROWD ⟶ activity halted

f) ELDER checks if CHILD properly purified

g) (special condition)

CHILD not purified ⟶ activity halted

h) ELDER mark CHILD's back

i) | method suggested

└⟶ do for all CUT-MARKS

The general idea should be clear without detailed explanation, and the choice of a remote, and imaginary, culture is for a good reason.

Suppose we now have the "story sentence"

(12) Little Kimathi's mother accidentally touched his arm during the puberty rite. The crowd drew back in horror.

If we wish to "understand" this sentence, would we need the "puberty rite frame" given earlier? The frame (11) covers (12) with line (e), given some adequate definition list of bad behaviors in the frame (to which we are directed by "§", let us imagine). And yet it is clear that we do understand (12) perfectly well without (11).

In common sense terms, we could say that we infer from (12) that the mother touching Kimathi during the ceremony was a bad thing; we do not need the frame stating that in order to understand. An earth-frameless Martian would understand (12) provided he knew the meanings of the words and had some smaller scale knowledge about aversion behavior being produced by bad events. Hence the choice of a puberty rite rather than a supermarket for the frame (11), for we are Martians where puberty rites are concerned. If we do understand (12) it cannot be from our associated frame because, presumably, we do not have one.

It is not being argued here that large scale frames have no function, only that, as regards concrete problems of language understanding, their function has not yet been made explicit.

One possible explicit claim about the function of large scale frames (and one almost explicit in (Schank 1975a)) could be put as follows: "In order to understand a story we need to know how basic stories (i.e. frames) of that type go; that is, we only understand, and can only understand, a particular story by judging how it follows, or diverges from, the normal story of that type." We could call this the plot-line hypothesis (PLH) and whether current psychological research is to be taken as confirming or disconfirming this claim is at present a matter of dispute.

Real World Procedures

At the time of Winograd's work and during the early phases of the design of PLANNER (see Chapter 1) there was considerable dispute about the slogan "meanings are procedures". The dispute was about the most appropriate form of intelligent programs and the slogan was intended to support the procedural form, which we saw examples of from Winograd.

This dispute has quietened down, but there is an analogue of it still about in language work, and concerned not with the internal procedures of the computer, but the relation of understanding to real activities in the world. The reader will have noticed that many of the systems described in this book (Winograd's, Charniak's, Scragg's, Rieger (1975)) actually concern themselves with real activities: picking up, cooking, etc., or, more precisely, with simulations of these physical activities. Is this just accident, or is there some real relationship between doing and understanding?

There is an implicit, but pervasive, hypothesis behing much current work that might be called the do-it hypothesis, namely that in representing knowledge, we should concentrate on the representation of human activities that we know how to perform: stacking blocks, eating in restaurants, shopping in supermarkets, cooking food, or, in Rieger's case (1975), the different matter of how a water closet performs its task.

The connexion with natural language comes with the further assumption that one cannot understand language about human activities unless one has performed, or can perform, them. The core of this argument is rather like the one that young school children in class cannot understand the discussions of love in the Shakespeare plays they have to study because they have not experienced the corresponding feeling.

There is clearly something in this argument, but the trouble comes with specifying just what it is that the non-performer does not understand, given that one could answer questions as if one did understand. People can answer a lot of questions about skiing, and certainly appear to understand newspaper stories about it, even when they cannot ski.

A stronger argument is that the relevance of an activity frame to natural language is bound up with the teachability of the activity. This version has the advantages of avoiding asserting the PLH directly, as Schank seems to do, but is weakened by the fact that the frame activities in question (eating in restaurants, shopping in supermarkets) are not, in any obvious sense, taught at all, at least not in the sense that using scissors or typing one's shoe laces are taught.

These are points in dispute, then, between those, like Charniak and Schank, who believe in the function of large scale frame structures, and those who, like Wilks, remain agnostic.

The Justification of Systems

Finally, one might usefully, though briefly, contrast the different modes of justification implicitly appealed to by the systems described earlier in this paper. These seem to reduce to four:

(i) In terms of the power of the inferential system employed. This form of justification has underlain the early predicate calculus-based language programs, and is behind Hayes' (1974) recent demand that any formalism for natural language analysis should admit of a set theoretic semantics (see Chapter 11), so as to gain "intellectual respectability", as he puts it. The same general type of justification is appealed to in some degree by systems with PLANNER-type formalisms.

(ii) In terms of the provision and formalization in any terms, including English, of the sorts of knowledge required to understand areas of discourse.

(iii) In terms of the actual performance of a system, implemented on a computer, at a task agreed to demonstrate understanding.

(iv) In terms of the linguistic and/or psychological plausibility of the proffered system of representation.

Oversimplifying considerably, one might say that Charniak's system appeals mostly to (ii) and somewhat to (i) and (iv); Winograd's to (iii) and somewhat to the other three categories; Colby's (as regards its natural language, rather than psychiatric, aspects) appeals almost entirely to (iii); and Schank's and Wilks' to differing mixtures of (ii), (iii) and (iv).

In the end, of course, only (iii) counts for empiricists, but there is considerable difficulty in getting all parties to agree to the terms of a test. A cynic might say that, in the end, all these systems analyze the sentences that they analyze or, to put the same point a little more theoretically, there is a sense in which systems, those described here and those elsewhere, each define a natural language, namely the one to which it applies. The difficult question is the extent to which those many and small natural languages resemble English.

E. Charniak and Y. Wilks (eds.), Computational Semantics

PSYCHOLOGY OF LANGUAGE AND MEMORY

Walter F. Bischof

INTRODUCTION

The intent of this chapter is twofold: first it should provide the non-psychologist with some basic concepts and some important experimental findings in the field of psycholinguistics and the psychology of memory. The second goal is to take a close look at the nature of psychological arguments and psychological evidence. To understand this second goal one has to realize that there is a growing interest among the AI community in psychology and that psychological arguments are gaining more and more importance in theoretical considerations. It is undeniable that psychology has had some beneficial influence on AI and that psychological evidence may sometimes be useful for the evaluation of AI models. But a closer look at today's psychology and some thoughts about the nature of psychological arguments will show that this psychological evidence is often due to misconceptions and that in general psychological arguments are often overestimated.

The chapter is roughly organized in the following way: after a discussion of association, a concept equally important both for language and memory, some topics in psycholinguistics are reviewed, followed by a discussion of methodological problems. The second section reviews some theories of memory, again followed by a discussion of more general problems. The last section considers the use of psychological findings and theories in AI.

One preliminary point should be made clear: it is not the intent of this chapter to give an encompassing review of the two fields. A few selected topics