ACTIVATION-BASED PARSING

Mark A. Jones*

State University of New York at Stony Brook
Long Island, New York 11704

ABSTRACT

A model is presented that describes natural language parsing in terms of a uniform activation algorithm which operates over an interconnected, declarative structure of nodes and node instances. The algorithm directs the flow of activation and expectation using only local information and, hence, supports substantial concurrency. A representation is introduced to express node relationships and activation agreement Examples of several linguistic processes and their interrelationships are described

1. Introduction

The concept of a layered or stratified system has been a central notion in many linguistic theories. A system is usually defined in terms of objects at various levels of linguistic description and inter- and intra-level mapping rules. Traditional levels include phonology, morphology, syntax, semantics, and pragmatics. The objects include syntax trees, deep structure trees, lambda calculus formulas, predicate calculus formulas, and frame (slot-filler) representations. The corresponding rule formalisms include tree transformations, lambda application and reduction, logical implication, and frame composition. The rules, particularly in computational theories, are often context-free in appearance, but occasionally possess augmentations to restrict their application, to build new objects, and/or coordinate the application of other rules.

As the role of pragmatic knowledge has grown in AI systems for tasks such as story understanding, a broad new spectrum of "knowledge structures" has been introduced.** The conceptual status of these proposed objects and the nature of the indexing or mapping rules are still the subjects of inquiry and debate in cognitive science The interplay of the processing components of the programs and the indexing properties of the objects sometimes makes it difficult to extract the exact character of the mapping rules, but they appear to be similar to other rule-oriented formalisms.

The current work attempts to unify the insights from linguistic and conceptual analysis into a computational theory based on *activation.* Activation-based computational methods are defined here to mean computational methods in which the interconnection *structure* of a body of declarative knowledge plays an active role in the *processing* of information. Our concern here is with models in which the interconnected memory structure is as active as possible, and our focus is particularly on the task of natural language parsing. The term activation-based parsing refers to the processing of an input string by the initial (sequential) stimulation of memory nodes corresponding to the constituent substrings (generally words) and by the subsequent activation of other nodes by these. The parse is represented by activation traces derived from the interconnection of existing nodes. A parser, named ABP, is implemented in Franz Lisp to simulate the model that is presented.

Particular attention has been given to the problems of representing and effectively using generalizations about language and the underlying concepts that language attempts to convey. Specifically, syntactic generalizations have not been distributed into the detailed semantics as in semantic grammars [1] or embedded in the lexicon as in word-based parsing approaches (cf. [7, 6 and its descendants]) The phenomena of metaphor processing (where semantic selectional restrictions may be violated), of sensitivity to the phrasing of an utterance, of sensitivity to agreement violations, of novel language use, and especially of language learning attest to human abilities to exploit syntactic generalizations profitably. Our view is that generalization is central to the classification and use of concepts in memory and accounts for much of the flexibility in human information processing. This does not imply that lexically activated (word-based) expectations for the idiosyncratic tendencies of language do not exist, nor that utterances must be completely "grammatical" to be understood [the model can account for both of these observations]. The point here is simply that generalizations (eg., syntactic generalizations) do exist and should be helpful in processing activities This emphasis on generalization anticipates future work on language learning and serves as an inspiration for many of the representational issues that we have faced.

2. The Parsing Model

Previous work in natural language understanding in AI has underscored the need for applying a wide range of comprehension activities into the parsing process. These activities involve syntactic processing and a wide variety of semantic processing including contextual integration, word sense disambiguation, anaphora resolution, metaphor processing etc The bandwidth among these processing activities in traditional implementations has been low, and as Small points out, "... it seems that any strict serial decoupling of associated tasks ... does not permit enough interdependence of function to explain adequately reasoning tasks that people find quite easy and natural to perform."[8]

Our parsing model is defined in terms of a set of semi-distinct levels that represent various types of concepts and processes. Each level has a generally hierarchical infrastructure of nodes and, additionally, possesses connections to other levels. The inter-level connections permit concurrent recognition behaviors at various levels. Strictly speaking, the nodes in the various levels are not activated themselves, but rather instances of them, and there may be more than one instance of a particular node active at any time. Each node can be viewed as a type of production rule which is described in terms of other node activations. A uniform activation algorithm directs the creation and interpretation of node instances. The algorithm is sensitive to the current state of active instances and the recency of expectations placed upon the node by other instances. Beyond the activation algorithm, the "process" component of natural language understanding is embodied in the declarative interconnection structure. This distinguishes the approach here from production systems or semantic network schemes in which a substantial amount of the processing behavior is still embodied in program code.

