Cognitive Science (in press).

Toward a connectionist model of recursion in human linguistic performance

Morten H. Christiansen
University of Southern California

Nick Chater
University of Warwick

Running head: A connectionist model of recursion

Address for correspondence:
Morten H. Christiansen
Program in Neural, Informational & Behavioral Sciences
University of Southern California
University Park MC-2520
Los Angeles, CA 90089-2520
Email: *morten@gizmo.usc.edu*
Phone: (213) 740-6299

After January 1, 1999:
Department of Psychology
Southern Illinois University
Carbondale, IL 62901-6502
Email: *morten@siu.edu*

Abstract

Naturally occurring speech contains only a limited amount of complex recursive structure, and this is reflected in the empirically documented difficulties that people experience when processing such structures. We present a connectionist model of human performance in processing recursive language structures. The model is trained on simple artificial languages. We find that the qualitative performance profile of the model matches human behavior, both on the relative difficulty of center-embedded and cross-dependency, and between the processing of these complex recursive structures and right-branching recursive constructions. We analyze how these differences in performance are reflected in the internal representations of the model by performing discriminant analyses on these representation both before and after training. Furthermore, we show how a network trained to process recursive structures can also generate such structures in a probabilistic fashion. This work suggests a novel explanation of people's limited recursive performance, without assuming the existence of a mentally represented competence grammar allowing unbounded recursion.

1 Introduction

Natural language is standardly viewed as involving a range of rarely occurring but important recursive constructions. But it is empirically well-documented that people are only able to deal easily with relatively simple recursive structures. Thus, for example, a doubly center-embedded sentence like (1) below is extremely difficult to understand.

(1) *The mouse that the cat that the dog chased bit ran away.*

In this paper, we present a connectionist network which models the limited human abilities to process and generate recursive constructions. The "quasi-recursive" nature of the performance of the connectionist network qualitatively models experimental evidence on human language processing.

The notion of recursion in natural language originates not from the project of trying to understand human linguistic *performance* which is the focus of this paper, but from the very different enterprise of specifying a "competence grammar"—a set of rules and/or principles which specify the legal strings of a language. It is standardly assumed that, if the competence grammar allows a recursive construction to apply at all, it can apply arbitrarily many times. Thus, if (2) is sanctioned by a recursive analysis with one level of recursion, then the grammar must thereby also sanction (1) with two levels of recursion and (3) with three levels of recursion.

(2) *The mouse that the cat bit ran away.*

(3) *The mouse that the cat that the dog that the man frightened chased bit ran away.*

Thus, the very idea that natural language is recursive requires a broadening of the notion of which sentences are in the language, to sentences like (3) which would presumably never be uttered or understood. In order to resolve the difference between language so construed and the language that humans are able to produce and comprehend, a distinction is typically made between linguistic *competence* and human *performance*. Competence in this context refers to a speaker/hearer's knowledge of the language, and is the subject of linguistic inquiry. In contrast, psycholinguists study performance—i.e., how linguistic knowledge is used in producing and understanding language, and also how extrinsic, non-linguistic factors may interfere with the use of that knowledge. It is here that "performance factors", such as memory limitations, can be invoked to show that some sentences, while consistent with linguistic competence, will never actually be said, or understood. The competence/performance distinction is also embodied in many symbolic models of language processing, such as CC-READER (Just & Carpenter, 1992). In this model, grammatical competence consists of a set of recursive production rules which are applied to produce state changes in a separate working memory. By imposing constraints on the capacity of the working memory system, performance limitations can be simulated without making changes to the competence part of the model.[1] The connectionist model we propose provides an alternative account of people's limited ability to

[1]See MacDonald & Christiansen (1998) for a critical discussion of CC-READER and similar language processing models based on production systems (Newell & Simon, 1976).

do recursion, without assuming an internally represented grammar which allows unbounded recursion—i.e., without invoking the competence/performance distinction.[2]

In the light of this discussion, it is clear that, from the point of view of modeling *psychological processes*, we need not take the purported unbounded recursive structure of natural language as axiomatic. Nor need we take for granted the suggestion that a speaker/hearer's knowledge of the language captures such infinite recursive structure. Rather, the view that "unspeakable" sentences which accord with recursive rules form a part of the knowledge of language is an *assumption* of the standard view of language developed by Chomsky and now dominant in linguistics and many areas of the psychology of language. The challenge for a computational model such as the connectionist model we propose is to account for those aspects of human comprehension/production performance which are suggestive of the standard recursive picture. If this can be done without making the assumption that the language processor really implements recursion, or that arbitrarily complex recursive structures are really sentences of the language, then it presents an alternative to adopting this assumption. Therefore, in assessing the connectionist simulations that we report below, the benchmark for performance of connectionist systems will be set by human abilities to handle recursive structures; we need not require that connectionist systems be able to handle recursion in full generality.

In this paper, we shall consider the phenomenon of natural language recursion in a 'pure' and highly simplified form. Specifically, we train connectionist networks on small artificial languages, which exhibit the different types of recursive structure found in natural language. We do this in order to address directly the classic arguments by Chomsky (1957) that recursion in natural language *in principle* rules out associative and finite state models of language processing. Indeed, the languages that we consider are based directly on the structures used in Chomsky's (1957) discussion. Considering recursion in a pure form permits us to address the in principle viability of connectionist networks in handling recursion, in much the same way as simple artificial languages have been used, for example, to assess the feasibility of symbolic parameter-setting approaches to the learning of linguistic structure (Gibson & Wexler, 1994; Niyogi & Berwick, 1996).

The structure of this paper is as follows. We begin by distinguishing varieties of recursion in natural language, considering the three kinds of recursion discussed in Chomsky (1957). We then summarize past connectionist research dealing with natural language recursion. Next, we introduce three artificial languages, based on the three kinds of recursion described by Chomsky, and present and analyze a range of simulations of connectionist networks trained on these languages. The results suggest that the networks are able to handle recursion to a degree comparable with humans. We close by drawing conclusions for the prospects of connectionist models of language processing.

2 Varieties of Recursion

Chomsky (1957) proposed that a recursive generative grammar consists of a set of phrase structure rules, complemented by a set of transformational rules (we shall not consider trans-

[2]The competence/performance distinction also leads to certain methodological problems—see Christiansen (1992, 1994) for further discussion.

formational rules further below). Phrase structure rules have the form $A \rightarrow BC$, with the interpretation that the symbol A can be replaced by the concatenation of B and C. A phrase structure rule is *recursive* if a symbol X is replaced by a string of symbols which includes X itself (e.g., $A \rightarrow BA$). The new symbol can then itself be replaced by a further application of the recursive rule, and so on. Recursion can also arise through the application of a recursive *set* of rules, none of which need individually be recursive. When such rules are used successively to expand a particular symbol, the original symbol may eventually be derived. A recursive *construction* in a natural or artificial language is one that is modeled using recursive rules; a *language* has recursive structure if it contains such constructions.

