The Logos Model as a Metaphorical Biological Neural Net

By Bud Scott, Chief Architect

To better account for the novel way in which the input stream and the linguistic knowledge store (dictionary, rules and tables) interact in the Logos Model, in what follows we describe the Logos Model in terms of a *biological* neural net or bionet. The Model is felt to have interesting parallels with certain features of brain reticula (Scott,1990). Though clearly superficial, the correspondences are nevertheless felt to be significant in several key respects. Readers will judge for themselves the aptness of the connection. In any event, this biological metaphor has proven to be an effective way to describe how the Model's memory (stored knowledge base) interacts with input signals (text). As you will see, it does so in ways that may be said to mimic associative memory assumptions about brain function.

No claim whatsoever is being made here that the Logos Model has anything in common with *artificial* neural nets (ANN's). It clearly does not. The reasons are straightforward:

- The Logos Model is symbolic in nature. ANN's generally entail purely quantitative, linear algebraic functions (ignoring for the moment some experimental hybrid exceptions).

- The Logos Model employs long-term memory units (rules) that have fixed, specified significance, reflecting what in neurobiology is called the "single neuron doctrine" (Barlow,1972). ANN's, by contrast, hold that meaning is represented by spreading activation over an entire network ("distributed representation") which means that individual cells have no fixed significance (Hinton, McClelland and Rumelhart, 1986). In ANN's, units are reusable like the receptors of a charged couple device in a digital camera's sensor.

- The Logos Model does not "learn" by means of back-propagation or any other means of unsupervised training. All training is supervised and adjustments to rule weights (e.g., to strengthen or weaken certain rules, as in **Res** modules) are effected manually.

The Model does share some broad features common to neural nets in general, ANN's or otherwise. These include:

- <u>Constraint satisfaction</u>. Processing may be viewed as a form of constraint satisfaction on individual units (rules). In the Logos Model, constraints are "strong" as opposed to "weak," i.e., represent a binary yes/no condition (unlike Boltzmann or Hopfield (1984) paradigms).

- <u>Emergent properties</u>. The parse of a given sentence emerges in unpredictable ways from the interactive processes of the net.

- <u>Competitive processing</u>. Units (rules) compete with each other in interpreting input data. The resulting parse can be construed as the set of computed winners. This competition is not network-wide, however, as in ANN's, but only among the small cluster of rules made active by any given input signal.

- <u>Self-ordering</u>. Units are self-ordering, self-applying objects (i.e., there are no meta-rules or master algorithms that apply rules to the input stream).

Furthermore, units define their own order and position within a layer. (Units however are not "self-organizing" in the sense of true Kohonen learning (Kohonen, 1982). No claim is being made that the Logos Model is an adaptive, learning system. It would be nice but it's not so.)

- Robustness. Processing extends to ambiguous, incomplete, ungrammatical, and unknown language strings.

- Graceful degradation. The incapacitation or loss of processing units (rules) will lead to degraded output but never to system failure.

1.0 Biological Neural Nets

The Logos Model was not designed with neural net technology in mind, in any of its varieties—indeed such thinking did not exist as a recognized NLP option in the 1970's. Nevertheless, the Logos Model has since been found to have arguable parallels to several properties of actual cortical circuits, seen from a purely computational perspective (Scott, 1990). The basis for this claim is as follows:

- Analogous to the cerebral cortex, the Logos Model consists of a very large number (thousands) of simple, neuron-like cells (rules) the individual function of which is extremely limited in scope, but which in conjunction with other cells contributes to large, complex effects

- Processing is accomplished by the reaction and interaction of tiny processing units (rules) across a sequence of layers, or laminae, as certain of these units are perturbed by input to the net and by unit interconnectivity. Rule interaction is effected entirely on the basis of memory associations (associative memory), not by supervening meta-logic or meta-rules of any kind.

- Ramón Y Cajal's classical view of cortical lamination (Crick and Asanuma, 1986) envisions six layers. The Logos Model serendipitously has six so-called "hidden" layers, with roughly the same proportional density of unit or rule distribution over the layers (See Fig. 1).

- Cortical circuits employ both vertical (inter-laminal) and lateral (intra-laminal) connections (Shepherd, 1994). So too the Logos Model. And like cortical circuits, the Model employs fan-in, fan-out, and recurrent circuitry. (See Fig 4. in: B. Scott, *Logos System—Principles and Motivations*.)

- Input vector/hidden layer interactions are highly specific, like synapses in cortical circuits (Anderson and Mozer,1989) (and unlike ANN's) (See Fig. 7).

- The role of amino-acid chains (neurotransmitters) in the recording and transmission of information in the cortex (synapse) is loosely mimicked by SAL chains in the Logos Model (See Fig. 7).

- Classical neuroanatomy has traditionally held that individual cells in long-term memory tend to be permanent, non-reusable records of information (Sherrington, 1941), spoken of nowadays as "local representation" (Hinton, McClelland, and Rumelhart, 1986). Local representation also characterizes the cell-equivalent rules of the Logos Model. (In neuroscience, such cells are said to have been "grandmothered" for their informational content, a term derived from experiments where a single simian brain cell was found to fire upon the appearance of the monkey's maternal grandmother.) This "single cell doctrine" is giving way in recent years to a more "clustered-cell" or "semi-local" explanation of cognitive function (Thorpe, 1995), and in that regard differs from the purely local representation of the Logos Model.

- Other points of similarity between the Model and neuroanatomy are discussed in Section 3.3 and are summarized in Fig. 4.

1.1 Logos Model Neural Net: Overview (Fig. 1)

The pipeline nature of the Logos Model is shown architecturally in Fig. 1 as a cascaded neural net comprising a sequence of six hidden layers interleaved with I/O vectors. The output of one layer constitutes input to the next. In Fig. 1, ambiguity is expressed by unit shading in **V1-V7**, which gradually lightens; complexity is simplified as the number of units in these vectors gradually reduces to **S.**

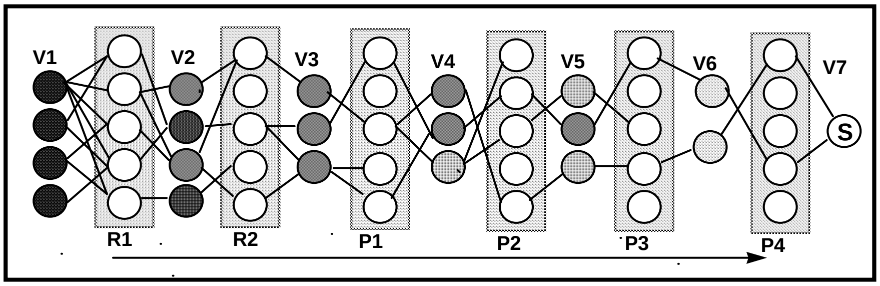


Fig. 1 Logos Model as a Biological Neural Net (Bionet). The Logos Model pipeline is seen here as a cascaded, six-stage neural net. **V1-V7** are input/output vectors comprising SAL objects that represent the elements of an input sentence. Each cell in the input vector represents a SAL element. Collectively these vectors constitute a progressive, bottom-up parse tree, ending in **S.** Shaded rectangles **(R1, R2, P1-P4)** are hidden layers. Cells in these layers represent pattern-based rules (tiny processing units). A layer may contain anywhere from two to four or five thousand such cells (rules). Each cell is specialized for a unique semantico-syntactic SAL pattern at levels of semantico-syntactic abstraction that tend to increase with each successive layer. See Fig.2 for other circuit types in the Logos Model not illustrated in Fig. 1.