The primary aspects of the parsing model are (a) the specification of the recognition processes and node interconnections

---

♦ This paper is based upon work supported by the National Science Foundation under grant #1ST-82081I3 and by the University Awards Program of the Rrsearch Foundation of the State University of New York, fellowship #0271-03-431-81-0.

**One recent dissertation [2] offered 17 different knowledge structures and 28 types of interactions among them.

that define the levels and (b) the design of the production language and the activation algorithm. The initial level definitions are given below. Most of our attention thus far has focused on the form and interplay of the levels 0-4.

Level 0 - the token level, consists of words and a small set of affixes which have a clear productive or agreement function. The model should be relevant to underlying phonemic and morphemic levels as well but we begin at the token level for simplicity and assume that nodes at this level are exogenously stimulated. Except for grouping conjugated verb forms such as see/saw/seen as see+ and noun forms with irregular plurals man/men as man-t- , there is no internal structure to level 0.

Level 1 - the syntactic level, consists of syntactic classifications of tokens, phrasal patterns and their classifications, idioms and syntactic agreement patterns. Syntactic word classes form the primary links to level 0, although some syntactic patterns require particular word forms such as do+ .

Level 2 - the conceptual classification level, provides semantic classifications of objects from levels 0 (words), and 1 (idioms and modifiers such as adverbial phrases and prepositional phrases). These classifications are used by higher levels to sort nominal and event descriptions.

Level 3 - the NP role level, uses activation from levels 1 and 2 to sort the NP concepts according to their (syntactic and semantic) roles in the sentence. The subject NP concept will be classified as a subject and either an agent or an object depending on the voice of the verb phrase. Based on the level 2 classifications, it may also be refined from *agent* into *actor* or *instrument.* Note that certain syntactic phenomena such as tag questions require distinctions such as subject that are intermediate to full conceptual resolution of roles.

Level 4 - the case frame level, is responsible for word sense disambiguation. Here are the detailed expectations of certain verb senses for particular prepositions and particles. The case frames are organized hierarchically for the purposes of shared inferences. For example, the senses of throw, roll, and go which indicate physical transfer will be assembled under a ptrans node.

Levels 5 and higher - the contextual levels, identify sequences of level 1 nodes which relate to temporal ordering, causality and plan recognition.

Figure 1 illustrates a fragment of a simple syntactic level structure. Each node may be viewed as a type of production rule which has been divided into a pattern *(pat)* and an optional test *{test).* Figure 2 lists the definitions for several of the nodes in Figure 1. The tests are listed only in Figure 2. The production language allows

