DIAGNOSIS AS A NOTION OF GRAMMAR

Mitchell Marcus
Artificial Intelligence Laboratory
M.I.T.

This paper will sketch an approach to natural language parsing based on a new conception of what makes up a recognition grammar for syntactic analysis and how such a grammar should be structured. This theory of syntactic analysis formalizes a notion very much like the psychologist's notion of "perceptual strategies" [Bever '70] and makes this formalized notion - which will be called the notion of <u>wait-and-see diagnostics</u> - a central and integral part of a theory of what one knows about the structure of language. By recognition grammar, we mean here what a speaker of a language knows about that language that allows him to assign grammatical structure to the word strings that make up utterances in that language.

This theory of grammar is based on the hypothesis that every language user knows as part of his recognition grammar a set of highly specific diagnostics that he uses to decide deterministically what structure to build next at each point in the process of parsing an utterance. By <u>deterministically</u> I mean that once grammatical structure is built, it cannot be discarded in the normal course of the parsing process, i.e. that no "backtracking" can take place unless the sentence is consciously perceived as being a "garden path". This notion of grammar puts knowledge about controlling the parsing process on an equal footing with knowledge about its possible outputs.

To test this theory of grammar, a parser has been implemented that provides a language for writing grammars of this sort, and a grammar for English is currently being written that attempts to capture the wait-and-see diagnostics needed to parse English within the constraints of the theory. The control structure of the parser strongly reflects the assumptions the theory makes about the structure of language, and the discussion below will use the structure of the parser as an example of the implications of this theory for the parsing process. The current grammar of English is deep but not yet broad; this has allowed investigation of the sorts of wait-and-see diagnostics needed to handle complex English constructions without a need to wait until a grammar for the entire range of English constructions could be written. To give some idea of the scope of the grammar, the parser is capable of handling sentences like:

Do all the boys the librarian gave books to want to read them?
The men John wanted to be believed by shot him yesterday.

It should be mentioned that certain grammatical phenomena are not handled at all by the present grammar, chief among them conjunction and certain important sorts of lexical ambiguity. There is every intention, however, of expanding the grammar to deal with them.

<u>Two Paradigms</u>

To explain exactly what the details of this wait-and-see (W&S) paradigm are, it is useful to compare this notion with the current prevailing parsing paradigm, which I will call the <u>guess-and-then-backup</u> (G&B) paradigm. This paradigm is central to the parsers of both Terry Winograd's SHRDLU [Winograd '72] and Bill Woods' LUNAR [Woods '72] systems.

In a parser based on the G&B paradigm, various options are enumerated in the parser's grammar for the next possible constituent at any given point in the parse and these options are tested one at a time against the input. The parser assumes tentatively that one of these options is correct and then proceeds with this option until either the parse is completed or the option fails, at which point the parser simply backs up and tries the next option enumerated in the parser's grammar. This is the paradigm of G&B: enumerate all options, pick one, and then (if it fails) backup and pick another. While attempts have been made to make this backup process clever, especially in Winograd's SHRDLU, it seems that it is very difficult, if not impossible in general, to tell from the nature of the <u>cul de sac</u> exactly where the parser has gone astray. In order to parse a sentence of even moderate complexity, there are not one but many points at which a G&B parser must make guesses about what sort of structure to expect next and at all of these points the correct hypothesis must be found before the parse can be successfully completed. Furthermore, the parser may proceed arbitrarily far ahead on any of these hypotheses before discovering that the hypothesis was incorret, perhaps invalidating several other hypotheses contingent upon the first. In essence, the G&B paradigm considers the grammar of a natural language to be a tree-structured space through which the parser must blindly, though perhaps cleverly, search to find a correct parse.