SAL elements in vectors **V1-V7** can be thought of as temporary, short-term memory (STM). Units with their SAL patterns in the hidden layers may be seen as permanent, long-term memory (LTM). Units in the input vectors do not interconnect with all hidden layer units, indicating unit interconnectivity is specialized. Not evident here is lateral connectivity within a hidden layer (See Fig. 2). Also not evident is the presence of recurrent circuitry (See Fig. 4). Recurrent circuitry allows rule actions to feed back to the other rules in the same layer, inhibiting or increasing their potential to fire. Units in the hidden layers can also feed back to the input vector and modify SAL assignments or other properties of the input stream.

1.1.1 Graphic Illustration of the Logos Model as a Bionet (Figs. 3-9)

The set of figures (Figs. 3-9) in the following pages graphically illustrates the input stream/rule base interaction of the Logos Model as analysis proceeds down the pipeline, using the neural net metaphor.

For illustration purposes, we processed the following complex, 57-word English Example Sentence through the present Logos System, portraying the analysis in successive stages via the metaphor of a six-layered neural net (See Figs. 3-9). The sentence was taken from a text typical of those being processed through the Logos E-F system (on Wang VS) by a customer (Office of the Secretary of State in Canada) in the late 80’s, and is felt to be representative of the system’s parsing capabilities, both then and now.

EXAMPLE SENTENCE

Let me also note that because of the relatively close movement of the Canadian dollar with the U.S. dollar, our currency has declined along with the U.S. dollar against these other currencies this past year, removing much of the exchange rate distortion that was hampering the ability of Canadian firms to compete with producers overseas.

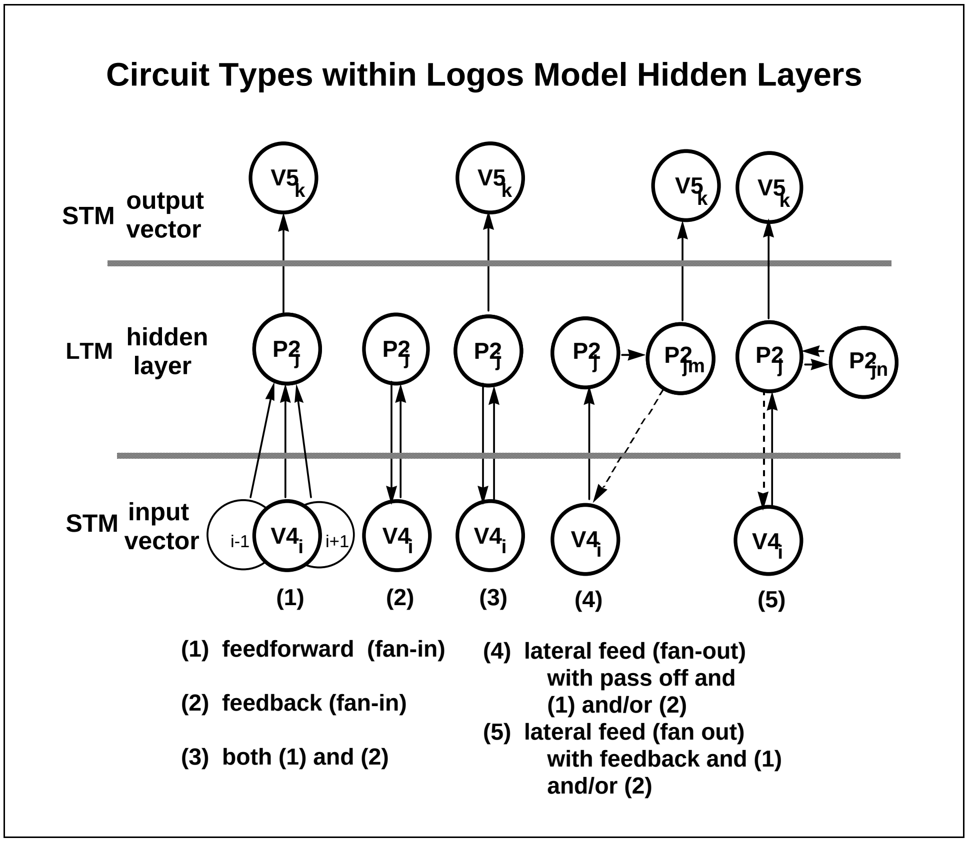


Fig. 2. Rule Base/Input Stream Interactions per the Bionet Metaphor. Graphic illustrates five basic circuits (interaction classes) between the knowledge base and the input stream. In (1), a rule fires when its SAL specification (pattern) best-matches the active segment of the input vector, and all rule constraints are satisfied. The effect of the firing typically is to re-write the pattern and output it to the output vector, which then serves as input to the next hidden layer. This represents classic forward-projecting synapse. In (2), the effect of the firing is back upon the input vector, typically causing some input code to be altered and the input segment to be resubmitted. In (3), both these effects are accomplished. In (4), the firing rule passes the input pattern which it matched to a local circuit (a nested block of rules) which takes over and effects actions as per (1), (2) or (3). In effect, $\mathbf{P2_j}$ functions as a filter to rule $\mathbf{P2_{jm}}$, which performs nested, finer-detailed matching. In (5) the same idea applies except that the nested block of rules, assuming one of them fires, returns control to $\mathbf{P2_j}$. (This is the interaction that takes place between a mainline rule and the **Semantic Table**.) An important feature of the Model's circuitry, too complex to show in this graphic, is recurrent circuitry whereby the firing of a cell effects an internal change to the current state of the entire hidden layer, increasing or inhibiting the subsequent potential for other rules to fire. (For an image of recurrent circuitry, see Fig. 4).

7.2.1 Bionet: Res1 and Res2 (Fig. 3)
The two **Res** (resolution) hidden layers, **R1** and **R2**, working successively, together effect a macro-parse of the Example Sentence. (In Fig. 3 we have conflated the two modules for purposes of this illustration.) Note, in Fig. 3, that input vector **V1** (comprising SAL objects that represent the NL string, following that string's lookup in the lexicon) is actually a two dimensional array, allowing for no more than three parts-of-speech for each NL word of the input sentence. This is an arbitrary limit imposed by complexity considerations, implications of which are discussed below.

Homograph Resolution. A key work of the **Res** modules is homograph resolution. This is accomplished by the resolving of a single path through the unresolved **V1** construction (out of a possible 2,239,488 paths in this particular sentence), such that each SAL element will have only one part-of-speech assignment in output vector **V3**.

Note that the word *close* in the Example Sentence (*close* is highlighted and underlined in Fig. 3) has three parts of speech, shown in Fig.3 in abbreviated form: (i) intransitive verb, (ii) transitive verb, (iii) adjective/noun. The action of **Res** on this particular word enables the path to **AJ/N** and inhibits the paths to *close* as **VI** or **VT**. The remaining adjective/noun ambiguity will be resolved by rules in a subsequent module (**Parse1**). This arrangement illustrates the incremental nature of analysis in a model of this type; ambiguity is removed as if by peeling the layers of an onion, one by one. Nevertheless, as stated, the chief reason for this limitation to three-parts-of-speech had to do with computational trade-offs.

To achieve a resolved path through the **V1** structure, the **Res1** and **Res2** software modules feed segments of **V1** and **V2** (as search arguments) to the **Res1** and **Res2** hidden layers. Segments are up to ten SAL elements in length, of each of the possible paths, starting at the top of the structure (beginning of sentence). Within the hidden layer, rules become active (i) that have a SAL pattern corresponding to the input segment and (ii) whose constraint conditions are satisfied. Active rules then compete for the right to fire, based on their dynamically computed relative weight. Weights are automatically generated from (i) rule length, (ii) semantic specificity, (iii) priority class, and (iv) a manually adjustable learning factor.

