Encyclopædia of Cognitive Science

Connectionist and Symbolic Representations of Language, #292

Mark Steedman
Professor, Division of Informatics
University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW
Scotland, UK

Tel: (0)131 650 4631
Fax: (0)131 650 6626
Email: steedman@cogsci.ed.ac.uk

Connectionist and Symbolic Representations of Language

Keywords: neural computation; grammar; language acquisition; language processing; grounded conceptual structure.

Contents

Connectionist or neural computational representations are based on numbers of multiplely connected identical neuron-like parallel processing units, the connections bearing modifiable weights adjusted on the basis of locally available information using a learning algorithm. Symbolic representations, by contrast, are defined in terms of rules relating expressions in a formal language. Among the claims that have been made for the former class of models is the "emergence" of generalizations that had been claimed to require the mediation of rule-based grammars and modular symbolically-represented processing architectures. A more convincing linguistic role for neural computational networks lies in their potential for inducing grounded conceptual structure and statistical models as infrastructure for acquisition and processing of standard symbolist representations of lexicalized syntax and semantics.

1 The Approaches

Connectionist theories of the parallel distributed processing (PDP) variety (Rumelhart et al. 1986) begin from the reasonable belief that phenomena of mind are the result of computation in richly interconnected networks of neurons or neuron-like units. They embody the hypothesis that the nature of this computational device critically determines the nature of the computation itself. Symbolic theories begin from the equally reasonable belief that such phenomena of mind as language use, understanding, and reasoning are symbolic in nature, in much the same sense as mathematical and logical inference are, and that as such the computation involved can be characterized independently of any specific device, whether parallel-distributed or not, that implements it. The connectionist approach is by nature reductionist (in the best sense of a much abused term) in that it attempts to generalize from low-level mechanisms to higher-order phenomena. The symbolic approach is phenomenological (again in a positive sense) in that it tries to work in the opposite direction, from a high-level description to the implicit underlying mechanism. These approaches are clearly compatible, and can coexist. As in any area of scientific research, both approaches are necessary, and the interesting question to ask is to which particular problems each is best suited, and where they meet up and can be unified.

In our present state of knowledge in this rapidly evolving field, the answers to these questions are far from clear. Neural Computational mechanisms have proved their worth in the field of pattern recognition and classification, where it is clear that they can extract structure latent in inputs such as images of faces, hand-written letters, and speech, and embody that structure in recognizers that would be impossible to specify by hand, or that are orders of magnitude more efficient than rule-based mechanisms, even when these are statistically optimized. On the other hand, except where they have been used to explicitly simulate the structure of a symbolic parser and associated devices such as the push-down automaton (PDA), these same devices have proved much less clearly successful in demonstrating the kind of recursive productivity (discussed below under the heading of "systematicity") that rule-based symbolic systems are good at, or in supporting semantic interpretation and inference.

In contrast, rule-based discrete symbolic systems express productivity or systematicity, semantic interpretation, and processes of inference immediately. However, it has proved very difficult to build rule-based linguistic or computational linguistic systems with coverage on the scale characteristic of human linguistic and reasoning abilities, and in recent years such systems have increasingly relied upon machine learning and statistical optimization techniques, of a kind quite closely related in mathematical terms to neural computational techniques. It seems likely that there is everything to be gained from combining these approaches.

In order to compare the approaches, it is helpful to recall that symbolist theories distinguish a number of distinct components to language processors. One fairly generally applicable architecture distinguishes: i) a *grammar*, characterized by syntactic and semantic rules of certain classes and a related characteristic automaton; ii) a nondeterministic *algorithm* characterized by properties such as the order in which rules are applied to the string, whether bottom-up or top-down, the order in which the words of the string are examined, whether first-to-last or otherwise, and by certain memory resources, such as those used in building structure and the charts or tables used in parsers based on dynamic programming, and iii) an *oracle* or decision criterion for rendering the algorithm deterministic and deciding which rule(s) to apply in cases where there is more than one possibility.

In any given theoretical presentation or implementation, these modules may be combined, but in rule-based theories they can usually be distinguished in functional terms. The fact that they are in that sense distinct modules does not of course imply that the corresponding computations must be carried out in series, in chronologically separate phases: for example it is quite possible to construct systems in which the oracle can call on the results of semantic interpretation and contextual reference in mid-sentence, while syntactic analysis is still under way.