The W&S paradigm rejects the notion of backup as a standard control mechanism for parsing. At each point in the parsing process, a W&S parser will only build grammatical structure it is sure it can use. The parser does this by determining, by a two part process, which of the hypotheses possible at any given point of the parse is correct before attempting any of them. The parser first recognizes the specific <u>situation</u> it is in, determined both on the basis of global <u>expectations</u> resulting from whatever structure it has parsed and absorbed, and from <u>features</u> of lower level substructures from a little ahead in the input to which internal structure can be assigned with certainty but whose function is as yet undetermined. Each such situation can be so defined that it restrains the set of possible hypotheses to at most two or three. If only one hypothesis is possible, a W&S parser will take it as given, otherwise it will proceed to the second step of the determination process, to do a <u>differential diagnosis</u> to decide between the competing hypotheses. For each different situation, a W&S grammar includes a series of easily computed tests that decides between the competing hypotheses. <u>The key assumption of the W&S paradigm, then, is that the structure of natural language provides enough and the right information to determine exactly what to do next at each point of a parse.</u> There is not sufficient room here to discuss this assumption; the reader is invited to read [Marcus ´74], which discusses this assumption at length.

<u>The Parser Itself</u>

To firm up this talk of "expectations", "situations", and the like, it it useful to see how these notions are realized in the existing W&S parsing system. Before we can do this, it will be necessary to get an overview of the structure and operation of the parser itself.

A grammar in this system is made up of <u>packets</u> of pattern-invoked demons, which will be called <u>modules</u>. (The notion of packet here derives from work by Scott Fahlman [Fahlman ´73].) The parser itself consists of two levels, a <u>group level</u> and a <u>clause level</u>, and any packet of modules is intended to function at one level or the other. Modules at group level are intended to work on a buffer of words and word level structures and to eventually build group level structures, such as <u>Noun Groups</u> (i.e. Noun Phrases up to the head noun) and <u>Verb Groups</u> (i.e. the verb cluster up to the main verb), which are then put onto the end of a buffer of group level structures not yet absorbed by higher level processes. Modules at clause level are intended to work on these substructures and to assemble them into clauses. The group buffer and the word buffer can both grow up to some predetermined length, on the order of 3, 4, or 5 structures. Thus the modules at the level above needn´t immediately use each structure as it comes into the buffer, but rather can let a small number of structures "pile up" and then examine these structures before deciding how to use the first of them. In this sense the modules at each level have a limited, sharply constrained look-ahead ability; they can wait and see what sort of environment surrounds a substructure in the buffer below before deciding what the higher level function of that substructure is. (It should be noted that the amount of look-ahead is constrained not only by maximum buffer length but also by the restriction that a module may access only the two substructures immediately following the one it is currently trying to utilize. This constraint is necessary because the substructure about to be utilized at any moment may not be the first in the buffer, for various reasons.)

Every module consists of a <u>pattern</u>, a <u>pretest procedure</u>, and a <u>body</u> to be executed if the pattern matches and the pretest succeeds. Each pattern consists of an ordered list of sets of <u>features.</u> As structures are built up by the parser, they are labelled with features, where a feature is any property of a structure that the grammar wants to be visible at a glance to any module looking even casually at that structure. (Structures can also have <u>registers</u> attached to them, carrying more specialized sorts of information; the contents of a register are privileged in that a module can access the contents of a register only if it knows the name of that register.) A module´s pattern matches if the feature sets of the pattern are subsumed by the feature sets of consecutive structures in the appropriate buffer, with the match starting at the effective beginning of the buffer.

Very few modules in any W&S grammar ae always active, waiting to be triggered when their patterns match; a module is active only when a packet it is in has been activated, i.e. added to the set of presently active packets. Packets are activated or deactivated by the parser at the specific order of individual modules; any module can add or remove packets from the set of active packets if it has reason to do so.

A priority ordering of modules provides still further control. Every module is assigned a numerical priority, creating a partial ordering on the active modules. At any time, only the highest-prioritied module of those whose patterns match will be allowed to run. Thus, a special purpose module can edge out a general purpose module both of whose patterns match in a given environment, or a module to handle some last-resort case can lurk low in a pool of active modules, to serve as default only if no higher-prioritied module responds to a situation.

<u>Firming Up The Notion Of Situation</u>

This, in brief, is the structure of the W&S parser; now we can turn to a discussion of how this structure reflects the theoretical framework discussed above. Let us begin by recasting a statement made above: In deciding what unique course of action to take at any point in the parse, the parser first recognizes a specific well-defined situation on the basis of a combination of global expectations and the specific features of lower level substructures which are as yet unabsorbed.