Clausal Segmentation. Another important work of the **Res** modules is clausal segmentation of the sentence, as depicted in Fig. 3. Every clause transition is identified and labeled, including parentheticals, relative clauses and other embeddings. It is obvious that homograph resolution and the detection of clausal transitions are mutually dependent and therefore problematic issues--one cannot be resolved with any degree of success without the other.

The **Res2** module is unique in having a limited look-ahead capability, provided in order to detect and avoid garden-path situations. Look-ahead is carried out by a common set of local rules, invoked by standard rules, an interaction depicted as Type (5) circuitry in Fig. 2. We illustrate this in (1), below:

(1) *The emphasis put on the question was wrong.* (Cp. *John put on his hat.)*

There is a **Res2** rule that handles the morphological class of verbs like *put* whose purpose in life is to resolve such verbs as *main verb* of the clause. But to do so a rule constraint must first be satisfied, namely, that there be no other more viable main verb candidate to the right in that clause. That constraint is tested by a function which invokes a block of rules looking for potential main verb candidates to the right. Effects of **Res2** rule actions on (1) may be seen in the raw German output in (1'):

(1') *Der Nachdruck, der auf die Frage gestellt wurde, wurde falsch.* (Cp. *John setzte seinen Hut auf.*)

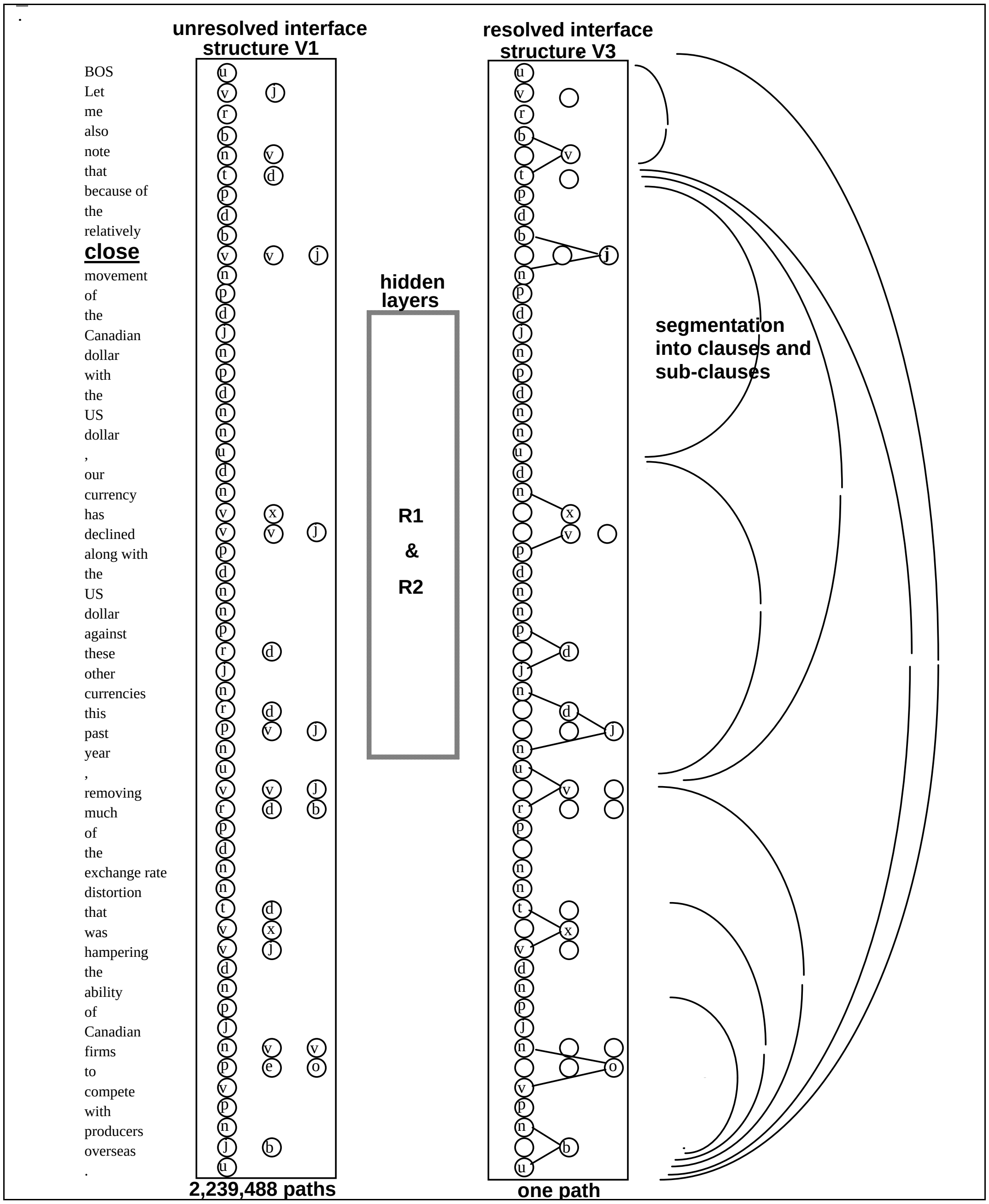


Fig. 3 - Bionet: Res1 and Res2. Res modules effect syntactic homograph resolution and analysis of clausal structure of the sentence. As illustrated in the present example, homograph resolution consists in finding a single path through the 2,239,488 possible paths afforded by this sentence in **V1**. SAL representation shown here is abbreviated to word class only. Actual representation includes the WC(Type; Form) triplet. Not all parts-of-speech are represented by a separate cell. For example, in the word *close* (shown above underlined and bold-faced), the adjective cell (j) itself is ambiguous, in reality representing *close* as both adjective and noun. Though the macro-parse effected by **Res1** and **Res2** eliminates *close* as a transitive and intransitive verb, the residual ambiguity must await the micro-parse effected by the **Parse** modules. A key strength of the **Res2** module in particular is its ability to analyze sentence clause structures, which in the present example is quite complex. The sentence overview effected by **Res2** provides critical top-down guidance to the subsequent micro-parse.

To avoid error propagation effects of misresolutions committed early in the pipeline, great pressure has always been placed on achieving high accuracy in the **Res** macro-parse. As a result, these modules have undergone several major redesign iterations over the course of time, inducing changes which have caused **Res2** in particular**,** the workhorse of the macro-parse, to become more and more cortical-like, as the following discussion will illustrate (see Fig. 4).

The **Res** modules taken together contain in excess of 6000 cell-like rules. As many as twenty percent of these rules may in fact be redundant, or ill-formed, having crept into the system one way or the other over the years. For whatever reason, these rules to date have not gotten the opportunity to fire and make their presence felt, and as such remain as so much undetected, seemingly useless baggage. Such rules could of course be systematically purged were it not for the fact that tests reveal that every so often, with new text, some of these rules do in fact get to fire for the first time and serve a good purpose. Given this uncertainty, such rules are generally left alone. This rather untidy circumstance reflects the fact that **Res** rules collectively do not constitute a body of coherent logic in the usual sense, and are not thought of or maintained that way. Developers can add or delete rules without having to think much beyond the empirical effect of an individual rule change as seen in large-scale testing. To be sure, rules presuppose and interact with other rules so the developer has to know what he or she is doing, but generally this does not entail elaborate reviews of logic. This is especially the case in rule writing for the hidden layers of **Res**. This freedom has resulted in a certain inelegance, but the very extensiveness of the rule base precludes the possibility of working in any other way. This untidiness in fact affords further grounds for likening the **Res** knowledge store to the neocortex which is also known to have memory cells that are redundant, that become inaccessible over time, and/or that may store error (e.g., word misspellings, mispronunciations, misconceptions, etc.), and so on. For a computer model, there is little to defend in such an arrangement beyond the fact that it works quite well: the error rate in homograph resolution achieved on previously unseen, unconstrained text averages 2% and, thus far, nothing has caused developers to believe these results are not still improvable, albeit marginally. Admittedly, such methodology will have little appeal to formalists, but one might reasonably argue that the method at least has the virtue of having yielded effective, industrial-strength machine translation. And it also has made the developers' job a good deal more fun.