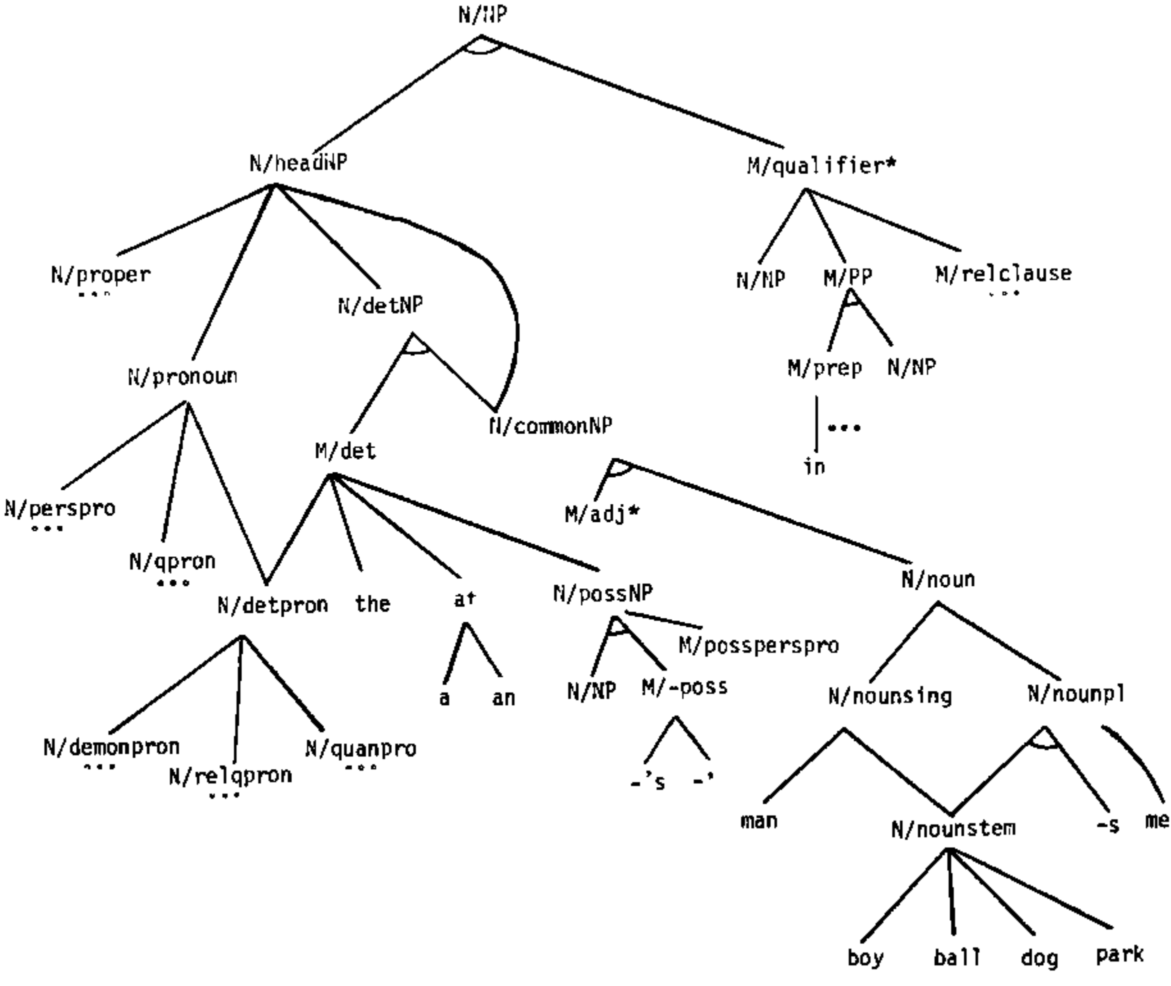


Figure 1: Example of a Syntactic Level Fragment

sequentiation *[seq),* alternation (*or),* a *case* construct, and a repetition operator (*). The case construct contains antecedent-consequent pairs; if an antecedent activation is seen then a consequent activation must also be seen or the construct is blocked.

```
N/NP
    pat: (seq (n1 . N/headNP) (m1 . M/qualifier)*)
N/headNP
    pat: (n1 . (or N/proper N/pronoun N/detNP N/commonNP))
M/PP
    pat: (seq (m1 . M/prep) (n1 . N/NP))
    test: (case ((n1 . N/perspro) (n1 . N/objperspro)))
N/detNP
    pat: (seq (m1 . M/det) (n1 . N/commonNP))
    test: (or (seq (m1 . m/singdet) (n1 . N/nounsing))
              (seq (m1 . M/plurdet) (n1 . N/nounpl)))
```

Figure 2: Examples of Node Definitions

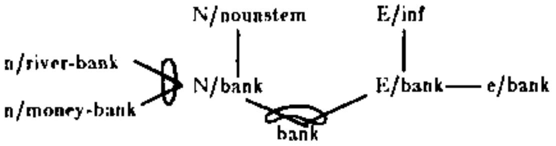
The production rules (nodes) explicitly index one another via the interconnections among them. An instance is active if its pattern or test has been partially satisfied. An instance is triggered (or fired) when no longer inhibited by its pattern or test. For syntactic level nodes, the pattern typically represents the context free base component of language, and the test implements context sensitive aspects such as subject-verb agreement, use of the objective case, and determiner-noun agreement. The pattern can alternatively be viewed as a generalization and the test as a specialization of it.