It is important to be clear on this last point, because the fact that phenomena like “garden path” effects in parsing can be affected by semantics and even by extrasentential context is often adduced as evidence in specific support of PDP models and against symbolic ones. However, parallel-constraint satisfaction drawing on multiple knowledge sources is commonplace among rule-based models of sentence processing, and is acknowledged by Fodor (1983, p.78, fn. p.134-135) to be entirely modular.

It follows by the same token that Connectionism is not intrinsically any less modular than any other approach. Nevertheless, there has been a considerable emphasis in the connectionist literature on the idea that the appearance of rule-like behavior is “emergent” in such systems, and that language processing and language acquisition can be modeled in monolithic non-modular PDP machines or algorithms without the explicit involvement of grammars.

In assessing the connectionist claims it is important to ask what it is that neural computational machinery actually learns, and whether it can in principle do the jobs that language does for us. In particular, we must ask whether it will deliver meaning in a form that will in turn support logical inference. In practice, inference systems of any generality have generally depended on explicitly representations of structure of some kind.

In investigating these questions, two specific claims are of particular interest. The first is the claim that grammars in the sense that symbolists understand the term are an emergent phenomenon of the learning of sequential dependencies by recurrent neural networks with “contextual” units. The second is the claim originating with Niklasson & van Gelder (1994) that structure can be represented in distributed memory and manipulated systematically (in the sense of that term that Fodor & Pylyshyn (1988) claimed to intrinsically require symbolic representation), without explicitly representing the pointer structure of the symbolic representation. Clearly if both of the claims are correct then the neural computational account has gone a long way towards delivering a distinctively non-symbolic account of language.

2 Networks and Grammars

2.1 Simple Recurrent Networks

The Simple Recurrent Network (SRN, Elman 1990) approximates the more costly but exact “backpropagation through time” algorithm of Rumelhart et al. (1986) for learning sequential dependencies. It does so by using a single set of

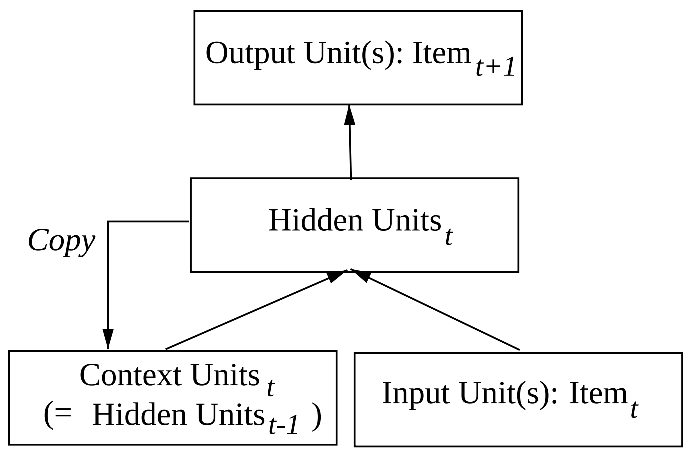

Figure 1
Simply Recurrent Network (SRN)

context units which store the activations of the hidden units at time $t-1$ as an input to the hidden units at time t, along with the activations of the normal input units corresponding to the current item, $item_t$, as in figure 1. Since the activations of the hidden units at time $t-1$ were themselves partly determined by the activations on the hidden units at time $t-2$, which were in turn determined by those at time $t-3$, and so on, the context units carry ever-diminishing echoes of ever more distant items in the preceding sequence.

Because there is no clear bound to the extent of the preceding sequence about which information can be captured in the context units, it is not entirely clear what is the precise automata-theoretic power of such “graded state machines”. (see Cleeremans et al. (1995)). However, the signal-to-noise ratio for information concerning distant items falls off very rapidly with this mechanism, and it is fairly clear that in practice SRNs of the kind that can actually be built and trained end up approximating the class of Finite State Markov Machines that can be learned using the exact technique to a degree of accuracy that depends on the maximum number of timesteps required.

Finite state machines are interesting devices, and it is often surprising to see the extent to which they can approximate the output of devices that are intrinsically higher on the automata-theoretic hierarchy. It is interesting to ask whether similar mechanisms play any part in natural language processing.