Another cortical-like aspect of **Res** is that there is often more than one way to achieve a parsing objective, and slight differences between otherwise similar input strings may in fact cause considerable differences in the selection and sequence of firing rules. This comports with the view that human sentence processing is also subject to variation, both inter-personally, in the way different people process the same sentence, and intra-personally, in the way the same individual may process the same sentence in different contexts, or with only slight sentential variation.

7.2.3 Recurrent Circuitry in Res2 (Fig. 4)

When a **Res2** rule fires, three likely actions occur: (i) paths may be enabled and simultaneously other paths inhibited; (ii) SAL elements may be re-labeled in output vector **V3** or be provided with other notation; (iii) the sentential state array may be updated to reflect rule firing. This last action mimics recurrent circuitry in cortical circuits, where the firing of a neuron may cause modulation of an entire cortical region, increasing the potential of some neurons to fire and inhibiting others. Fig. 4 illustrates recurrent circuitry in **Res2** of the Logos Model.

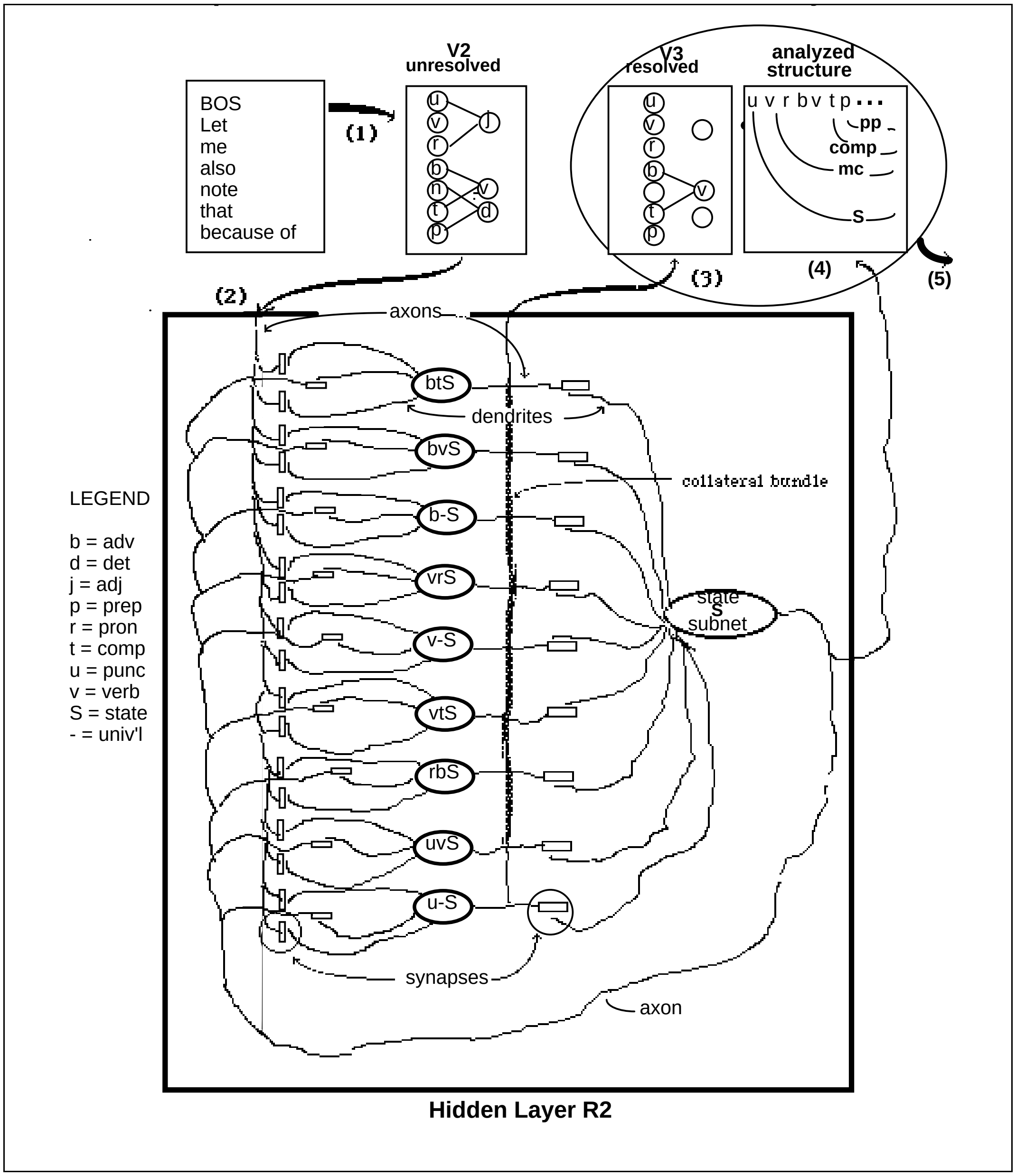


Fig. 4 - Recurrent Circuitry in Bionet Model. Res2 (**R2**) completes the work of homograph resolution and clause analysis begun in **Res1**. (Graphic illustrates the fact that the **Res2** module, over time, has become the most cortical-like component of the Logos Model.) In (1) we see the NL sentence, now as input vector **V2**, with still unresolved parts-of-speech. In (2), segments of **V2's** possible paths are submitted to hidden layer **R2**, units of which become active if their SAL specification matches and their state constraint (S) is satisfied. Units may be longer but typically entail only two or three SAL elements, i.e., enough to advance a path through **V2**. Active rules compete to fire, based on rule weights. A careful inspection of the cells will show how various rules have contributed (3) to a resolved path in **V3. V3** will serve as input vector to the micro-parse conducted by the **Parse** modules (5). As illustrated, rule firing also sends signals to a sentential state subnet, causing the state of the hidden layer to be altered. This altered state is communicated back to the entire net (recurrent circuitry), enhancing or inhibiting the firing potential of cells in the layer. The top-down picture of sentence structure thus acquired (4) then helps guide the micro-parse of the next layers (5).