Although node names are completely uninterpreted and arbitrary, naming conventions help to distinguish the levels* Token node names in level 0 are not prefixed. Syntactic node names prefixed by N/, E/. and M/ represent syntactic nominal, event and modifier concepts, respectively in level 1. Node names at levels greater than 1 are prefixed by n/, e/, and m/ in this paper. The terminal nodes (English morphemes) in Figure 1 are from the token level. Just above these nodes are the terminal syntactic level nodes which include lexical categories The interior nodes in the syntactic hierarchy provide phrase patterns and phrasal combinations.

3. Non-determinism

A purely bottom-up activation-based parser encounters trouble due to the fact that the connection structure is not exclusively tree-like Figure 1 contains several nodes leading upward to competing alternatives - N/nounstem, N/comrnonNP, N/detpron, and N/NP. Competing alternatives for node activation cause purely bottom-up schemes to be highly non-deterministic.

A shared structure may be linked to independent sets of alternatives, termed dimensions. Alternatives in different dimensions are non-competing. For example, pronouns activate features in dimension of gender, number, person and case in addition to participating in NP recognition Activation from a node to non-competing alternatives is deterministic with respect to the activation in each dimension. The figure below shows an example of competing (circled) and non-competing alternatives.



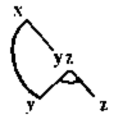
Competing alternatives can be further distinguished into several cases:

(a) *Final epsilon-transitions*: Natural language syntax commonly involves productions in which optional elements may appear following an obligatory element. For example, a (regular) noun stem can be optionally followed by -s to form the plural. Present tense verb forms are another similar example. The situation is not confined to suffixes. A pronominal determiner (that, which, some) can serve as an NP when used pronominally or can serve as a determiner for a common noun NP. The head of an NP can be optionally followed by qualifiers. Many verbs can be used transitively and intransitively, which causes premature VP firing. Furthermore, the VP may have trailing PP modifiers. The sentence Y*ou walk ed htr dog -s in the park* contains contains several initial subsequences:

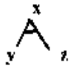
(i) You walk.
(ii) You walk -ed.
(iii) You walk -ed her.
(iv) You walk -ed her dog.
(v) You walk -ed her dog -s.
(vi) You walk -ed her dog -s in the park.

Two subclasses of optional elements can be distinguished:

(a.1) *Final Optionality*:
x → y | y z represented as*

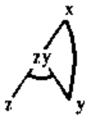


(a.2) *Final Repetition*:
x → y z* represented as

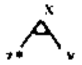


(b) *Initial epsilon-transitions*: These cases are similar to (a.1) and (a.2) except that the optional element precedes the obligatory one. Optional determiners and adjectives are examples of initial optional elements.

(b.1) *Initial Optionality*:
x → y | z y represented as



(b.2) *Initial Repetition*:
x → z* y represented as



Shartd sub-structures: The epsilon-transition cases (a) and (b) above appear to occur primarily in endocentric constructions in which the optional element does not affect the "type" of the recognized phrase in natural language syntax. The type is established by the head of the phrase which is the obligatory element -- for example, the noun in a noun phrase or the verb sequence in a verb phrase. The same syntactic interpretation survives in these cases and the trailing constituents serve for purposes of agreement or qualification. In other cases, however, competing alternatives may play fundamentally different roles in the interpretive process. Structures such as NP or PP may be shared by many different constituents. For example, an NP can appear as the subject of a sentence, the first or second object, the object of a preposition, etc. Token level words may give rise to multiple syntactic categories. Polysemic words activate multiple nodes at the semantic levels.

---

*The level distinction itself is, of course, somewhat arbitrary and primarily exists to emphasize the general processing behavior characterized by the nodes in a given level.

•This representation was chosen in lieu of a notation for optionality, because the resulting cases (x -> y) and (x -> y z) often need to be explicitly referenced elsewhere (e.g., for context-sensitive restrictions). The new node, yz, conveniently allows this reference.

(d) *Recursion:* Recursion introduces cycles into the node structure. Figure 1 includes two cases of right recursion (the NPs appearing in the appositive and the PP) and one case of left-recursion (in M/possNP). Recursion can also involve center embedding as with relative clauses.