It should now become clear that what it means to have a global expectation is that the appropriate packet is active in the parser, and that each module is itself the specialist for the situation that its packet, pattern and pretest define. The grammar activates and deactivates packets to reflect its global expectations about syntactic structures that may be encountered as a result of what it has seen so far. (The parser might also activate packets on the basis of what some higher level process in a natural language understanding system tells it to expect by way of discourse phenomena.) These packets often reflect rather large scale grammatical expectations; for example, the following are some packets within the existing grammar: Simple-sentence start, a packet of modules that parse the subject, main verb, and initial modifiers of major clauses, determining clause type and the like; Simple-sentence after VG, whose modules parse objects and clause level prepositional phrases in top level clauses; WH-question after VG, similar to Simple-sentence after VG, except that its modules are responsible for actively deciding where to "replace" the group fronted to form a WH-question and also where to replace the NP "pulled out" of a relative clause. A different sort of packet is NG-object or To-complement, a packet of modules that includes as subpacket the packet To-complement, whose modules assign deep subjects to to-complements with "deleted" subjects, e.g. "John wanted to buy a bicycle.". Beyond this, NG-object or To-complement includes modules which delay modules in other active "after VG" packets from assigning any NP appearing after the VG as the object of that VG until it can be determined whether or not it is the subject of a following to-complement - Compare "He wanted a free pass to the movie.", "He wanted a friend to see the movie.", "He wanted a friend to see the movie with.". It should be noted that packets of modules, i.e. expectations in our theoretical framework, roughly correspond to states in Woods´ Augmented Transition Network model with some major differences: Not only do packets correspond to much larger grammatical chunks than the typical usage of ATN states, but more importantly, many packets may be active at the same time and the modules in these packets may strongly interact with each other if they so choose.

The other determinant of situation in the theoretical framework is the general features of unabsorbed lower level substructures. It should be obvious that this corresponds to the pattern-match between a module´s pattern and the substructures in the buffer below it. That these lower structures are available at clause level is the result of the interaction between the modules at group level and those at clause level. Again, for a discussion of why such substructures can be built independently, the reader is referred to [Marcus ´74]. It is an important claim of the W&S theory that though these patterns are very simple and - viewed from the level above - very local, they suffice, in conjunction with expectations reflected by packet activations, to so restrict the range of possible options open to the parser at any point that a deterministic diagnosis of those options can be made. While a module´s pretest might utilize very complex tests that strongly aid in the restriction of situation, in fact pretests seem to be needed mainly to compute simple boolean functions of features tha are more complex than the implicit logical conjunction demanded by the patterns themselves.

Differentiating Between Hypotheses

Now that it is clear how a situation is defined, i.e. how a given module becomes active, we must consider how the correct course of action can be determined given the set of possible alternatives defined by a situation, i.e. what a module knows that makes it a diagnostic specialist. We need to investigate in particular what sorts of tests a module can use to dependably yield a differential diagnosis for the situation it is a specialist for.

A module has available several different sorts of information for diagnosis; it can use the syntactic information it can ferret out directly from the structures it has built, and it can ask questions of both a crude but fast semantic model and a full world model. To gather syntactic information the parser has a facility for investigating any node of any tree structure it knows about; thus it can investigate the features or registers of any node it pleases. While the parser itself attempts to build an annotated surface structure similar to that built by Winograd´s SHRDLU, it converses constantly with a case frame interpreter that is intended to serve as interface between it and deep world modelling. The parser is obliged to inform the case frame interpreter as soon as it can of information such as what the main verb of a given clause is, what its subject is, what its objects and prepositional phrases are. (The case frame interpreter speaks the parser´s language; it knows how to map the grammatical relations the parser deals with into its own case roles.) In return, the parser can ask the case frame interpreter for many different sorts of information: whether or not a given NP or prepositional phrase can fill a given grammatical role, given the information the case frame has so far; which of two potential candidates the case frame would prefer in a given grammatical role, and by how much; how many slots of various sorts are still open in the case frame; etc. Furthermore, when necessary, the parser can ask precompiled fill-in-the-blank questions of the world model itself (the world model in the current system being the author); to resolve the ambiguity of a phrase like "a third cup of sugar", for example, the parser might ask the world model a precompiled question equivalent to asking whether it makes sense in the present discourse context to speak of a 3rd (as opposed to 1/3rd) cup of sugar.