Modern generative grammar employs a wide range of formalisms, some quite distantly related to phrase structure rules. Nevertheless, corresponding notions of recursion within those formalisms can be defined. We shall not consider such complexities here, but use the apparatus of phrase structure grammar throughout.

There are a number of kinds of recursion relevant to natural language. First, there are kinds of recursion which produce languages which could equally well be generated without using recursion at all—specifically they could be generated by iteration, the application of a single procedure arbitrarily many times. For example, consider the case of *right-branching* recursion shown in Figure 1. These rules can be used to generate the right-branching sentences (4)–(6):

(4) *John loves Mary.*

(5) *John loves Mary who likes Jim.*

(6) *John loves Mary who likes Jim who dislikes Martha.*

But these structures can be produced or recognized by a simple iterative process, which can be carried out by a finite state machine. The recursive structures of interest to Chomsky, and of interest here, are those which cannot be replaced by iteration, and thus which appear to go beyond the capacities of finite state machines. Chomsky (1957) invented three simple artificial languages, generated by recursive rules, and which cannot be generated or parsed, at least in full generality, by a finite state machine using iteration.

————insert figure 1 about here————

The first artificial language can be defined by the following two phrase structure rules rules (where {} denotes the empty string; we shall not consider this "degenerate" case in the simulations below):

1.
   X → aXb
   X → {}

   which generate the strings:

   {}, *ab, aabb, aaabbb, aaaabbbb, . . .*

We call this *counting recursion*, because in order to parse such strings from left to right it is necessary to count the number of '*a*'s and note whether it equals the number of '*b*'s. This implies that full-scale counting recursion cannot be parsed by any finite device processing from left to right, since the number that must be stored can be unboundedly large (because there can be unboundedly large numbers of '*a*'s), and hence will exceed the memory capacity of any finite machine.

Chomsky's second artificial language can be characterized in terms of the phrase structure rules:

2.
$$\begin{aligned} \mathsf{X} &\rightarrow \mathsf{aXa} \\ \mathsf{X} &\rightarrow \mathsf{bXb} \\ \mathsf{X} &\rightarrow \{\} \end{aligned}$$
which generate the strings:

{}, *aa, bb, abba, baab, aaaa, bbbb, aabbaa, abbbba, ...*

We call this *mirror recursion*, because the strings exhibit mirror symmetry about their midpoint.

The final recursive language, which we call identity recursion, unlike counting and mirror recursion, cannot be captured by a context-free phrase structure grammar. Thus, in order to capture the final non-iterative recursive language we need to annotate our notion of rewrite rules. Here we adapt the meta-grammatical notation of Vogel, Hahn & Branigan (1996) to define the third artificial language in terms of the following rule set:

3.
$$\begin{aligned} \mathsf{S} &\rightarrow \mathsf{W}_i\mathsf{W}_i \\ \mathsf{W} &\rightarrow \mathsf{X} \\ \mathsf{X} &\rightarrow \mathsf{aX} \\ \mathsf{X} &\rightarrow \mathsf{bX} \\ \mathsf{X} &\rightarrow \{\} \end{aligned}$$
which generates the strings:

{}, *aa, bb, abab, aaaa, bbbb, aabaab, abbabb, ...*

We call this *identity recursion*, because strings consist of the concatenation of two identical copies of an arbitrary sequence of '*a*'s and '*b*'s. The index on W ensures that the two W's in the first rule are always the same.

Chomsky (1957) argued that each of these types of recursive language can be identified with phenomena in natural language. He suggested that *counting* recursion corresponds to sentence constructions such as *'if S_1, then S_2'* and *'either S_1, or S_2'*. These constructions can, Chomsky assumed, be nested arbitrarily deeply, as indicated by (7)–(9):

(7) *if S_1 then S_2.*

(8) *if if S_1 then S_2 then S_3.*

(9) *if if if S_1 then S_2 then S_3 then S_4.*

Mirror recursion is assumed to correspond to center-embedded constructions which occur in many natural languages (although typically with low frequency), as illustrated already in sentences (1)–(3). In these sentences, the dependencies between the subject nouns and their respective verbs are center-embedded, such that the first noun is matched with the last verb, the second noun with the second but last verb, and so on. Chomsky (1957) used the existence of center-embedded constructions to argue that natural language must be at least context-free, and hence beyond the scope of any finite state automaton.

In much the same way, *identity* recursion can be mapped on to a less common pattern in natural language, *cross-dependency*, which is found in Swiss-German and in Dutch,[3] as exemplified in (10)-(12) (from Bach, Brown & Marslen-Wilson, 1986):

(10) *De lerares heeft de knikkers opgeruimd.*
Literal: The teacher has the marbles collected up
Gloss: *The teacher collected up the marbles.*

(11) *Jantje heeft de lerares de knikkers helpen opruimen.*
Literal: Jantje has the teacher the marbles help collect up.
Gloss: *Jantje helped the teacher collect up the marbles.*

(12) *Aad heeft Jantje de lerares de knikkers laten helpen opruimen.*
Literal: Aad has Jantje the teacher the marbles let help collect up.
Gloss: *Aad let Jantje help the teacher collect up the marbles.*

In (10)–(12), the dependencies between the subject nouns and their respective verbs are crossed such that the first noun is matched with the first verb, the second noun with the second verb, and so on. The fact that cross-dependencies cannot be handled using a context-free phrase structure grammar has meant that this kind of construction, although rarely produced even in the small number of languages in which they occur, has assumed considerable importance in linguistics, because it appears to demonstrate that natural language is not context-free.[4]

Turning from linguistics to language processing, it is clear that, whatever the linguistic status of complex recursive constructions, they are very difficult to process, in contrast to right-branching structures. The processing of structures analogous to counting recursion has not been studied in psycholinguistics, but sentences such as (13) are plainly difficult to make sense of, though containing just one level of recursion (see also Reich, 1969).

(13) *If if the cat is in, then the dog cannot come in then the cat and dog dislike each other.*

The processing of center-embedded constructions has been studied extensively in psycholinguistics. These studies have shown, for example, that English sentences with more than one center-embedding (e.g., sentences (1) and (3) presented above) are read with the same intonation as a list of random words (Miller, 1962), cannot easily be memorized (Foss & Cairns,

[3]Cross-dependency has also been alleged, controversially, to be present in "respectively" constructions in English, such as *'Anita$_1$ and the girls$_2$ walks$_1$ and skip$_2$, respectively'*. Church (1982) questions the acceptability of these constructions with two cross-dependencies, and indeed, even one cross-dependency, as in this example, seems bizarre.

[4]Pullum & Gazdar (1982) have argued that natural language is, nonetheless, context-free, although their arguments are controversial (see Shieber, 1985, for a critique and Gazdar & Pullum, 1985, for a defense).

1970; Miller & Isard, 1964), and are judged to be ungrammatical (Marks, 1968). Bach et al. (1986) found the same behavioral pattern in German, reporting a marked deterioration of comprehension for sentences with more than one embedding. It has been shown that using sentences with a semantic bias or giving people training can improve performance on such structures, but only to a limited extent (Blaubergs & Braine, 1974; Stolz, 1967).