4. The Activation Algorithm

This section describes an activation algorithm which uses a breadth-first, parallel expansion of activation. The algorithm attempts to curb non-determinism up to genuine ambiguity by augmenting the bottom-up activation with a complementary operation of top-down expectation to provide a biasing affect on alternatives within a dimension. The expectation-driven interpretation of node activation is a powerful principle for controlling the integration of knowledge structures that have been recognized bottom-up. The characteristic pattern of the system is a rippling of activations "upward" and of expectations "downward" both withen and among levels. Under this principle, the non-determinism represented by parallel, competing alternatives is restricted by the context.

The phrase *the man* illustrates the principle as it applies to case (b.l) in section 3. The following sequence of actions takes place:

(i) the0 is exogenously created and triggered.

(ii) M/detO is activated, followed by N/detNPO which is blocked.

(iii) manO is then created, triggered and followed by N/nounsingO, N/nounO and N/commonNPO.

The node structure reveals two competing alternatives for N/commonNP: N/detNP and N/headNP. The alternative N/detNP is biased, however, because N/detNPO posted an expectation for N/commonNP as it became blocked in (ii).

(iv) The remaining instance activations and firings are: N/detNPO, N/headNPO, N/NPO, ...

The following rule summarizes the constraint:

Rule 1:

for all dimensions stemming from a node (e.g., N/commonNP)

if any active instances (N/detNPO) expect an instance (N/cornmonNPO) of the node,

then the most recent expecting instances are activated and other expecting instances are inhibited.

As it has been developed thus far, the model seeks to rely on a discrete view of activation and inhibition instead of using more continuous notions of activation levels, thresholds, and decay. In our implementation, the recency effect needed for Rule 1 is modeled by having active instances "post down" their expectations for constituents on FIFO lists that are associated with the nodes. The implementation treats each expectation that arises from the same exogenous activation as having the same age. The activation and expectation processes are currently assumed to complete before the next input. Recency could alternatively be implemented by comparing continuous activation time stamps on instances or by using decaying activation levels to emulate the time stamps.

The phrase *the boy -8* demonstrates the type of non-determinism described in ease (a.l). The initial substring *the boy* is itself a noun phrase. At node N/nounstem there is an upward split leading to nodes N/nounsing and N/nounpl. The paths rejoin at N/noun, and any top-down expectations posted through N/noun will fail to distinguish a competing alternative. The following sequence of activations tales place:

(i) theO, M/detO, N/detNPO (blocked)

(ii) boyO, N/nounstemO, N/nounplO (blocked) and N/nounsingO, N/nounO, N/commonNPO, N/detNPO, N/headNPO, N/NPO, ...

When -sO and N/nounplO are activated next, we are faced with a dilemma at node N/noun. Should we create a new instance N/nounl (for *boy -s)* or modify our previous instance, N/nounO, that now represents *boy?.* N/nounl cannot be generated, because the prior expectation environment has been altered by the sequence in (ii) and it would load to a new NP instance. The solution is to link posted expectations together in such a way that the bottom-up activation retraces the expectation structure. At junctures in the expectation structure (eg., N/nounO) in which an instance (N/nounsingO) has already satisfied the expectation, new instance activations of the juncture node are suppressed. The old activation links from N/nounsingO to N/noun() can be severed and a new link from N/nounpl() to N/nounO can be established. The general rule is:

Rule 2:

fur all dimensions stemming from a node (e.g., N/nounpl)

if any instance (N/nounsingO) has already satisfied the same expectation (in N/nounO) that an instance of the node (N/nounplO) wants to satisfy

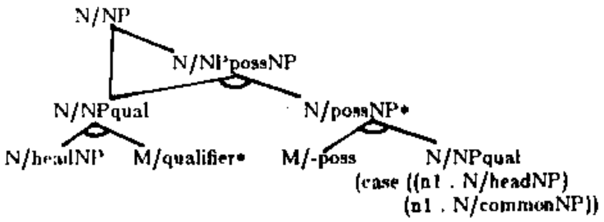
then the last binding (from N/nounO) is redirected (to N/nounplO instead of N/nounsingO)

Due to the semantic penetrance of activation from the initial words in a sequence, language has evolved certain relationships across levels in the form of constraints on number, person and gender. In the previous examples, the determiner *the* failed to bias the N/nounsing and N/nounpl alternatives of N/noun; however, determiners such as *a, every, that, and these* can express a preference. In our model these determiners activate feature concepts such as n/singdet and n/plurdet in a number dimension in addition to syntactic level activations. The agreement restriction is defined at the coordinating node (N/detNP) and is taken into account when posting expectations For the phrase, *a boy,* expectations will only be posted in the N/nounsing branch below N/noun. This will cause the bottom-up activation at N/nounstern to direct itself only toward N/nounsing.