Regardless of which source or sources of information a module uses, it will often need to know not merely whether one hypothesis or another is acceptable, but which of two hypotheses is better. A module will often need to ask the case frame interpreter, for example, not merely whether some NP is semantically acceptable in a given role for a given verb, but rather which of two NPs is better in that role and by how much. A good example of this is the diagnostic used by the module which resolves the question of which NP serves as direct and indirect object in sentences such as

(i) Who did John give the book?
(ii) What did John give the elephant?
(iii) Who did John give the elephant?

(where (iii) may be at least questionable in the idiolects of some readers). This diagnostic is also a good general example of the sorts of diagnostic rules that a W&S grammar contains. The module containing this diagnostic is in packet WH-question after VG, active when the underlying role of the question group has not yet been determined. It is triggered when a single NP follows the main verb of the clause, followed by neither another NP nor a Prepositional Phrase. The module first asks the case frame interpreter whether more than one slot is feasibly open for NP objects. The answer here is yes as there are two slots open, so the module knows that it can use both the following NP and the question group to fill these slots and it tells the case frame interpreter to commit itself to using both slots. The remaining diagnostic is essentially this: while the module has a mild syntactic preference to use the following NP as indirect object, it will accept the question group as indirect object if the case frame strongly prefers it to the following NG as IO, with a feature added to the top level clause indicating that the clause is a wee bit ungrammatical; this is the path taken for (i) above. If the case frame does prefer the next NP to the question group, the module is very happy; this is the path for (ii). If the case frame mildly prefers the question group to the following NP, balancing the syntactic preference, as in (iii), the sentence is perceived as very wierd (although a few people have no trouble here), and the module will do in desperation what many speakers seem to do in this case - take the question group as IO on semantic grounds while complaining loudly.

W&S As A Psychological Model

Though the diagnostic above is far more specific in both applicability and in detail than what is normally considered a perceptual strategy, it fills much the same role that perceptual strategies are assumed to play. There are many crucial differences, however. Wait-and-see diagnostics are treated as rules of grammar in the W&S theory, and the parser applies them in a consistent, rule-like manner. Indeed, in the grammar itself (unlike the theoretical framework), no differentiation is made explicitly between grammar code that decides what to do and grammar code that does it; both are integral parts of the grammar. Wait-and-see diagnostics, unlike perceptual strategies, in general use syntactic and semantic distinctions together to diagnose the correct course of action, although the information a diagnostic seeks is usually very specific and very focused. It is also important to note that a diagnostic chooses between a set of very specific possible options, and is not at all a general rule of thumb.

But what if one of these W&S diagnostics fails? If the parser takes a wrong turn because one of the W&S diagnostics in its grammar was misled, then the parse cannot be successfully completed and the sentence is a "garden path" with respect to that grammar. In these situations, however, a W&S parser will not "fail", in the sense of throwing up its hands and yielding no parse at all, but rather it will yield up whatever structure it has constructed at the point of blockage and will attempt to build whatever grammatical substructures it can with the remainder of the input, much as people seem to do when confronted with sentences that are garden paths with respect to their grammars. A higher level problem solver might then use higher level grammatical knowledge to try to diagnose the source of the garden path, e.g. it might know about dangling participles and how to fix them.

This property of W&S grammars suggests a global test of the plausibility of the W&S theory as a psychological model as well as a local test for the adequacy of any diagnostic within a W&S grammar: Not only should an ideal W&S grammar be able to parse correctly all sentences that people parse correctly, but it should also perceive as garden paths exactly those sentences that people perceive as garden paths. At the present stage of grammar development, it should be added, it does not seem to be difficult to build individual W&S diagnostics that behave locally as people do in terms of perceptions of garden paths. To this extent it seems reasonable to suggest that people may use diagnostics that are similar to the diagnostics the W&S model posits.