There has been much debate concerning how to account for the difficulty of center-embedded constructions in accounts of human natural language processing (e.g., Berwick & Weinberg, 1984; Church, 1982; Frazier & Fodor, 1978; Gibson, in press; Gibson & Thomas, 1996; Kimball, 1973; Pulman, 1986; Reich, 1969; Stabler, 1994; Wanner, 1980), typically involving postulating some kind of "performance" limitation on an underlying infinite competence. Cross-dependencies have received less empirical attention, but appear to present similar processing difficulties to center-embeddings (Bach et al., 1986; Dickey & Vonk, 1997), and we shall consider this data in a more detail when assessing our connectionist simulations against human performance, below.

3 Connectionism and Recursion

We aim to account for human performance on recursive structures as emerging from intrinsic constraints on the performance of a particular connectionist architecture, namely the Simple Recurrent Network (SRN) (Elman, 1990). But before presenting our simulation results, we first review previous connectionist approaches to natural language recursion.

One way of approaching the problem of dealing with recursion in connectionist models is to "hardwire" symbolic structures directly into the architecture of the network (e.g., Fanty, 1985; McClelland & Kawamoto, 1986; Miyata, Smolensky & Legendre, 1993; Small, Cottrell & Shastri, 1982). The network can therefore be viewed as a non-standard implementation of a symbolic system, and can solve the problem of dealing with recursive natural language structures by virtue of its symbol processing abilities, just as do standard symbolic systems in computational linguistics. Connectionist re-implementations of symbolic systems may potentially have novel computational properties and even be illuminating regarding the appropriateness of a particular style of symbolic model for distributed computation (Chater & Oaksford, 1990). Such models do not figure here, because we are interested in exploring the viability of connectionist models as *alternatives* to symbolic approaches to recursion.[5]

There are two classes of models which may potentially provide such alternatives—both of which *learn* to process language from experience, rather than implementing a prespecified set of symbolic rules. The first, less ambitious, class (e.g., Chalmers, 1990; Hanson & Kegl, 1987; Niklasson & van Gelder, 1994; Pollack, 1988, 1990; Stolcke, 1991) attempts to learn grammar from *"tagged"* sentences. Thus, the network is trained on sentences which are associated with some kind of grammatical structure and the task is to learn to assign the appropriate grammatical structure to novel sentences. This means that much of the structure of the language is not learned by observation, but is built into the training items. These models are

[5]The possibility remains, of course, that connectionist models might, on analysis, be found to achieve what success they do in virtue of having learned to approximate, to some degree, symbolic systems. Smolensky (in press) has argued that connectionist networks can only capture the generalizations in natural language structure in this way.

related to statistical approaches to language learning such as stochastic context-free grammars (Brill, Magerman, Marcus & Santorini, 1990; Jelinek, Lafferty, & Mercer, 1990) in which learning sets the probabilities of each grammar rule in a prespecified context-free grammar, from a corpus of parsed sentences.

The second class of models, which includes the model presented in this paper, attempts the much harder task of learning syntactic structure from strings of words. The most influential approach, which we shall follow in the simulations reported below, has been based on SRNs (Elman, 1990). An SRN involves a crucial modification to a feedforward network (see Figure 2)—the current set of hidden unit values is "copied back" to a set of additional input units, and paired with the *next* input to the network. This means that the current hidden unit values can directly affect the next state of the hidden units; more generally, this means that there is a loop around which activation can flow for many time-steps. This gives the network a memory for past inputs, and therefore the ability to deal with integrated sequences of inputs presented successively. This contrasts with standard feedforward networks, the behavior of which is determined solely by the current input. SRNs are thus able to tackle tasks such as sentence processing in which the input is revealed gradually over time, rather than being presented at once.

————insert figure 2 about here————

Recurrent neural networks provide a powerful tool with which to model the learning of many aspects of linguistic structure, particularly below the level of syntax (e.g., Allen & Christiansen, 1996; Christiansen, Allen & Seidenberg, in press; Cottrell & Plunkett, 1991; Elman, 1990, 1991; Norris, 1990; Shillcock, Levy & Chater, 1991). Moreover, SRNs seem well-suited to learning finite state grammars (e.g., Cleeremans, Servan-Schreiber & McClelland, 1989; Giles, Miller, Chen, Chen, Sun & Lee, 1992; Giles & Omlin, 1993; Servan-Schreiber, Cleeremans & McClelland, 1991). But relatively little headway has been made towards grammars involving complex recursion that are beyond simple finite-state devices. Previous efforts in modeling complex recursion have fallen within two general categories: simulations using language-like grammar fragments and simulations relating to formal language theory.

In the first category, networks are trained on relatively simple artificial languages, patterned on English. For example, Elman (1991, 1993) trained SRNs on sentences generated by a small context-free grammar incorporating center-embedding and a single kind of right-branching recursive structures. The behavior of the trained networks are reported to be qualitatively comparable with human performance in that a) the SRN predictions for right-branching structures are more accurate than on sentences of the same length involving center-embedding, and b) performance degrades appropriately when the depth of center-embedding increases. Weckerly & Elman (1992) corroborate these results and suggest that semantic bias (incorporated via co-occurrence restriction on the verbs) can facilitate network performance as they have been found to be in human processing (Blaubergs & Braine, 1974; Stolz, 1967). These results are encouraging, but preliminary. They show that SRNs can deal with specific examples of recursion, but provide no systematic analysis of their capabilities. Within the same framework, Christiansen (1994, 1998) trained SRNs on a recursive artificial language incorporating four kinds of right-branching structures, a left branching structure, and center-embedding. Again, the desired degradation of performance on center-embedded

constructions as a function of embedding depth was found, as were appropriate differences between center-embedding and right-branching structures.[6] However, a closer study of the recursive capabilities of the SRNs showed that the prediction accuracy for the right-branching structures also degraded with depth of recursion—albeit not as dramatically as in the center-embedding case. Additional simulations involving a variant of this language, in which cross-dependency constructions substituted for the center-embedded sentences (rendering a mock "Dutch" grammar) provided similar results. Together these simulation results indicate that SRNs can embody constraints which limit their abilities to process center-embeddings and cross-dependencies to levels similar to human abilities. This suggests that SRNs can capture the quasi-recursive structure of actual spoken language. One of the contributions of the present paper is to show that the SRN's general pattern of performance is relatively invariant over variations in network parameters and training corpus—thus, we claim, the human-like pattern of performance arises from *intrinsic* constraints of the SRN architecture.