Such results are possible because some neural network algorithms are capable of inducing extremely efficient—and correspondingly opaque—representations, when compared with standard Hidden Markov Models (HMM). However, as SRNs are actually used by psychologists and linguists they appear to approximate something much closer to a familiar standard symbolist finite-state device, namely the *n*-gram part-of-speech (POS) tagger. (This also seems to be their role in "hybrid architecture" connectionist parsers which use networks to implement a push-down stack and structure-building modules in a more standard parser architecture.)

2.2 Finite-state Part-of-speech Tagging

N-gram POS tagging—that is, the determination of the form-class of ambiguous lexical items like *bear* on the basis of sequential probabilities at the word level can be remarkably successful in reducing the degree of nondeterminism that practical parsing algorithms must cope with (accuracies over 97% are standard). Moreover, there is growing evidence that if the standard Brown Corpus POS categories like VB are replaced with the more informative lexical categories that are used in lexicalized grammars such as Lexical Functional Grammar (LFG), Tree Adjoining Grammar (TAG), Head-driven Phrase Structure Grammar (HPSG) and the various forms of Categorial Grammar (CG), and if different senses of the same syntactic type are also distinguished as different lexical items, this effect may do a great deal of the work of parsing itself, leaving only structural or "attachment" ambiguities to be resolved by parsing proper.

2.3 Why do SRNs and Part-of-Speech Taggers Work?

Finite-state POS taggers and by assumption SRNs work reasonably well on tasks like category- and sense- disambiguation and prediction of succeeding category because the implicit Markov processes encode a lot of the redundancy (in the information-theoretic sense of the term) that is implicit in grammar, interpretation, and world-knowledge. This means that, like standard Markov processes, SRNs can be made the basis of quite good predictors of processing difficulty. For example, the SVO word-order of English and our knowledge of the world between them determine the fact that the transitive category for the word *arrested* is more likely to follow the noun *cop* than the past participial category, while these preferences are reversed following the word *crook*.

(1) a. The cop arrested by the detective was guilty.
    b. The crook arrested by the detective was guilty.

These likelihoods are reflected in increased reading times for human subjects at the word *by* in (a) as compared with (b), for example.

2.4 Are Grammars Emergent from SRNs?

While claims exist in the literature to the effect that recognizers for stringsets of kinds that in general require grammars of higher than finite-state power have been acquired by SRNs (Christiansen & Chater 1994), none of these results suggests that the grammars in question are therefore "emergent" properties of mechanisms like SRN, any more than they are of n-gram POS taggers. Although the context defined by the context units is in a limited sense unbounded, and SRNs can in theory be used to model long distance agreement dependencies, because of already-noted properties of the context unit representation, reliability falls off with distance, and these dependencies cannot be regarded as unbounded in the technical grammatical sense. They should instead be regarded as a finite-state cover of the higher-power grammar to some limited maximum string-length. This means that claims to model human language acquisition using SRNs must be treated with some caution, although such models raise developmentally interesting questions. (Interestingly, there are conflicting claims by Elman et al. (1996) and by Rohde & Plaut (1999) as to whether SRN learning depends on "starting small," ordering presentation of simple examples before complex ones, or is rather inherently biased towards acquisition of local dependencies before long-range dependencies, and therefore independent of the order in which training examples are presented.)

Even within these limits, error-free sequences of grammatical categories fall short of semantic interpretability, as can be seen from the fact that the following sequence has *two* interpretations based on identical Brown corpus categories:

(2) Put the block in the box on the table.

Although SRNs can be regarded as disambiguating lexical items, this other kind of ambiguity—structural or attachment ambiguity—remains, as in the case of POS taggers.