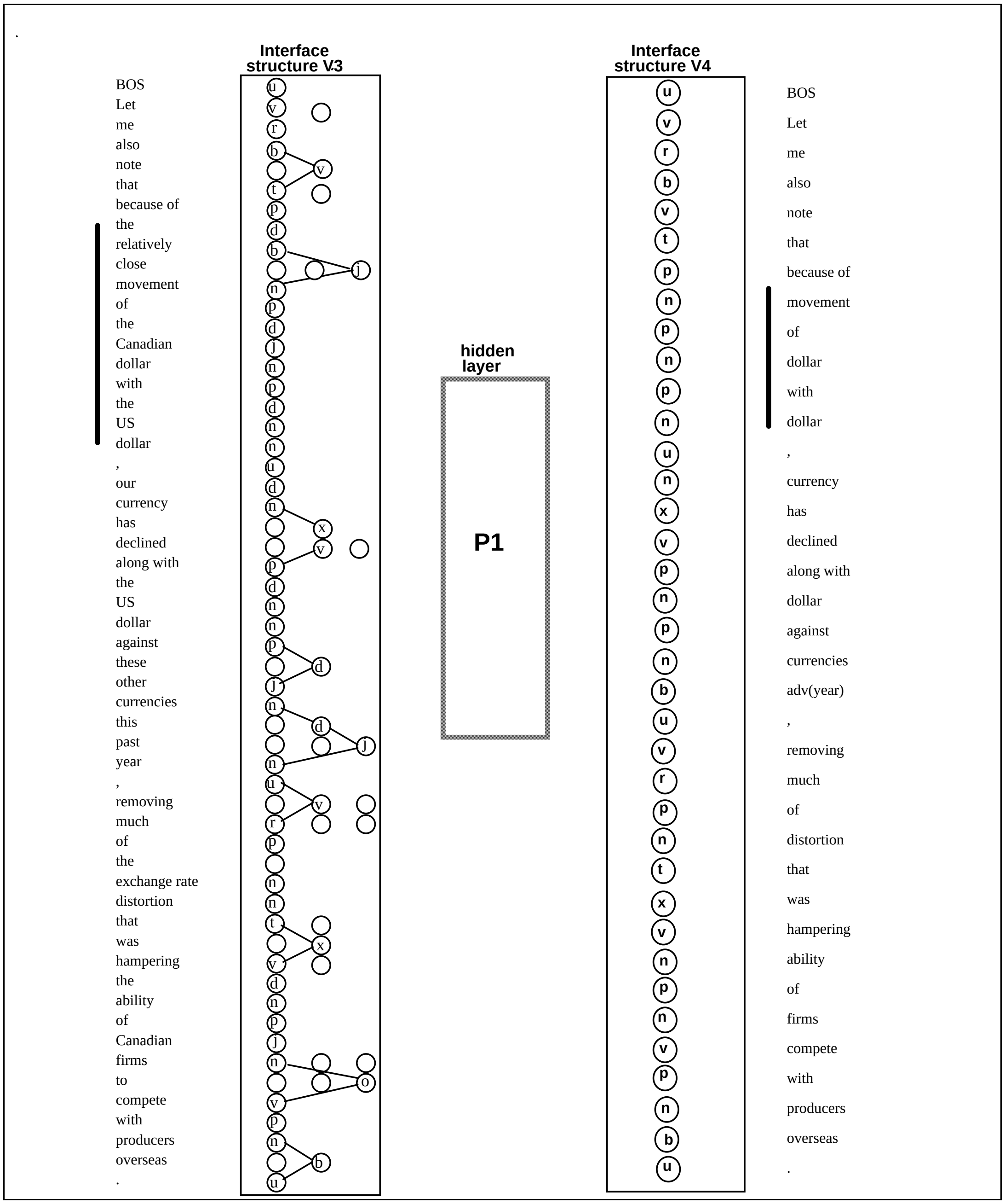


Fig. 5 - Bionet: Parse1. The **Parse1** module initiates the micro-parse of the sentence, drawing upon the results of the **Res** macro-parse output in **V3**. Notice how the complex noun phrase (highlighted by the vertical bar) is progressively handled over the sequence of **Parse** modules shown in Figs. 5-9. Here in **Parse1**, the simple noun phrase *the relatively close movement* is concatenated as **NP** with the semantics of the head noun, *movement*. In bottom-up parse fashion, output vector **V4** now serves as input to **Parse2.**

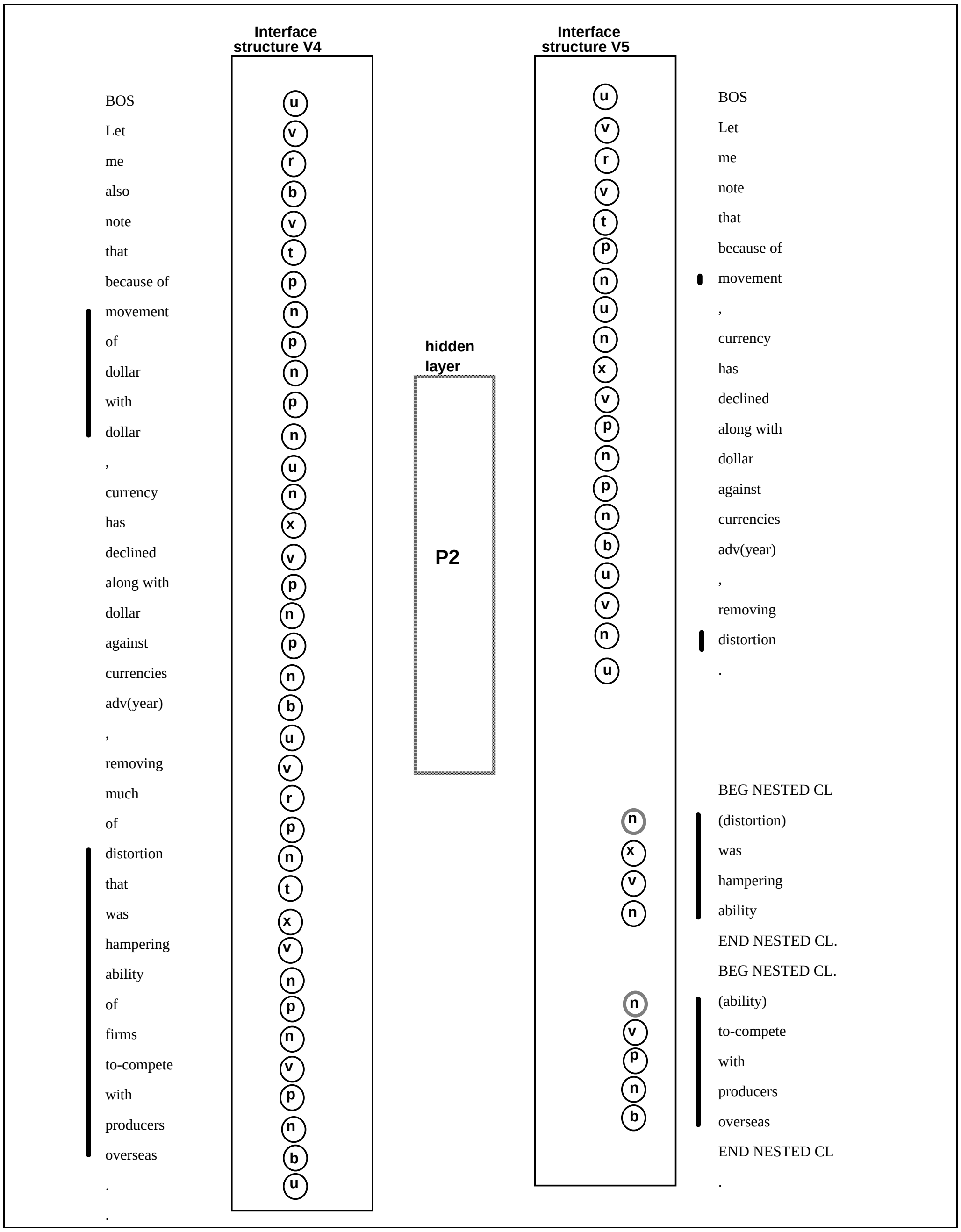


Fig. 6 - Bionet: Parse2. Note, in **V5,** how the long, complex noun phrase is now concatenated as **NP** with the semantics of *movement*. Also note how clausal embeddings have been extracted in order to simplify kernel sentence. Note that the subjects of the clausal complements (viz., *distortion* and *ability*) are repeated in the extracted segments. This is done in order to afford extracted materials all information needed for semantic analysis of clausal verb argument structure in **Parse3**. Any modification that **Parse3** analysis might make to the dummy subject is conveyed back to the true subject, including target transfers.