While work pertaining to recursion within the first category has been suggestive but in many cases relatively unsystematic, the second category of simulations related to formal language theory has seen more detailed investigations of a small number of artificial tasks, typically using very small networks. For example, Wiles & Elman (1995) made a detailed study of what we have called counting recursion using the simplest possible language $a^n b^n$. They studied recurrent networks with 2 hidden units,[7] and found a network that was able to generalize successfully to inputs far longer than those on which they had been trained. They also presented a detailed analysis of the nature of the solution found by one of the networks. Batali (1994) used the same language, but employed SRNs with 10 hidden units and showed that networks could reach good levels of performance, when selected by a process of "simulated evolution" and then trained using conventional methods. Based on a mathematical analysis, Steijvers & Grünwald (1996) "hardwired" a second order recurrent network (Giles et al., 1992) with 2 hidden units such that it could process the context-sensitive counting language $b(a)^k b(a)^k \ldots$ for values of k between 1 and 120. An interesting outstanding question, which we address in the simulations below, is whether these levels of performance can be obtained if there are more than two vocabulary items—e.g., if the network must learn to assign items into different lexical categories ("noun" and "verb") as well as paying attention to dependencies between these categories. This question is important with respect to the potential relevance of these results for natural language processing.

No comparable detailed study has been conducted with either center-embedding or crossed-dependency type (mirror and identity recursion) constructions.[8] In the studies below, we therefore aimed to comprehensively study and compare all three types of recursion discussed in Chomsky (1957)—that is, counting, mirror, and identity recursion—with the less complex right-branching recursion as a baseline. We also used syntactic categories which contained

[6] The networks also demonstrated sophisticated generalization abilities, ignoring local word co-occurrence constraints while appearing to comply with structural information at the constituent level. Some of these results were reported in a reply by Christiansen & Chater (1994) to Hadley's (1994) criticism of connectionist language learning models such as that of Elman (1990, 1991).

[7] The nets were trained using back-propagation through time (Rumelhart, Hinton & Williams, 1986) rather than the standard method for training SRNs—for a discussion of differences and similarities between the two types of networks, see Chater & Conkey (1992) and Christiansen (1994).

[8] The only exception we know of is our own preliminary work regarding Chomsky's (1957) three artificial languages reported in Christiansen (1994).

a number of different vocabulary items, rather than defining the grammar over single lexical items, as in the detailed studies of counting recursion and the context-sensitive counting language described above. Using these simple abstract languages allows recursion to be studied in a "pure" form, without interference from other factors. Despite the idealized nature of these languages, the SRN's performance qualitatively conforms to human performance on similar natural language structures.

Another novel aspect of the present studies is that we provide a statistical benchmark against which the performance of the networks can be compared. This is a simple prediction method borrowed from statistical linguistics based on n-grams, i.e., strings of n consecutive words. The benchmark program is "trained" on the same stimuli used by the networks, and simply records the frequency of each n-gram in a look-up table. It makes predictions for new material by considering the relative frequencies of the n-grams which are consistent with the previous $n-1$ words. The prediction is a vector of relative frequencies for each possible successor item, scaled to sum to 1, so that they can be interpreted as probabilities, and are therefore directly comparable with the output vectors produced by the networks. Below, we report the predictions of bigram and trigram models and compare them with network performance.[9] Although not typically used for comparison in connectionist research, these simple models might provide insight into the sequential information to which the networks may be responding, as well as a link to non-connectionist corpus-based approaches to language learning in computational linguistics (e.g., Charniak, 1993).

4 Three Benchmark Tests Concerning Recursion

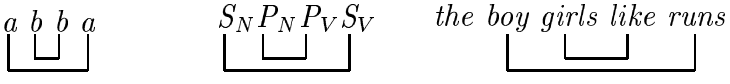
We constructed three benchmark test languages for connectionist learning of recursion, based on Chomsky's three artificial languages. Each language involved two kinds of recursive structure: one of the three complex recursive constructions and the right-branching construction as a baseline. Vocabulary items were divided into "nouns" and "verbs", incorporating both singular or plural forms. An end of sentence marker completed each sentence.

1. *Counting recursion*

   aabb *NNVV*

   For counting recursion, we treat Chomsky's symbols *'a'* and *'b'* as corresponding to the word categories of noun and verb, respectively, while ignoring singular/plural agreement.

2. *Center-embedding recursion*

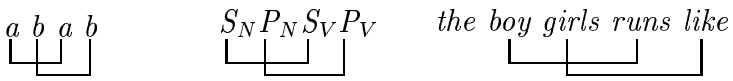
   

   In center-embedding recursion, we map Chomsky's symbols *'a'* and *'b'* in mirror recursive constructions onto the categories of singular and plural words (whether nouns

[9] Intuition would suggest that higher order n-grams models should fare better than simple bigram and trigram models. However, computational results using large text corpora have shown that higher order n-grams provide for poor predictions because of the frequent occurrence of "singletons"; i.e., n-grams with only a single or very few instances (Gale & Church, 1990; Redington, Chater & Finch, in press).

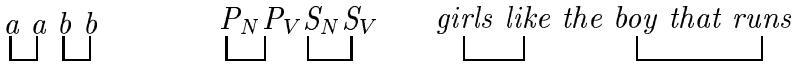
or verbs). Nouns and verbs agree for number as in center-embedded constructions in natural language.

3. *Cross-dependency recursion*



In cross-dependency recursion we map Chomsky's symbols *'a'* and *'b'* in identity recursive constructions onto the categories of singular and plural words. Nouns and verbs agree for number as in cross-dependency constructions in natural language.

4. *Right-branching recursion*



For right-branching recursion, we map the symbols *'a'* and *'b'* onto the categories of singular and plural words. Nouns and verbs agree for number as in right-branching constructions in natural language.

Thus, the counting recursive language consisted of both counting recursive constructions (1) interleaved with right-branching recursive constructions (4), the center-embedding recursive language of center-embedded recursive constructions (2) interleaved with right-branching recursive constructions (4), and the cross-dependency recursive language of cross-dependency recursive constructions (3) interleaved with right-branching recursive constructions (4).

How can we assess to what degree a connectionist net has mastered these simple languages? By analogy with standard linguistic methodology, we could train the net to make explicit "grammaticality judgments", i.e., to distinguish legal and non-legal sentences of the languages. But the concern of this paper is people's *performance* on recursive structures, rather than judgments concerning grammaticality (which is often assumed to relate to linguistic competence).[10] Therefore, we chose to use a task which directly addressed the way in which the system processes sentences of the languages, rather than requiring it to make meta-linguistic judgments about sentence legality.