For the same reason, it does not seem legitimate to regard "trajectories" through the high-dimensional space defined by the hidden units as the equivalent of parses or interpretation (Tabor et al., 1997). Many other defining properties of interpretations—such as the ability to support the kind of structure-dependent transformations characteristic of inference—seem to be lacking in trajectories or category sequences of this kind. To find such properties in neural computational

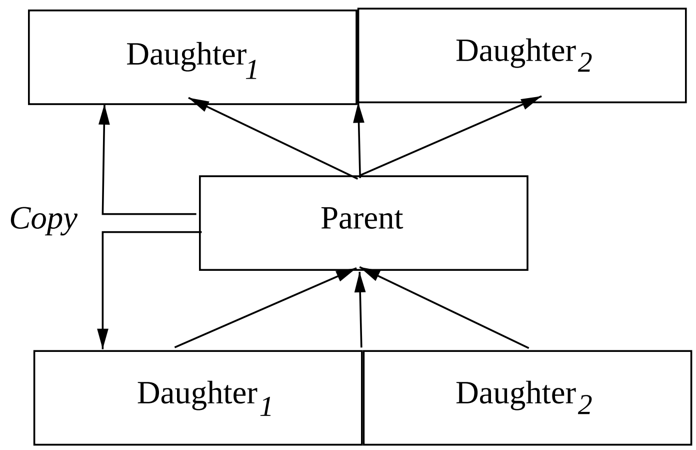

Figure 2
Recursive Auto-Associative Memory (RAAM)

representations, we need to look to devices other than SRN.

3 Interpretable Structure and Associative Memory

Much neural computational work has explicitly or implicitly taken on board the need for explicitly representing the equivalent of trees or pointer structures to represent syntactic or semantic analyses (see papers in Hinton 1990a) using associative memories of various kinds.

Such devices are of interest for (at least) two reasons: First, they inherit some psychologically desirable properties of distributed representations, such as content-addressability and graceful degradation under noise and damage. Second, they offer a way to think about the interface between neurally embedded map-like sensory-motor inputs and outputs, and symbolic knowledge representation.

3.1 Recursive Auto-Associative Memory (RAAM)

Recursive Auto-Associative Memory (RAAM, Pollack 1990) is a device that uses hidden unit activation patterns to store associations between input and output patterns. It is called “auto-associative” because it uses the same patterns as input and output.

An n-ary recursive structure can be stored bottom-up in the RAAM starting with the leaf elements by recursively auto-associating vectors comprising up to *n* hidden-unit activation patterns corresponding to either leaves or previously encoded structures. The activation pattern that results from each auto-association of the daughters can then be treated as the address of the parent.

Since by including finitely many further units on the input and output layers we can associate node-label or content information with the nodes, a modification sometimes referred to as Labeled RAAM (LRAAM), this device can store recursive parse structures, thematic representations, or other varieties of logical form of sentences.

The device should not be confused with a parser: it is trained with fully articulated structures which it merely efficiently stores. The hidden units can be regarded as encoding to some approximation the context-free productions that defined those structures, in a fashion similar to the way Hinton (1990b) encoded part-whole relations, enabling recognition of pattern instances which have not been encountered before. In that sense the device has been claimed to be capable of inducing the corresponding grammar from the trees (Pollack, 1990), contrary to the claims of Fodor & Pylyshyn (1988) concerning the systematicity or generalizing capacities of neural networks.

This claim was challenged by Hadley (1994a), who extended Fodor and Pylyshyn's critique to distinguish a number of levels of systematicity in the induced classifier, contingent on the relation of the test examples to the training set. The system of levels of systematicity was further refined by Niklasson & van Gelder (1994) who among other levels distinguished: Level 3, generalization to all and only legal seen structures with novel constant/position pairings; Level 4, generalization to all and only legal novel embeddings of seen structures with seen constant/position pairs; Level 5, generalization to all and only legal novel embeddings of seen structures with novel constant/position pairings—where "legal" means permitted by the corresponding rule-based system.

3.2 Scope and Limits of Systematicity

Niklassen and van Gelder's 1994 experiment had the novel feature of representing both formula-like structures and (via a separate network) operations over those representations resembling logical equivalence transformations, such as the rule that replaces formulae of the form $P \rightarrow Q$ with a corresponding formulae $\neg P \vee Q$, or syntactic transformations, such as the rule that relates sentences of the form *cat chase dog* to *dog chased by cat*. Niklassen and van

Gelder were able to demonstrate level 3 systematicity on this task (although Hadley (1994b) expresses some reservations about statistical reliability). Remarkably, these operations worked without explicitly unpicking the representation, via pointer-dereferencing of the kind standard in symbolic representations of logical formulae and their transformation using list-processing computer programming languages like LISP and Prolog.