The repetition cases, which only involve a single production, are easier to handle. Case (b.2), initial repetition, is similar to normal sequential pattern activation. In case (a.2), final repetition, the production fires after the obligatory elements, and the optional, repeated elements simply attach in the trace. Expectations are regenerated after the attachment.

Space does not permit full examples of the recursion cases, so we will only highlight them The existing rules handle the case of right recursion with no difficulty. Center-embedded structures simply require that the UFO lists (stacks) of node expectations are popped as instances are created. The algorithm does this anyway since a node expectation is "transferred" of an instance when the instance is created*

Left-recursion poses the greatest difficulty for the expectation (top-down) mechanism due to the possibility of expectation cycles. The two options are to eliminate left-recursion by rewriting the grammar or to define additional rules to handle it. To eliminate the left-recursion in Figure 1, the M/possNP node can be deleted as a determiner (leave M/possperspro) and the top of the NP structure can be changed as follows:



The objection to modifying the grammar is primarily that the syntactic trace structure does not mirror semantic intuitions as closely as the left-recursive trace. To handle left-recursion, the expectation process must be modified to eliminate cycling by having a node realize when it receives an expectation that it has already

---

*An equivalent way of thinking about expectations is to consider them to be *proto-instances* created by top-down "activation" which survive in the trace only if reinforced by bottom-up activation.

sent *[i.e.,* that the expectations are in the same expectation structure). The recursively called node non-deterministically activates the left-recursive instance and the calling instance. Expectations below the recursive node are spawned again when the left-recursive instance fires and the calling instance is rebound to the new recursive instance.

An Knglish grammar subset of 110 syntactic level nodes has been encoded to explore various interesting recognition issues. The subset includes the verb tense system for declarative sentences, relative clause constructions and active/passive voices. The syntactic level algorithm described above has been implemented and handles the subset.

5. Related Work

Fahlman's NETL system [3| is an example of a marker-passing scheme. An interesting use of NETL to determine noun-noun modifying relationships is described in [5]. Woods [I0| discusses schemes using marker propogation in taxonomic lattices that might be used in conjunction with an ATN (augmented transition network) grammar.

ATN implementations and, in particular, Woods' notion of cascaded ATN grammars [11] are related to the work presented here. Many ATN implementations have used techniques such as backtracking, arc-ordering, bounded lookahead and well-formed substring tables which are designed for efficient execution on sequential machines. It is possible, however, to define parallel ATN implementations which do not need these techniques. ATN registers are used for feature information extracted from the lexicon, for constituent structures such as subject and object, and for dislocated constituents. Pop arcs return structures that are assembled from the registers. Our view has been *not* to explicitly build and pass syntactic or semantic representations (data structures), but to allow the "meaning" of an input to be represented by the impact that it makes upon the instance and expectation structure*

Cascaded ATNs include the concept of concurrent operation of multiple ATNs at different levels. The emphasis is still on the passing of structures that are created and held at one level until an opportune time to transmit them. In our work there is a less rigid distinction of levels in terms of global and immediate availability of node activations and there is a greater assumed bandwidth among recognition processes. Our architecture allows a greater penetration of both activations and expectations to reduce non-determinism. The uniform architecture also is desirable for future considerations of learnability.

A detailed comparison of the standard ATN grammars and the type encouraged by the activation-based formalism is beyond the scope of this paper, but one example from [I I| will be discussed. An interrogative sentence can begin with an auxiliary verb (a form of *do, be, have,* or a modal) followed by the subject; whereas, the order is reversed in a declarative sentence. Woods discusses the awkwardness of ordinary PSGs (phrase structure grammars) to account for this. In our grammar, the pattern for one of the sentence nodes, E/Qaux, is of the form:

(seq (ml . (or do+ be+ have+ M/futmodal)) (el . E/declS))

This behaves similarly to Gazdars *holes [14].* If the auxiliary is seen, then E/Qaux posts an expectation for a declarative sentence. Note that some verb sequence instances have already been created in the prior activation of the auxiliary. Now ihe sentence is recognized as usual and the verb sequence activations are completed. NP holes in relative clauses can be treated similarly.