Elman (1990) suggested such an approach, which has become standard in many of the SRN studies of natural language described above. The approach is to train the network to predict the *next* item in a sequence given previous context. That is, the SRN gets a word as input at time t and then has to predict the next word at time $t+1$. Although the prediction task is not deterministic, the structure of the languages imposes a number of regularities which make prediction a meaningful task. At the beginning of a sentence it is impossible to know whether the sentence will involve a complex recursive construction (1-3 depending on the language) or a right-branching recursive construction (4). However, once the second word is encountered it becomes clear whether the sentence involves a complex or right-branching construction: A verb indicates a right-branching construction whereas another noun indicates a complex recursive construction. A right-branching construction may end after the first

[10]The relation between grammaticality judgments and processing mechanisms both within linguistics and psycholinguistics is a matter of much controversy (for further discussion, see Christiansen, 1994; Schütze, 1996).

noun/verb pair, or continue with one or more embeddings. For none of the three complex recursive constructions is it possible to determine precisely how many more nouns will be encountered in a sentence or whether they will be singular or plural. Once the nouns have been encountered, however, it is possible to determine exactly how many verbs a sentence will have and which (singular/plural) form each of them will have (except, of course, for counting recursion). Assuming that the system can learn to distinguish nouns from verbs, it should be able to make correct predictions about subsequent verbs as well as the end of sentence marker once it receives the first verb as input. Specifically, the number of verbs will correspond to the number of nouns, and the form of the verbs will agree with the form of the nouns as specified by each particular language. The end of sentence marker should be predicted after the last verb.

Bearing this in mind, we briefly consider the demands of learning each of the four recursive constructions. Construction 1, counting recursion, should be the easiest of the complex recursive structures to process. Because this construction does not have any agreement constraints imposed on it, correct performance can be achieved simply by counting the number of occurring nouns, and then predicting the same number of verbs. The simplest way to process construction 2, center-embedding recursion, is to develop a last-in-first-out memory or *"stack"* to store agreement information. Thus, the agreement information for each noun is retrieved in the opposite order in which it was stored—leading to the mirror agreement pattern. The most obvious way to process construction 3, cross-dependency recursion, is to develop a first-in-first-out memory or *"queue"*. Thus, the agreement information of each noun is retrieved in the same order in which it was stored—leading to the identity agreement pattern. Finally, in construction 4, right-branching recursion, in contrast to the processing of the three complex kinds of recursion mentioned above, processing unbounded right-branching structures does not involve unbounded memory load because each noun is immediately followed by a verb which agrees with it.

In practice, a connectionist network is unlikely to learn to implement any standard symbolic parsing method, which has unbounded competence, limited by memory restrictions (e.g., as in CC-READER, Just & Carpenter, 1992). Nonetheless, an appreciation of how these methods work is useful in assessing the nature and difficulty of the task that the network faces. Parsing each of these languages using symbolic means requires quite elaborate computational machinery. However, as we show below, networks are able to obtain good levels of performance on each of the three languages, each containing complex recursive constructions of a given kind (1-3) interleaved with the right-branching base-line constructions (4).

5 Simulation Results

We trained SRNs on each of the three languages, using a sixteen word vocabulary with four singular nouns, four singular verbs, four plural nouns, and four plural verbs.[11] All SRNs had 17 input and output units, where each unit corresponded to a single word, or to the end of

[11] Most of the simulations presented here were carried out using the *Tlearn* neural network simulator available from the Center for Research on Language at the University of California, San Diego. For the sentence generation simulations reported in Section 5.6 we used the *Mlearn* simulator developed from the *Tlearn* simulator by Morten Christiansen.

sentence marker. We used SRNs in which the hidden layer contained between 2 and 100 units. Except when explicitly noted otherwise, the training corpora consisted of 5000 sentences of variable length, and the test corpora of 500 novel sentences, generated in the same way as the training sentences and excluded from the original training corpora. Each corpus of sentences was concatenated into a single, long string which was presented to the SRN one word at a time. Both training and test corpora were comprised of 50% complex recursive constructions of the appropriate kind for a given language interleaved with 50% right-branching constructions. The distribution of depth of embedding is shown in Table 1. The mean sentence length in both training and test corpora was approximately 4.7 words (SD: 1.3).

————insert table 1 about here————

Since the input consists of a single concatenated string of words (including end of sentence markers), the network has to discover that the input consists of sentences; that is, nouns followed by verbs (ordered according to the constraints of the language being learned) and delineated by end of sentence markers. Consider, as an example, an SRN being trained on the center-embedding language and presented with the two sentences: '$n_1 v_5 \# N_3 n_8 v_2 V_4 \#$'.[12] First, the network gets 'n_1' as input and is expected to produce 'v_5' as output. The weights are then adjusted depending on the discrepancy between the actual output that the net produced and the desired output using the back-propagation learning algorithm (Rumelhart, Hinton & Williams, 1986). Next, the SRN receives 'v_5' as input and is required to output the end-of-sentence marker ('#'). At the next time-step, '#' is provided as input and 'N_3' is the target output, followed by the input/output pairs: 'N_3/n_8', 'n_8'/'v_2', 'v_2'/'V_4', and 'V_4'/'#'. Training continues in the same manner for the remainder of the training corpus.

During the test phase, test corpora were presented to the SRNs and the output recorded for each input word while the weights remain frozen. In any interesting language-like task, the next item is not deterministically specified by the previous items. In the above example at the start of the second sentence, the grammar for the center-embedding language permits both noun categories, 'n' and 'N', to occur at the beginning of a sentence. If the SRN has acquired the relevant aspects of the grammar which generated the training sentences, then it should activate all word tokens in both 'n' and 'N' following an end of sentence marker. Hence, it is appropriate for the prediction to take the form of a probability distribution of possible next items. We can assess the degree to which the network has succeeded in learning this probability distribution by comparing the output of the network against some estimation of the conditional probabilities given the previous input (note that this gives a less noisy indication of network performance than comparing the predictions against the actual next items in the training corpus). Next, we use empirically derived conditional probabilities for global performance comparisons, whereas in Section 5.2 we introduce a measure of Grammatical Prediction Error to evaluate the SRN's performance in more detail.

[12]Here and in the following, we adopt the convention that 'n' and 'N' corresponds to categories of nouns, 'v' and 'V' to categories of verbs with capitalization indicating plural agreement where required by the language in question. The end of sentence marker is denoted by '#'. Individual word tokens are denoted by adding a subscript to a word category, e.g., 'N_3'.

5.1 Overall performance

In order to asses the overall performance of the SRNs, we made comparisons between network output probability distributions and the full conditional probabilities given prior context. To see how this comparison is done, consider the prediction of the next word at a particular point in the test corpus. For example, the full conditional probabilities given the context, '$N_6 n_1 v_3$', can be represented as a vector containing the probabilities of being the next item for each of the tokens in the categories 'n', 'N', 'v', 'V', and '#'. Because there are no statistical dependencies between sentences, the conditional probability distribution need only take account of the previous words in the sentence. More formally, the probability of the pth item, $\mathbf{w_p}$, in a sentence is conditional on the previous $p-1$ items:[13]

$$P(\mathbf{w_p}|\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_{p-1}}) \tag{1}$$

Following Elman (1991), these values can be estimated empirically from the training corpus, according to the relation:

$$P(\mathbf{w_p}|\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_{p-1}}) \simeq \frac{Freq(\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_{p-1}}, \mathbf{w_p})}{Freq(\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_{p-1}})} \tag{2}$$