This is probably a more appropriate use for RAAM than building parse trees, since RAAM is slow to train, and inherits poor scaling properties from its use of backpropagation. Devices such as the Holographic Reduced Representations (HRR) proposed by Plate (1994) are an interesting alternative. Their properties for the storage and holistic transformation of such structures has been investigated by Neumann (2001), who reports replication of Niklasson & van Gelder (1994) using RAAM and an extension to their level 5 systematicity (generalization to novel embeddings of seen structures with novel constant/position pairings) for a related system using HRR.

Neumann shows that these networks can learn superimposed collections of linear transformational rules that depend on the relation of isotopy which holds between the distribution of their inputs and that of their outputs in the hyperspace defined by the weights. (This space can be examined using Principal Components Analysis (PCA) and other clustering techniques. This representation of whole disambiguated structures as points in a high dimensional space is unrelated to the SRN representation of sequences as trajectories in such a space.) Among the rules that Neumann shows can be learned in this way is one mapping $P \wedge (P \Rightarrow Q)$ onto Q. This rule is related to the rule of *Modus Ponens,* the cornerstone of any practical inference system. It therefore seems possible in principle that structures represented in this way could support inference.

However, some caution is in order in interpreting these results, as Neumann points out. To build a practical inference engine to exploit this property requires a number of further steps, including the identification of suitable *pairs* of formulae of the form P and $P \rightarrow Q$ from a larger set to form the input to the rule, and a process of *search* for proofs through a potentially exponentially growing space of sequences of inference steps. It is simply unclear whether such processes can be helpfully thought of in distinctly neurocomputational terms, or whether this is the level at which at least part of cognition becomes distinctively symbolic.

4 Using Classifiers to Learn Grounded Conceptual Categories

One promising use of associative memory models like RAAM and HRR might be to learn the bounded syntactic/semantic structures that are associated with lexical items, particularly verbs, which provide the input to lexicalized grammars and parsers of the kinds referred to earlier.

We might assume that a subset of such structures is available prelinguistically, and result relatively directly from the evolved or learned structure of connections to the sensorium, short term memory, and the like. At higher levels, such structure may arise from non-linguistic network concept-learning along lines set out by Hinton (1990b), without mediating symbolic forms.

Part of the interest of this proposal lies in the possibility that the interaction of such structured sensory-motor manifolds and this novel form of concept learning might give rise to "grounded" conceptual categories within a standard symbolist approach. Grammar acquisition would then mainly reduce to the association of lexical items to concepts, and decisions such as whether the syntactic type corresponding to the *walking* concept looks forwards, backwards, or either way, for its subject in the particular language that the child is faced with, and how the multiple arguments of transitives, ditransitives, and the like map onto the underlying universal logical form, as reflected for example in the possibilities for relexivization. Since directionality can be represented as a value on an input unit, and since an individual category can be defined as a finite state machine, and its mapping to universal logical form can be captured as a finite state transduction, such categories are good candidates for learning with neurally computational devices.

A similar tendency towards lexical involvement is evident in current statistical computational linguistic research. Much work in probabilistic parsing including recent proposals by moves away from autonomous Markovian POS tagging and prefiltering, and towards a greater integration of probabilities with grammar—see Manning & Schütze (1999) for a review.

Part of the interest of this proposal lies in the possibility that such learning might capture word-order generalizations over the lexical categories. Certain constraints that have been discussed within Optimality Theory (Prince & Smolensky 1997), such as the tendency for semantically related categories (such as tensed transitive verbs) to have the same directionality (such as SVO order) are "soft," in the sense that they can have exceptions (such as English auxiliary verbs). It seems likely that the associative memory -based lexical acquisition device sketched above might also be suited to acquiring such soft-constraint-based lexicons. If so, then the claim that the form of possible human lexicons was "emergent" from the neural mechanism would have real force.

In this connection it is interesting to note that monolithic PDP-based connectionist models have been quite successful in modeling the acquisition and processing of systems of morphologically inflected lexical items that mix regular and irregular forms, such as the English past tense system (first approached by Rumelhart & McClelland 1986), competing successfully if not conclusively with rule-and-exception based models (Pinker 1999) in accounting for the course of acquisition in children, including the "U-shaped curve" in frequency of correct vs. incorrect use of irregulars, where children initially use forms like "ran", then drop them in favor of overregulations like "runned", before returning to using "ran".