One other property of the implemented W&S system is also interesting in terms of the plausibility of the W&S model as a psychological model. It turns out that the length of time required for the parser to parse input is directly proportional to the length of the input; the time per word required by the parser to parse any sentence seems to vary by no more than 40% over the time taken to parse "simple" sentences. Constructions that do take proportionally longer than simple sentences do so because of factors such as the additional computation needed by diagnostics that determine where to insert "deleted" structures and the like, but, again, the total increase in time per word, averaged over the entire sentence is rarely more than 40% greater than simple sentences. These time relations are consistent with the range of results from psychological experiments that attempt to measure latencies for sentence comprehension (such as [Wanner ´74]), although no detailed comparison of the time behavior of the parser and that measured by psychological testing has yet been undertaken. It is possible that this time behavior will change as the W&S grammar grows, but this sort of behavior would seem to be intrinsic to any parser and grammar based on the W&S model.

REFERENCES

Bever, T.G., The cognitive bases for linguistic structures. In Cognition and the Development of Language, J. Hayes, editor, John Wiley & Sons, 1970

Fahlman, S., A Hypothesis-Frame System For Recognition Problems, Working Paper 57, MIT-AI Lab, 1973

Marcus, M., Wait and See Strategies for Parsing Natural Language, Working Paper 75, MIT-AI Lab, 1974

Wanner, E., , Kaplan, R. and Shiner, S., Garden Paths in Relative Clauses, unpublished, Harvard University, 1974

Winograd, T., Understanding Natural Language, Academic Press, 1972

Woods, W.A., Kaplan, R.M. and Nash-Webber, B., The Lunar Sciences Natural Language Information System, BBN Report No. 2378, 1972

COMPUTATIONAL UNDERSTANDING

Christopher K. Riesbeck

I. METHODOLOGICAL POSITION

The problem of computational understanding has often been broken into two sub-problems: how to syntactically analyze a natural language sentence and how to semantically interpret the results of the syntactic analysis. There are many reasons for this subdivision of the task, involving historical influences from American structural linguistics and the early "knowledge-free" approaches to Artificial Intelligence. The sub-division has remained basic to much work in the area because syntactic analysis seems to be much more amenable to computational methods than semantic interpretation does, and thus more workers have been attracted developing syntactic analyzers first.

It is my belief that this subdivision has hindered rather than helped workers in this area. It has led to much wasted effort on syntactic parsers as ends in themselves. It raises false issues, such as how much semantics should be done by the syntactic analyzer and how much syntactics should be done by the semantic interpreter. It leads researchers into all-or-none choices on language processing when they are trying to develop complete systems. Either the researcher tries to build a syntactic analyzer first, and usually gets no farther, or he ignores language processing altogether.

The point to realize is that these problems arise from an overemphasis on the syntax/semantics distinction. Certainly both syntactic knowledge and semantic knowledge are used in the process of comprehension. The false problems arise when the comprehension process itself is sectioned off into weakly communicating sub-processes, one of which does syntactic analysis and the other of which does semantic. Why should consideration of the meaning of a sentence have to depend upon the successful syntactic analysis of that sentence? This is certainly not a restriction that applies to people. Why should computer programs be more limited?

A better model of comprehension therefore is one that uses a coherent set of processes operating upon information of different varieties. When this is done it becomes clearer that the real problems of computational understanding involves questions like: what information is necessary for understanding a particular text, how does the text cue in this information, how is general information "tuned" to the current context, how is information removed from play, and so on. These questions must be asked for all the different kinds of information that are used.

Notice that these questions are the same ones that must be asked about ANY model of memory processes. The reason for this is obvious: COMPREHENSION IS A MEMORY PROCESS. This simple statement has several important implications about what a comprehension model should look like. Comprehension as a memory process implies a set of concerns very different from those that arose when natural language processing was looked at by linguistics. It implies that the answers involve the generation of simple mechanisms and large data bases. It implies that these mechanisms should either be or at least look like the mechanisms used for common-sense reasoning. It implies that the information in the data bases should be organized for usefulness -- i.e., so that textual cues lead to the RAPID retrieval of ALL the RELEVANT information -- rather than for uniformity -- e.g., syntax in one place, semantics in another.

The next section of this paper is concerned with a system of analysis mechanisms that I have been developing. While the discussion is limited primarily to the problem of computational understanding, I hope it will be clear that both the mechanisms and the organization of the data base given are part of a more general model of human memory.