However, empirical word-based conditional probabilities as determined by Equation 2 are not useful to assess performance on corpora consisting entirely of novel sentences. Consider a novel string, '$N_2 V_5 n_1 v_3 n_2 v_6 N_4 V_4 \#$', in which only the substring, '$N_2 V_5 n_1 v_3$' has occurred in the training corpus. Empirical word-based probability estimation can be carried out for the first four words, but not for the second half of the string because there are no prior context which can be relied upon. One solution to this problem is to estimate the conditional probabilities based on the prior occurrence of lexical categories—i.e., 'NVnvnvNV#' in the above example—rather than individual words. Thus, with $\mathbf{c_i}$ denoting the category of the ith word in the sentence we have the following relation:

$$P(\mathbf{c_p}|\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_{p-1}}) \simeq \frac{Freq(\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_{p-1}}, \mathbf{c_p})}{Freq(\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_{p-1}})} \tag{3}$$

where the probability of getting some member of a given lexical category as the pth item, $\mathbf{c_p}$, in a sentence is conditional on the previous $p-1$ lexical categories. Note that for the purpose of performance assessment singular and plural nouns are assigned to separate lexical categories throughout this paper as are singular and plural verbs.

Given that the choices of lexical item for each category are independent, and that each word in the category is equally frequent,[14] the probability of encountering a particular word $\mathbf{w_n}$, which is a member of a category $\mathbf{c_p}$, is simply inversely proportional to the number of items, $\mathbf{C_p}$, in that category. So, overall,

$$P(\mathbf{w_n}|\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_{p-1}}) \simeq \frac{Freq(\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_{p-1}}, \mathbf{c_p})}{Freq(\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_{p-1}})\ \mathbf{C_p}} \tag{4}$$

[13] We use bold for random variables.

[14] These assumptions are true in these artificial languages, although they will not be in natural language, of course.

If the network is performing optimally, then the vector of output unit activations should exactly match these probabilities. We evaluate the degree to which the network performs successfully by measuring the summed squared difference between the network outputs and the conditional probabilities. More formally, we define Squared Error as follows:

$$\text{Squared Error} = \sum_{j \in W} (out_j - P(\mathbf{w_n} = j))^2 \tag{5}$$

where W is the set of words in the language (including the end of sentence marker), and there is an output unit of the network corresponding to each word. The index j runs through each possible next word, and compares the network output to the conditional probability of that word. Finally, we obtain an overall measure of the network's performance by calculating the Mean Squared Error (MSE) across all the items in the 500 test sentences. MSE calculated in this way will be used to give a global measure of the performance of both networks and n-gram prediction models in the simulations below. Thus, if the network or n-gram model has learned the conditional probability distribution perfectly, the resulting MSE would be 0.

5.1.1 Intrinsic constraints on SRN performance

Earlier simulations concerning the three languages (Christiansen, 1994) have shown performance to degrade as the depth of embedding increases. As mentioned earlier, SRN simulations in which center-embedded structures were included in small grammar fragments have resulted in the same outcome (Christiansen, 1994, 1998; Elman, 1991, 1993; Weckerly & Elman, 1992)—as did the inclusion of cross-dependency structures (Christiansen, 1994, 1998). This performance degradation on complex recursive structures qualitatively fits human performance on similar constructions (as described in section 2). Symbolic models, which embody a grammatical competence which allows unbounded recursion, are also able to mimic the human constraints on complex recursive structures. However, these models can only fit the human data by adding in otherwise arbitrary performance limitations specifically aimed at capturing human limitations on doubly center-embedded construction. Examples include limits on stack depth (Church, 1982; Marcus, 1980), limits on the number of allowed sentence nodes (Kimball, 1973) or partially complete sentence nodes (Stabler, 1994) in a given sentence, the "self-embedding interference constraint" (Gibson & Thomas, 1996), and an upper limit on sentential memory cost (Gibson, in press). On the other hand, what constrains the performance of the SRN appears to be architectural limitations interacting with the statistics of the recursive structures. In this connection, it is interesting to note that the difficulty of processing center-embedded structures is not confined to a linguistic context. Larkins & Burns (1977) demonstrated that when subjects were asked, for example, to name pairs of letters and digits given their center-embedded relations (e.g., the pairs *L-2*, *G-8* and *W-5*, from the string *'WGL285'*), they experienced the same difficulty on this task as when processing center-embedded sentences. This suggests that constraints on complex recursive structures, such as center-embedding, may derive from non-linguistic processing constraints. The well-documented human limitations on center-embedded sentences may therefore be more appropriately modeled by non-linguistic architectural constraints—such as those exhibited by the SRN—than the linguistically motivated extrinsic constraints of the symbolic approaches.

A possible objection this suggestion is that perhaps the human-like performance limitations of the SRN follow from using a hidden unit layer of a particular size, rather than from

intrinsic architectural properties. This seems reasonable by analogy with standard feedforward networks where the size of the hidden unit layer is typically assumed to correlate with the processing ability of a network. If this objection were correct, then the SRN architecture could be compatible with almost any level of performance on recursive constructions. This would mean that the size of an SRN's hidden unit layer would provide an arbitrary limitation on recursion in similar ways to the extrinsic constraints within the symbolic approaches.

To address this objection, we carried out a series of simulations in which SRNs with 2, 5, 10, 15, 25, 50, and 100 hidden units were trained on the three artificial languages. Across all simulations, the learning rate was 0.1, no momentum was used, and the initial weights were randomized to values in the interval [-0.25,0.25]. Although the simulation results presented in this paper were replicated across different initial weight randomizations, we focus on a typical set of simulations for the ease of exposition. Networks of the same size were given the same initial random weights to facilitate comparisons across the three languages.

Figure 3 shows the performance of the different sized nets on test corpora consisting entirely of novel complex recursive structures of varying length as a function of the number of epochs trained on the three languages. The SRNs performed well on the task, as reflected in the low error scores. In the case of the counting recursion language (top panel), after some initial differences—mainly within the first 40 epochs of training—all nets apart from those with 2 hidden units settled on a similar level of performance. A similar pattern of uniformity of learning was also found for the center-embedding recursion language (middle panel), with the exception that the SRN with 5 hidden units also showed a higher MSE. The SRNs trained on the cross-dependency recursion language (bottom panel) followed the same trend as the counting recursion networks. The results of this series of simulations suggest that the above objection does not apply to the SRN. Increasing the hidden unit layer size did not result in an increase in performance (i.e., lower MSE)—once the SRN had a necessary minimum of units (the number of which for the present tasks appears to lie around 5-10 hidden units).

————insert figure 3 about here————

SRN performance as a function of hidden unit layer size on test corpora consisting entirely of novel right-branching constructions of varying length can be seen in Figure 4. These results showed a strong uniformity across both language type and size of hidden unit layer—with the one minor exception that the SRN with 2 hidden units trained on the center-embedding language (middle panel) performed slightly worse than the larger SRNs trained on this language as well as worse than the 2 hidden unit SRNs trained on the counting recursion (top panel) and cross-dependency (bottom panel) languages, respectively.