5 Implications for Nativist Theories of the Language Faculty

Given the origins of the term "connectionism" in the behaviorist theories of Thorndike and others as modified by Hebb, it is perhaps not surprising that its advocates see themselves as in conflict with the nativist position associated with Noam Chomsky, the opponent of behaviorist theories of language and founder of modern symbolic approaches to language.

The conflict, which has been most eloquently pursued by Elman et al. (1996) and in the response by Marcus (2001), is more apparent than real. Chomsky's point has always been that attempts to explain the universal form of language and the course of language acquisition in terms of more general-purpose cognitive faculties or psychological laws have *in practice* not been notably effective. While constantly deriding the inability of currently available theories of learning, cognitive development, or semantics to make any significant contribution to linguistic explanation, and advocating the study of innate universal principles of language in isolation from other aspects of cognition, he has consistently advanced this position as a matter of methodological expediency. The fact remains that the only remotely plausible source for the innate component lies in the conceptual structure with which the child comes to language learning, and which either evolved or was learned for more general cognitive purposes, an observation which is implicit at least as early as Chomsky (1965) (see section I.5) and is explicit in Pinker's early work on modeling acquisition.

Elman et al. 1996 does a very good job of demolishing certain very dubious claims for evidence of specifically linguistic genetic components. (In particular, its review of the unseemly rush to overinterpret the significance of the heritable disorder of the KE family is very telling.) However, if the burden of the present review is correct, the contribution of neural computational theories to our understanding of language is unlikely to be to demonstrate the emergence of grammar from monolithic neural computation. Machines that are emergent in this sense are intrinsically implausible as psychological models, because they offer the same kind of obstacles to further evolutionary development that unstructured programs offer to human developers, as Holland (1998) points out in different terms. The distinctive contribution of neural computational models is more likely to lie in explaining how the structure implicit in the sensory manifold and our interactions with the world can be extracted by classifiers based on algorithms like back-propagation, the restricted Boltzmann machine learning algorithm, and the like, to provide the grounded conceptual structure upon which both reasoning about the world and the development of language depend. The question of how much of this conceptual structure is actively learned by the individual prelinguistic child, and how much of it has been compiled into heritable hard-wired components during the process of evolution of humans and their animal ancestors, as well as the question of what further apparatus is needed for the development of language and whether its origins can also be traced to more generally useful cognitive abilities, remain open. (One promising but in formal terms under-investigated source for the latter apparatus that is suggested by both developmental and neuroanatomical evidence lies in the system for planning action in the world.) These will be questions of considerable empirical interest to both symbolic and neural-computational theorists, to which both will doubtless continue to make distinctive contributions.

References

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

Christiansen, M. & Chater, N. (1994). Generalization and connectionist language learning. *Mind and Language*, *9*, 273–287.

Cleeremans, A., Servan-Schreiber, D., & McClelland, J. (1995). Graded state machines: The representation of temporal contingencies in feedback. In Y. Chauvin & D. E. Rumelhart (eds.), *Backpropagation : Theory, Architec-*

tures, and Applications, Developments in connectionist theory, pp. 274–312. Hillsdale, NJ: Lawrence Erlbaum Associates.

Elman, J. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Elman, J., Bates, E., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., & Plunkett, K. (1996). *Rethinking Innateness: A Connectionist Persective on Development*. Cambridge, MA: MIT Press.

Fodor, J. (1983). *The Modularity of Mind*. Cambridge, MA: MIT Press.

Fodor, J. & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: a critical analysis. *Cognition*, *35*, 183–204.

Hadley, R. (1994a). Systematicity in connectionist language learning. *Mind and Language*, *9*, 247–272.

Hadley, R. (1994b). Systematicity revised: Reply to christiansen and chater and niklasson and van gelder. *Mind and Language*, *9*, 431–444.

Hinton, G. (ed.) (1990a). *Connectionist Symbol Processing*. Cambridge, MA: MIT Press/Elsevier. Reprint of *Artificial Intelligence, 46:1-2*.

Hinton, G. (1990b). Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, *46*, 47–75. Reprinted in Hinton (1990a).

Holland, J. (1998). *Emergence : From Chaos to Order*. Reading MA: Perseus.