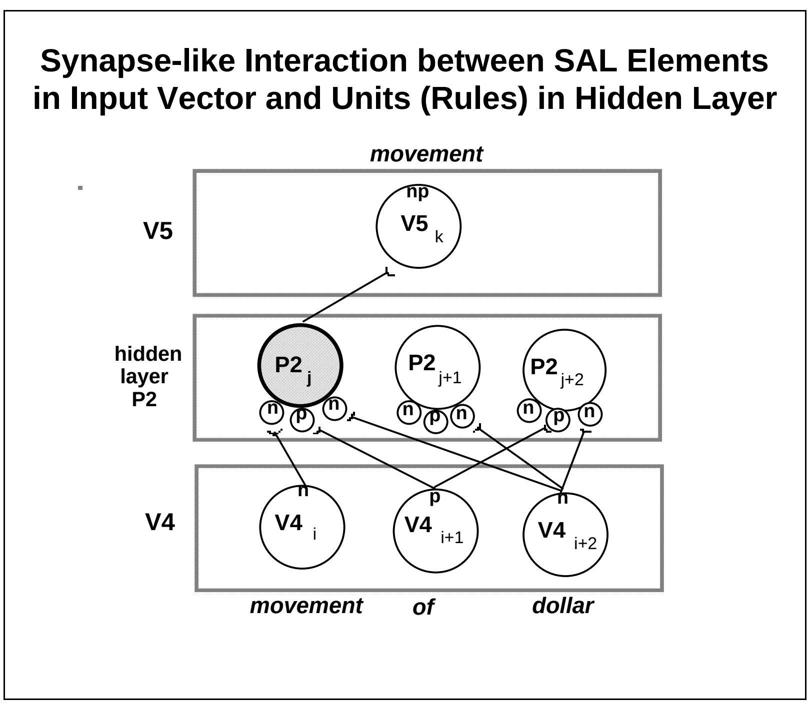
The sentential state array provides a summary, top-down picture of a sentence and its clauses, including embedded clauses, that has been gradually recorded during the **Res** macro-parse, affording top-down guidance to both the marco-parse and the subsequent micro-parse by the **Parse** modules. A simple example: when a **Res2** rule fires causing some ambiguous **N/V** SAL element to be resolved as **V**, the firing rule communicates this fact to the sentential state array. This automatically causes a change in the sentential state, the consequence of which being that any subsequent rule that might want to resolve ambiguous **N/V** elements in that clause to **V** is now inhibited from doing so. (Such inhibition of course would be qualified if the first resolved **V** were a SAL verb type that invites verbal complementation.) At the conclusion of the macro-parse, each element of output vector **V3** (input vector to **Parse1**) carries with it a compressed picture of the sentential state that held at the time the element was processed, viz., (i) type of sentence, (ii) type of clause the element is in, (iii) if an embedded clause, type of parent clause, (iv) SAL Type and tense of clausal verb, (v) SAL Type and number of clausal subject, etc.

7.2.4 Bionet: Parse1 (Fig. 5)

The output vector (**V3**) of **Res2** serves as input vector to the **Parse1** module. SAL elements in **V3** are now resolved with respect to part-of-speech (with some few exceptions, noted earlier and discussed below). **Parse1** does have the ability to reverse **Res** decisions regarding part-of-speech selection but this is rather rare. **Parse1** also makes use of the top-down picture of the sentence and clausal structures afforded by the **Res** macro-analysis. For example, a rule that sought to parse as **NP** the string **N CJ(CRD) N N** would have to satisfy, among other things, a rule constraint that the coordinating **CJ** not be a clause boundary. Macro-parse information about the clausal structure of a sentence enables the rule to test for this constraint.

Note that in **V4** (Fig. 5), **Parse1** has reduced all simple noun phrases to **NP**. Note also that **Parse1** recognizes and concatenates the phrase *this past year* as an **ADV** of time. Regarding the unresolved **AJ/N** ambiguity of the word *close*, **Parse1** will resolve this by forming a search argument of *close* plus the right-adjacent SAL noun element (*movement*) and then querying appropriate rules in **Semtab**. Absent some rule specifying a context that would dictate otherwise (e.g., *close procedure*), the default ruling for **AJ/N** in the adjectival position would be to **AJ**.

Fig. 7 - Detail of NP Formation in Parse2. In this Figure, a segment of short-term memory cells in input vector **V4** interacts with one of the long-term memory cells in hidden layer **P2** specialized for this SAL pattern. Rule constraints also have to be satisfied for rule to fire. Output in **V5** is rewrite of input pattern with semantics of head. **NP** node is annotated for **PP** complementation.