————insert figure 4 about here————

That SRN performance is independent of the number of hidden units is further illustrated in Figure 5 which shows MSE averaged across epochs for both complex recursion (left panels) and right-branching recursion (right panels) for each size of net (grey bars). These values were calculated as the average of the MSEs sampled at every second epoch (from epoch 0 to epoch 100), and plotted for each construction in Figures 3 and 4. The MSE for bigram and trigram models are included (black bars) for comparison. For the nets trained on the

counting recursion language, once a network had 15 or more hidden units they obtained a low level of MSE on complex recursive structures (top left panel). Performance on right-branching structures (top right panel) was very similar across all hidden unit layers sizes. For both types of recursion, the counting recursion nets outperformed the bigram and trigram models. In the case of the nets trained on the center-embedding recursion language, all nets with 10 or more hidden units achieved essentially the same level of performance on complex recursive structures (middle left panel), whereas the nets with 5 or more hidden units performed quite similar on the right-branching structures (middle right panel). Again, the SRNs were doing better on the both recursion types than the bigram and trigram models—at least for hidden unit layer sizes larger than 5. Nets with 15 or more hidden units trained on the cross-dependency language all reached the same level of performance on complex recursive structures (bottom left panel). As with counting recursion, performance was quite uniform on right-branching recursive constructions (bottom right panel) across all hidden unit layers sizes. Again, the SRNs outperformed the bigram and trigram models.

————insert figure 5 about here————

Comparing across the three languages we see that the SRN found the counting recursion language slightly easier to learn (as predicted in Section 4) than the two other languages. Surprisingly, the nets appeared to find the cross-dependency language easier to learn than the center-embedding language (at least in terms of their ability to reduce MSE). This is an important result because people appear to be better at dealing with cross-dependency constructions than equivalent center-embedding constructions. This is surprising from the perspective of linguistic theory because, as we noted above, cross-dependency constructions are typically viewed as more complex than center-embedding constructions because they cannot be captured by phrase-structure rules.

Another interesting result is that in contrast to the SRNs the bigram and trigram models showed the opposite effect, achieving a better level of performance on the center-embedding language than on the cross-dependency language. Finally, the SRNs with 10 or more hidden units had a lower MSE on complex recursive structures than on right-branching structures. This could be due to the fact that the complex recursive constructions essentially become deterministic (with respect to length) once the first verb has been encountered, but this is not true for the right-branching constructions at any point (except at depth 3).

The above results show that the size of the hidden unit layer, when sufficiently large, does not influence the processing capability on test corpora with constructions of varying size. Yet it is conceivable that hidden unit layer size may be important when processing the crucial doubly embedded complex recursive structures which are beyond the limits of human performance. To investigate this possibility we therefore retested the SRNs (trained on complex and right-branching constructions of varying length) on corpora which consisted exclusively of novel doubly embedded structures. The results from these tests are presented in Figure 6, showing approximately the same performance uniformity as found in Figure 5. Thus, once an SRN has a sufficient number of hidden units, the size of the hidden layer does not seem to matter for the performance on novel doubly embedded complex constructions drawn from the three languages. Figure 6 also shows that once an SRN has a sufficient size it performs considerably better on doubly embedded constructions than both n-gram models.

————insert figure 6 about here————

Given the lack of effect of hidden unit layer size on performance, we concentrate on SRNs with 15 hidden units in the remaining simulations. Inspection of Figure 3 reveals that the performance on complex recursive constructions for these networks reached an asymptotic level after 35–40 epochs of training (with subsequent training resulting only in minor differences in performance). From the sets of MSEs recorded for epochs 2 through 100, we chose the number of epochs at which the 15 hidden unit SRNs had the lowest MSE. The best level of performance was found after 54 epochs training on the counting recursive language, 66 epochs of training on the center-embedding language, and 92 epochs of training on the cross-dependency language. All results reported below are from SRNs trained for these number of epochs (except when explicitly noted otherwise).

5.2 Performance at different depths of embedding

We have seen that the overall performance of the SRNs averaged across different depths of recursion appears comparable with the quasi-recursive structure of actual human utterances. We now consider performance at different levels of embedding, measuring the differential effects of depth of recursion on the various types of complex recursion and right-branching recursion. Human data would suggest that performance should rapidly degrade as embedding depth increases for complex recursive structures, where performance should degrade only slightly for right-branching recursive constructions.

In the previous section we used empirical conditional probabilities based on lexical categories to assess SRN performance (Equations 4 and 5). However, this measure is not useful for assessing performance on novel constructions which either go beyond the depth of embedding found in the training corpus, or deviate, as ungrammatical forms do, from the grammatical structures encountered during training. For comparisons with human performance we therefore use the measure of Grammatical Prediction Error (GPE). This measure of SRN performance has been shown elsewhere to provide for good approximations of two behavioral measures: reading times (MacDonald & Christiansen, 1998) and human grammaticality ratings (Christiansen, 1998; Christiansen & MacDonald, 1998).

When evaluating how well the SRN has learned the grammar, which it was exposed to via the sentences in the training corpus, it is important from a linguistic perspective not only to determine whether the words that were activated given prior context are grammatical, but also which items were *not* activated despite being sanctioned by the grammar. The GPE provides an indication of how well a network is obeying the training grammar in making its predictions, taking hits, false alarms, correct rejections and misses into account.

Hits and false alarms are calculated as the accumulated activations of the set of units, G, that are grammatical and the set of ungrammatical activated units, U, respectively:

$$\text{hits} = \sum_{i \in G} u_i \tag{6}$$

$$\text{false alarms} = \sum_{i \in U} u_i \tag{7}$$

Traditional sensitivity measures, such as *d'* (Signal Detection Theory, Green & Swets, 1966)

or α (Choice Theory, Luce, 1959), are based on the assumption that misses can be calculated as the difference between total number of relevant observations and hits. In terms of network activation it is not clear what would correspond to "total number of relevant observations".[15] Consequently, we need an alternative means of quantifying misses; that is, a way to determine an activation-based penalty for not activating all grammatical units and/or not allocating sufficient activation to these units. With respect to GPE, the calculation of misses involves the notion of a target activation, t_i, computed as a proportion of the total activation (hits and false alarms) determined by the lexical frequency, f_i, of the word that unit i designates and weighted by the sum of the lexical frequencies, f_j, of all the grammatical units:

$$t_i = \frac{(\text{hits} + \text{misses}) f_i}{\sum_{j \in G} f_j} \tag{8}$$

The potential missing activation for each unit can then be determined as the positive discrepancy, m_i, between the target activation for a grammatical unit, t_i, and the actual activation of that unit, u_i:

$$m_i = \begin{cases} 0 & \text{if } t_i - u_i \leq 0 \\ t_i - u_i & \text{otherwise} \end{cases} \tag{9}$$

Finally, the total activation for misses is calculated as the sum over all single unit missing activation values:

$$\text{misses} = \sum_{i \in G} m_i \tag{10}$$

The GPE for predicting a particular word given previous sentential context can then be determined by:

$$\text{GPE} = 1 - \frac{\text{hits}}{\text{hits} + \text{false alarms} + \text{misses}} \tag{11}$$

Thus construed, the GPE provides a measure of how much of the activation for a given item has been placed correctly according to the grammar (hits) in proportion to the total amount of activation (hits and false alarms) and the penalty for not activating grammatical items sufficiently (misses). Although not an explicit part of the above equation, correct rejections are also taken into account under the assumption that they correspond to zero activation for units that are ungrammatical given previous context.