Manning, C. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Marcus, G. (2001). *The Algebraic Mind: Integrating Connectionism and Cognitive Science (Learning, Development, and Conceptual Change)*. Cambridge, MA: MIT Press.

Neumann, J. (2001). Holistic processing of hierarchical structures in connectionist networks. Ph.D. thesis, University of Edinburgh.

Niklasson, L. & van Gelder, T. (1994). On being systematically connectionist. *Mind and Language*, *9*, 288–302.

Pinker, S. (1999). *Words and Rules: The Ingredients of Language*. New York: Basic Books.

Plate, T. (1994). Distributed representations and nested compositional structure. Ph.D. thesis, University of Toronto.

Pollack, J. (1990). Recursive distributed representations. *Artificial Intelligence*, *46*, 77–105. Reprinted in Hinton (1990a).

Prince, A. & Smolensky, P. (1997). Optimality: From neural networks to universal grammar. *Science*, *275*, 1604–1610.

Rohde, D. & Plaut, D. (1999). Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, *72*, 67–109.

Rumelhart, D. & McClelland, J. (1986). On learning the past tenses of english verbs. In J. McClelland, D. Rumelhart, & the PDP Research Group (eds.), *Parallel Distributed Processing,* Vol. 2*, Psychological and Bilogical Models*, pp. 216–271. Cambridge, MA: MIT Press.

Rumelhart, D., McClelland, J., & the PDP Research Group (eds.) (1986). *Parallel Distributed Processing*, vol. 1. Cambridge, MA: MIT Press.

Tabor, W., Cornell, J., & Tanenhaus, M. (1997). Parsing in a dynamical system: An attractor-based account of the interaction of lexical and structural constraints in sentence processing. *Language and Cognitive Processes*, *12*, 211–271.

Glossary

Connectionism: The term appears to originate with the american behaviorist Thorndike, who used it to refer to theories based on graded strength of association between stimulus and response. It is first used in its modern sense (somewhat in passing, and with explicitly dissociation from the notion of *S-R* pairs) by Hebb, in *The Organization of Behavior,* to refer to the use of multiplely connected identical neuron-like units with modifiable weights on connections, adjusted on the basis of locally available information by a learning algorithm (although Hebb' theory itself was not fully computationally explicit and his learning algorithm is rarely used in its original form). The term came later to be associated with the PDP school of neurocomputational models whose comparison with symbolic models is the main concern here. More recently, the term has become less used in computational circles, and is currently more associated with psychological modeling and philosophical speculation concerning the nature of mind.

Grounding (of conceptual representations): A mental concept (or its computational equivalent) is grounded if it is defined at the lowest level in terms of sensory-motor inputs from, and outputs to, the physical world. Since it is likely that the nature of natural concepts is distinctively shaped by the fact that they are grounded, unlike the knowledge representation in a standard database or expert system, a major attraction of neural computation is that it supports such grounding directly, for example by direct embedding of inputs and outputs in the neural sensory-motor arrays.

Isotopy: Two figures are isotopic if one can be transformed into the either solely by processes of stretching and compression along the dimensions of the figure. For example, a figure drawn on a sheet of rubber is isotopic to all figures that can be obtained by pulling and poking but not cutting it.

Systematicity: The property of being able to process all and only examples of a given, usually infinite, class of inputs correctly, subject only to limitations of time, memory and the like, also sometimes referred to as "productivity" or "generativity". Since humans exhibit systematicity with respect to a number of domains, and in particular to the sentences of their native language, a major attraction of symbolic systems like generative grammars is that they immediately exhibit systematicity with respect to the stringset or language that they define.

Nativism: The hypothesis that a process of cognitive development requires the mediation of an innately provided mechanism. The frequently cited result due to Gold, who showed that not even the class of finite-state languages can be learned from the mere exposure to positive instances of strings, is often derided as irrelevant in the face of the fact that statistical approximations for many interesting classes can provably be learned on this basis. Nevertheless, the problem of actually carrying out such induction scales very badly. In fact it is certain that learning systems as complex as natural grammars involves an innate component.

The only question is what the nature of that component is (to which the answer will almost certainly turn out to be “the semantics,” or rather the conceptual representation that underpins it), and how specialized to language as distinct from the rest of cognition it is (to which the answer will almost certainly turn out to be “not very”).