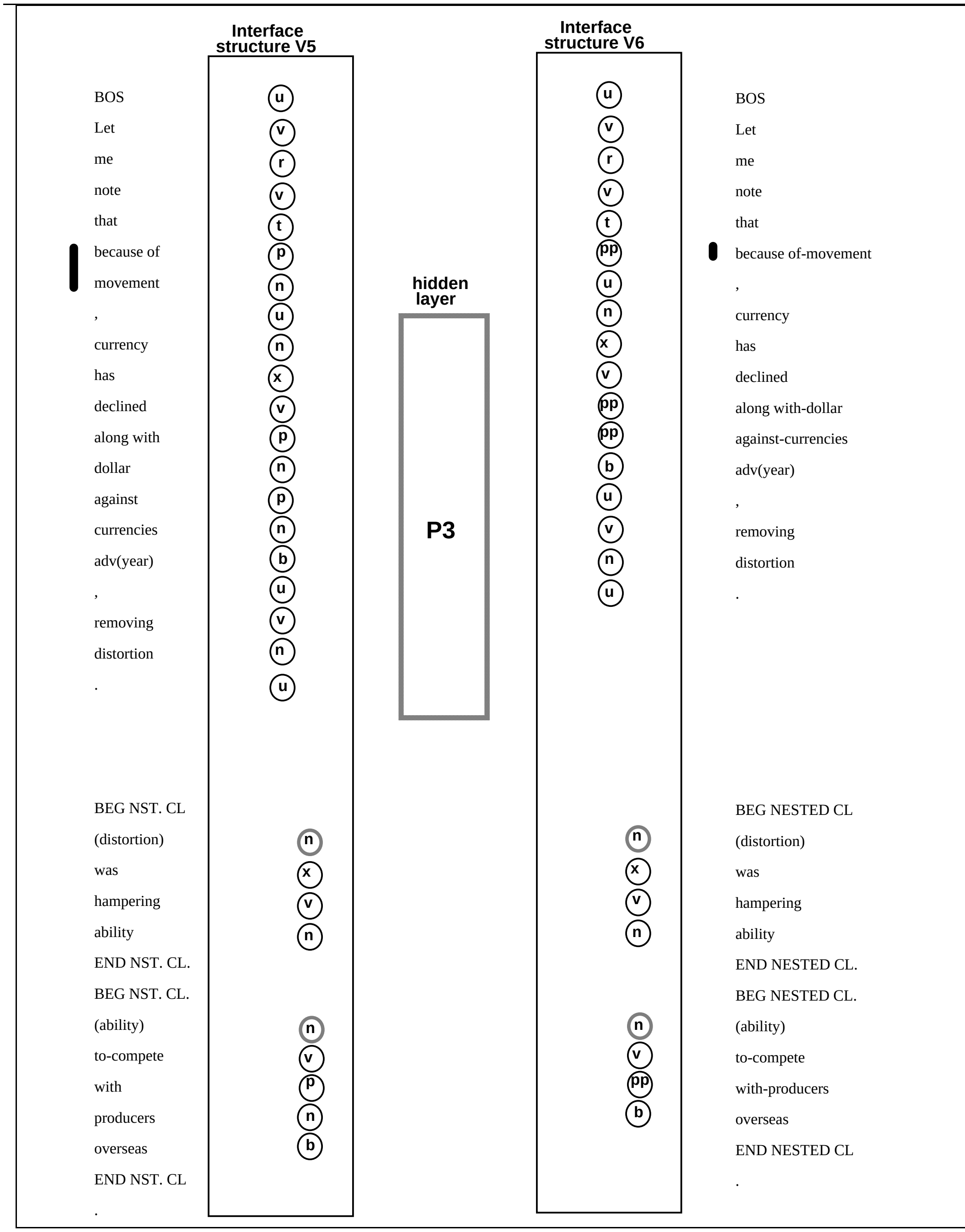


Fig. 8 - Bionet: Parse3. The graphic shows prepositional phrases being concatenated as **PP**. Other key **Parse3** functions include (i) analysis of grammatical relationships within a clause, and (ii) polysemy resolution of verbs and converbal prepositions, via interaction with the **Semantic Table**. **Parse3** also identifies and re-labels any as yet unanalyzed constituents, e.g., subjects, indirect objects, objects, and various types of adverbial **PP**'s.

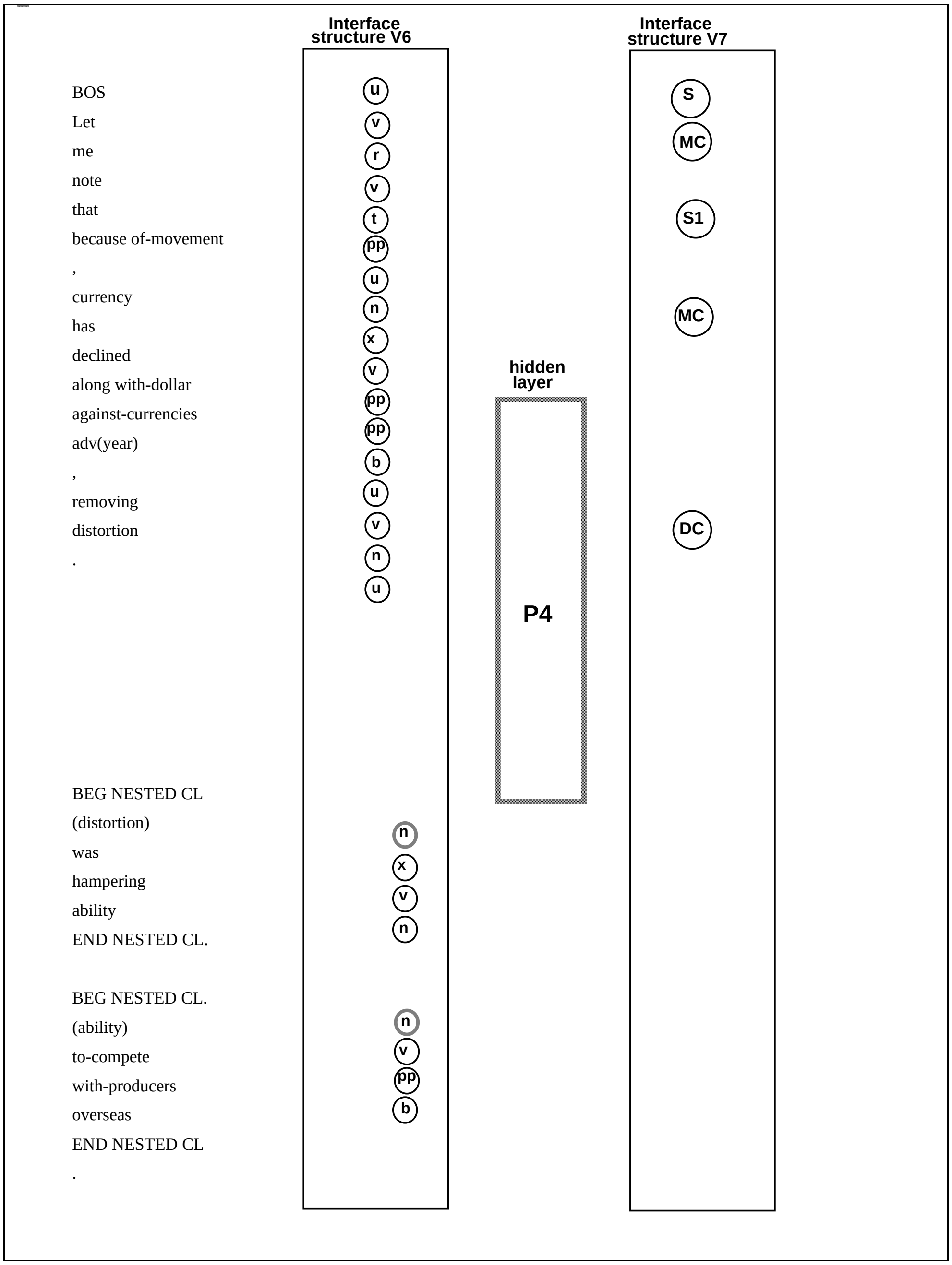


Fig. 9 – Bionet: Parse4. Source analysis is completed, materials at the end of the sentence are restored to their original places, and source constituents are submitted to linked target rules for structural transfer. (See *Logos System—Principles and Motivations* for discussion of target generation). Multi-language translations of the Example Sentence are given in 1.3.

1.2.5 Bionet: Parse2 (Fig. 6)

The **Parse2** module includes the following functions: (i) concatenation of simple **NP's** with their prepositional phrase complements; (ii) extraction of relative clauses and other clausal embeddings from their parent clause, leaving behind a trace linked to the extracted material. Extracted clausal matter is thereafter treated as separate (but not unrelated) sentences. This is done for two reasons: (i) to simplify the original sentence; (ii) to afford the extracted material benefit of complete parsing functionality. Push-down list methods allow handling of up to ten such embeddings.

In Fig. 7 we see illustrated the synapse-like way the input vector interacts with units in the hidden layer. The way rules are ordered and matched assures that rules of a more specific nature are always consulted first, before less specific rules. That is, rules whose index element is specified at the SAL subset level are looked at before rules specified at the set, superset or universal set level.

1.2.6 Bionet: Parse3 (Fig. 8)

The principal work of **Parse3** is to examine the main verb of each clause in relation to its clausal context. This is done by sending the SAL string containing the verb and its clausal context to the **Semantic Table (Semtab)** for analysis (pattern matching). Deep structure rules in **Semtab** seek match-up with the verb's argument structure, allowing (i) resolution of polysemy among the clausal constituents, especially verbs and prepositions; (ii) labeling of converbal and adverbial **PP's**; (iii) connecting of verbs with non-contiguous verb particles. Deep structure **Semtab** rules can be either word-specific or generic (based on SAL Type).

1.2.7 Bionet: Parse4 (Fig. 9)

Parse4 completes the analysis of the Example Sentence, reducing the sentence to its abstract clausal constituents, which together constitute **S**. The materials extracted in **Parse2** are now restored. In German source, some case ambiguities are not resolved until **Parse4**. For example, the case ambiguity of the two **NP's** in (36) must be resolved on the basis of agentiveness of one of the **NP's**. Output is unedited.