6. Future Work

We have begun expanding the model to deal with levels 2-5. Level 2 contains a conceptual recognition structure similar to standard isa-hierarchies, but with a rich categorization structure. The interior nodes in the level 2 structure serve largely to facilitate the conceptual expectations of higher level structures.

Level 3 uses the semantic information in syntactic order from level 1 and the conceptual categorizations to sort the NP concepts into case role relationships for level 4. We are designing a description language to aid in expressing compositional relationships that can be used to "compile" new node descriptions. Let us assume that a declarative sentence pattern, E/declS, looks like: (seq (nl . N/NP) (el E/VP)).

We can express our semantic intuitions about the level 3 and 4 mappings in the following (head modifier*) structure:

```
((el . e/event) (if ((el . e/active) (m/agent (nl . n/nominal)))
                    ((el . e/passive) (m/object (nl . n/nominal)))))
```

I'sing the compositional relationship and the pattern results, for example, in a compiled node pattern for m/agent of: (seq (nl . n/nominal) (el . e/active)).

7. Conclusions

An activation-based model for natural language processing has been outlined An activation algorithm that uses local information regarding the current state of active instances and expectation structures has been applied to syntactic processing and is now being extended to other processes. The algorithm uniformly uses expectations to bias the activation spread at non-determininstic points in the declarative node structure.

The model should have interesting applications to difficult problems such as ellipsis, deletion, conjunction and anaphora which depend highly on the *recency* and type of node activity. The concurrency in the model suggests ways of using the redundancies in natural language communication to proceed with an interpretation in spite of ill-formed or incomplete input. The model also offers advantages in extensibility including (1) the properties of additivity and continuity in the integration of new knowledge, and (2) the fact that system performance should not seriously degrade with additional knowledge.

REFERENCES

[I] Burton, R. Semantic Grammar: An Engineering Technique for Constructing Natural Language Understanding Systems, BBN Report No 3153, Bolt, Beranek and Newman Inc., Cambridge, MA. 1970.

(2] Dyer, M. In-Depth Understanding, Research Report #219 (doctoral dissertation), Dept. of Computer Science, Yale University, New Haven, CT, 1982.

[3] Fahlrnan, S. *NETL: A System for Representing and Using Real-World Knowledge,* MIT Press, Cambridge, 1979.

[4] Gazdar, (J. A Phrase Structure Syntax for Comparative Clauses, *Lexical Grammar* (ed. Teun Hoekstra), Foris Publications, Dordrecht, 1980, 165-179.

|5] MacDonald, D.B. Understanding Noun Compounds, doctoral dissertation, Carnegie-Mellon University, Pittsburgh, 1982.

|(>] Riesbeck, C. and Schank, R. Comprehension by Computer: Expectation-Based Analysis of Sentences in Context, Research Report #78, Dept. of Computer Science, Yale University, New Haven, CT, 1970.

[7] Small, S. Word Expert Parsing. A Theory of Distributed Word-Based Natural Language Understanding, TR-954 (doctoral dissertation), University of Maryland, College Park, MD, 1980.

[8] Small, S., Cottrell, G. and Shastri, L. Toward Connectionist Parsing, AAAI-82 Proceedings, Pittsburgh, 1982, 247-250.

[9] Tarnawsky, GO. Knowledge Semantics, doctoral dissertation, Dept. of Linguistics, New York University, 1982.

110] Woods, W.A. Taxonomic Lattice Structures for Situation Recognition, TINLAP-2, Theoretical Issues in Natural Language Processing-2, University of Illinois at Urbana-Champaign, 1978, 33-41.

(II) Woods, W.A. Cascaded ATN Grammars, American Journal of Computational Linguistics, 6, 1 (1980), 1-12.

---

* See [9| for a linguistic theory of a similar notion of meaning.