Emergence: The property of getting something for nothing from a representation. The form of a crystal or a snowflake is emergent from thermodynamics and the form of the molecules that make them up, in the sense that the low level description of the system does not represent properties like “octahedral’ or “six-fold symmetric”. The latter are descriptions of a quite different kind, involving interactions with the world at a different scale. Emergence is a property crucial to the evolution of complex organisms and behaviors, in the sense that variation proceeds by variation at the molecular level, but natural selection operates at the level of interactions with the world. However, such a combination of variation and selection defines a search problem that can be analyzed in terms familiar from automated game-playing. If all variation proceeds by random change at the level of individual base-pairs on the DNA molecule, whether by mutation or recombination, then the more complex the organism becomes, the more variants have to be examined to find viable (let alone improved) alternatives. To use another analogy, this puts evolution in the position of a programmer having to improve larger and larger programs written in machine code. Such tasks rapidly overwhelm game-players, programmers, and reproductive organisms alike. The way out for the programmer is to write structured programs, including modular subroutines with simple interactions. Typical real programs—at least, the ones we can develop further—involve many layers of structure of this kind. It is very likely that the genetic code works this way too. Once an emergent property —say, a cilium or whisker—has proved to have survival value, the way to make further development possible is to represent the cilium as a module of the genetic program. That way, variants involving two cilia, or many, can be tried without expanding the search space beyond practical limits. There are many indications that this is how the genetic code works, from the observation that the sheer amount of DNA required to specify the most complex animals is not that much greater than that required for the simplest (or at least oldest), to the fact that *hox* genes appear to directly represent the linear order of body segments by position on the genome. When we look at language, we are looking at the end product of many evolutionary developments of this kind. To expect language itself to be a unitary emergent property of low-level general-purpose assemblages of neurons is to assume that it is quite unlike any other product of evolution—a standpoint that amounts to the most naive form of linguistic nativism.

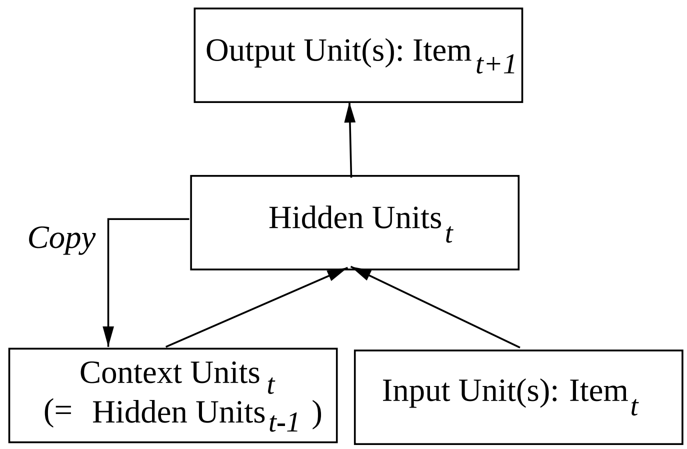

Figure 1
Simply Recurrent Network (SRN)

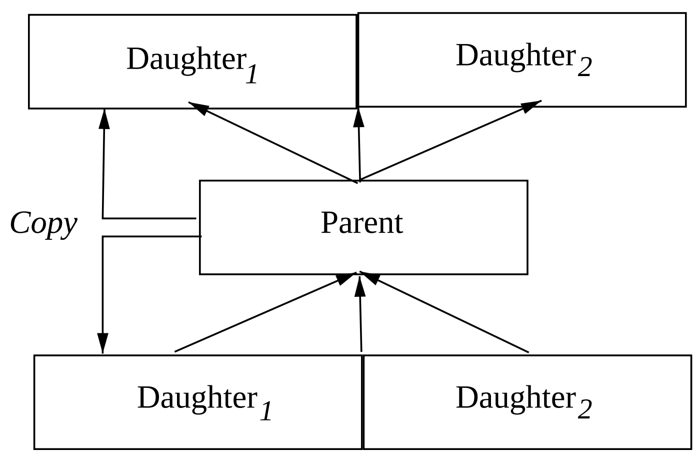

Figure 2
Recursive Auto-Associative Memory (RAAM)