(2) *Dieses Garten liebt meine Mutter.*
(2') *My mother loves this garden.*

1.3 Multi-Lingual Translations of Example Sentence

The **Tgt Gen** module generates output by taking the instructions for transfer, word order and morphology that were established during analysis and the incremental transfer phase, and applying them now to literal target strings. Raw output of **Tgt Gen** is given below for French, German, Spanish, Italian, and Portuguese, the five target languages currently linked to English source. (Portuguese is the most recent of these and is least developed, as may be evident.)

(3) English Example Sentence

Let me also note that because of the relatively close movement of the Canadian dollar with the U.S. dollar, our currency has declined along with the U.S. dollar against these other currencies this past year, removing much of the exchange rate distortion that was hampering the ability of Canadian firms to compete with producers overseas.

French Output

Permettez-moi également de noter qu'à cause du mouvement relativement proche du dollar canadien avec le dollar américain, notre monnaie a décliné cette année dernière avec le dollar américain contre ces autres monnaies, en enlevant beaucoup de la distorsion de taux de change qui gênait la capacité des compagnies canadiennes de faire concurrence aux producteurs outre-mer.

German Output

Erlauben Sie mir auch, zu bemerken, dass wegen der relativ nahen Bewegung des kanadischen Dollars mit dem US Dollar, unsere Währung dieses letzte Jahr mit dem US Dollar gegen diese anderen Währungen gesunken ist, was viel von der Wechselkurs-Verzerrung entfernt, die die Fähigkeit kanadischer Firmen behinderte, mit Herstellern nach Übersee zu konkurrieren.

Spanish Output

Deje también observe que debido al movimiento relativamente cercano del dólar canadiense con el dólar estadounidense, nuestra moneda ha disminuido este año último junto con el dólar estadounidense contra estas otras monedas, retirando mucha de la deformación de tarifa de cambio que impedía la posibilidad de las empresas canadienses de competir con los productores a ultramar.

Italian Output

Fatemi notare anche che a causa del movimento relativamente vicino del dollaro canadese con il dollaro degli Stati Uniti, la nostra valuta ha declinato questo anno scorso con il dollaro degli Stati Uniti contro queste altre valute, togliendo molta della distorsione del cambio che impediva la capacità delle ditte canadesi di competere con i produttori all'estero.

Portuguese Output

Deixe observa me que devido ao movimento relativamente fechar do dólar canadiano com o dólar E.U., nossa moeda diminuiu este ano final juntamente com o dólar E.U. contra estas outras moedas, mudando muito da distorsão de taxa de câmbio que impediu a possibilidade das empresas canadianas de competir com os produtores do estrangeiro.

References

Anderson, J. A., J. Silverstein, S. Ritz, and R. Jones, 1977: 'Distinctive features, categorical perceptions, and probability learning: Some applications of a neural model', in *Psychological Review* 84:413-451, reprinted in J. Anderson and E. Rosenfeld (eds.) *Neurocomputing: Foundations of Research* (1989), MIT Press, Cambridge, MA., pp.287-326.

Anderson, James A. and M. C. Mozer, 1989: 'Categorization and Selective Neurons', in Geoffrey Hinton and James A. Anderson (eds.) *Parallel Models of Associative Memory*, Lawrence Erlbaum Associates, Hillsdale, New Jersey

Crick, F. H. C, and C. Asanuma, 1986: 'Certain Aspects of the Anatomy and Physiology of the Cerebral Cortex', in James McClelland and David Rumerlhart (eds) *Distributed Parallel Procession*, Vol. 2, MIT Press, Cambridge, MA, pp. 333-371 .

Beaven, J. L., 1992: 'Shake and Bake Machine Translation', in *Proceedings of the fifteenth (sic) International Congress on Computational Linguistics. COLING-92*, Nantes, France, pp 603-609.

Chomsky, Noam, 1957: *Syntactic Structures*, Mouton, The Hague.

---- 1965, *Aspects of the Theory of Syntax*, MIT Press, Cambridge, MA.

---- 1990: 'On Formalization and Formal Linguistics', in *Natural Language & Linguistic Theory*, Volume 8, No. 1, Kluwer Academic Publishers, Dordrecht, pp. 143-147.

Harris, Zellig,1968: *Mathematical Structures of Language,* John Wiley & Sons, New York.

Hinton, G. E., James McClelland and David Rumelhart: 1986: 'Distributed Representations', in David.Rumelhart and James McClelland (eds) *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge, MA, pp.77-109.

Hopfield, J. J., 1984: 'Neurons with graded response have collective computational properties like those of two-state neurons', *Proceedings of the National Academy of Science* 87:3088-3092, reprinted in James A. Anderson and Edward Rosenfeld, *Neurocomputing: Foundations of Research* (1989), MIT Press, Cambridge, MA., pp. 579-584.

Hutchins, W. J., 1986: *Machine Translation: past, present, future*, Ellis Horwood, Chichester.

Jackendorf, Ray, 2002: *Foundations of Languages*, Oxford University Press, New York.

Juola, Patrick, 1989: 'Machine Translation and Lojban', in *ju'i lobypli #8,* http://www.lojban.org/files/why-lojban/mactrans.txt. Consulted 22.10.2002

Karttunen, Lauri, 1987: 'Radical Lexicalism', *Technical Report,* Center for Study of Language and Information, Stanford.

Kohonen, T, 1982: 'Self-organized formation of topologically correct feature maps', in *Biological Cybernetics* 43: 59-69, reprinted in J. Anderson, and E. Rosenfeld (eds) *Neurocomputing: Foundations of Research* (1989), MIT Press, Cambridge.

McClelland, James, David Rumelhart, G. E. Hinton, 1986: 'The Appeal of Parralel Distributed Processing', in David.Rumelhart and James McClelland (eds) *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge, MA, pp. 46-76.

Magi, Ken, 1997: 'Response Selectivity and Neuron Doctrine', in *Austrian Society of Cognitive Science Technical Report 97-01*.

Shepherd, Gordon, 1994: *Neurobiology*, Oxford University Press, New York.

----- 1996, *Foundations of the Neuronal Doctrine*, Oxford University Press, New York.

Sherrington, Charles,1941: *Man on his Nature*, Cambridge University Press, Cambridge.

Scott, Bernard, 1989: 'The LOGOS System', *Proceedings of Machine Translation Summit II*, Munich.

-----1989: *Japanese to English Machine Translation* (incorporating talk and graphs by Bernard Scott*),* Report of a Symposium, Office of Japan Affairs, Computer Science and Technology Board, National Research Council, National Academy Press.

-----1990: 'Biological Neural Net for Parsing Long, Complex Sentences', Logos Corporation Technical Report; available from the author.

-----1998: 'Linguistic and Computational Motivations for the Logos Machine Translation System', Logos Corporation Technical Report, available at http://www.iai.uni-sb.de/iaien/iaiwp/p11/index.html.

---- 2000: 'Logos System -- Principles and Motivations.' Logos Corporation Technical Report; available from the author.

Somers, H. L., 1992/3: 'Current Research in Machine Translation', *Machine Translation 7*, 231-246.

Thorpe, Simon, 1995: 'Localized versus distributed representations', in M. A. Arbib (ed.), *The Handbook of Brain Theory and Neural Networks,* MIT Press, Cambridge, pp.549-552.

Yngve, V. H., 1964: 'Implications of mechanical translation research', *Proceedings of the American Philosophical Society* 108, 275-281, reprinted in H. E. Bruder, *Automatische Sprachuebersetzung*, Wissenschaftliche Buchgesellschaft (1982), Darmstadt, pp. 33-47.