The GPE score ranges between 0 and 1, and provides a very stringent measure of performance. To obtain a perfect GPE score of 0 the SRN will not only have to predict all and only the next items prescribed by the grammar, but also be able to scale those predictions according to the lexical frequencies of the legal items. Notice that to obtain a low GPE score the network must make the correct subject noun/verb agreement predictions. Consider first a hypothetical situation in which an SRN trained on the center-embedding language is violating subject noun/verb agreement. Given the previous context '$N_3 n_8 v_2$' the network is incorrectly activating the group of singular verbs by 0.80 and the end of sentence marker by 0.10, while only activating the grammatically appropriate group of plural verbs by 0.10. This yields a hit activation of 0.10, a false alarm activation of 0.90, and a miss activation of 0.90, resulting in

[15] Note that total activation cannot be construed as corresponding to "total number of relevant observations" in these measures of sensitivity. This is because the difference between the total activation and hit activation (as specified by Equation 6) corresponds to the false alarm activation (as specified by Equation 7).

a very high GPE score of ($1 - \frac{0.10}{0.10+0.90+0.90} =$) 0.95. Consider then the opposite hypothetical situation in which the SRN given the same previous context instead highly activates the group of grammatically correct plural verbs by 0.8, and the (incorrect) group of singular verbs by 0.1 and the (incorrect) end of sentence marker by 0.1. This yields a hit activation of 0.80, a false alarm activation of 0.20, and a miss activation of 0.20, resulting in a relatively low GPE score of ($1 - \frac{0.80}{0.80+0.20+0.20} =$) 0.33. Thus, in order to obtain low GPE scores the SRN predictions must comply with subject noun/verb agreement along with other grammatical constraints.

The GPE value for an individual word reflects the difficulty that the SRN experienced for that word given the previous sentential context. Previous studies using the GPE measure of SRN performance (Christiansen, 1998; MacDonald & Christiansen, 1998) have found that individual word GPE can be mapped qualitatively onto word reading times, with low GPE values reflecting a prediction for short reading times and high values indicating long predicted reading times. The average GPE across a whole sentence expresses the difficulty that the SRN experienced across the sentence as a whole, and this measure has been found to map onto sentence grammaticality ratings, with low average GPE scores indicating a high "goodness" rating and high scores reflecting low ratings (Christiansen & MacDonald, 1998) .

5.2.1 Embedding depth performance

With the GPE measure in hand, we now turn to SRN performance on different depths of embedding. Figure 7 shows the average GPE on complex and right-branching recursive structures as a function of embedding depth for 15HU SRNs, bigram models, and trigram models (trained on complex and right-branching constructions of varying length). Each data point represents the mean GPE measured across 10 novel sentences. For the SRN trained on the counting recursion language there was little difference between the performance on complex and right-branching recursive constructions, and performance only deteriorated slightly across embedding depth. In contrast, the n-gram models (and especially the trigram model) exhibited better performance (i.e., lower GPE scores) on right-branching structures than on complex recursive structures. Both n-gram models showed a sharper decrease in performance across depth of recursion than the SRN. The SRN trained on the center-embedding language also performed better than the n-gram models, although it, too, had greater difficulty with complex recursive structures than with right-branching structures. Interestingly, SRN performance on right-branching recursive structures decreased slightly with depth of recursion. This contrasts with what would be expected from symbolic models in which infinite depth of right-branching recursion pose no processing problems (e.g., Church, 1982; Gibson, in press; Marcus, 1980; Stabler, 1994). However, the pattern of performance deterioration of the SRN appears to be in line with human data (see Section 5.4.3). A comparison between the n-gram models' performance on the center-embedding recursion language shows that whereas both exhibited a similar pattern of performance decrease on the complex recursive constructions across embedding depth, the trigram models performed considerably better on the right-branching constructions than the bigram model. As with the MSE results presented above in Section 5.1.1, SRN performance on the cross-dependency language was better than on the center-embedding language. Although the SRN, as in the previous case, obtained lower GPE scores on right-branching constructions compared with complex recursive structures, the increase in GPE across embedding depth on the latter was considerably less for the cross-dependency net

than for its center-embedding counterpart. The bigram model performed rather poorly on the cross-dependency language both on right-branching and complex recursive structures. The trigram model performed substantially better, although it was no match for the SRN on complex recursive constructions—even though the SRN performed slightly worse on right-branching structures compared with the trigram models.

————insert figure 7 about here————

The differential SRN performance on the complex recursive and right-branching constructions from both the center-embedding and the cross-dependency languages provides a good fit with human data.[16] We will discuss the lack of such difference in the SRN performance on the counting recursion language in Section 5.4.4. Finally, it should be noted that constructions of recursive depth 4 did not exist in any of the training corpora. Yet, there was no abrupt breakdown in performance for any of the three languages at this point, and this was true of both SRNs and n-gram models. This suggests that these models are able to generalize to at least one extra level of recursion beyond what they have been exposed to during training (and this despite only 1% of the training items being of depth 3).

5.3 Training exclusively on doubly embedded complex constructions

An alternative objection to the idea of intrinsic constraints being the source of SRN limitations on multiple embeddings of complex recursive structures is that perhaps the constraints stem from the specific statistics of the training corpora. Thus, one can concede that the size of the hidden unit layer is not the source of the constraint, but contend that the fact that only 7% of the sentences involved doubly embedded complex recursive structures is what explains the observed limitations of the nets with these structures. Thus, if the percentage of doubly embedded constructions was to be increased considerably the networks would perhaps be able to process these constructions without noticeable difficulty.

To investigate this possibility, we trained SRNs with 15 hidden units on versions of the three languages consisting exclusively of doubly embedded complex recursive constructions without interleaving right-branching constructions. Using the same number of words in each epoch as in the previous simulations, best performance was found for the counting recursion depth 2 trained SRN (D2-SRN) after 48 epochs, after 60 epochs for the center-embedding D2-SRN, and after 98 epochs for the cross-dependency D2-SRN. When tested on the test corpora containing only novel doubly embedded sentences (also used to produce the results in Figure 6), the

[16]It could be objected that the GPE measure may hide a failure to make correct agreement predictions for singly center-embedded sentences, such as *'The man$_1$ the boys$_2$ chase$_2$ likes$_1$ cheese'*. If correct, one would expect a high degree of agreement error for the two verb predictions in the singly center-embedded (complex depth 1) constructions in Figure 7. Agreement error can be calculated as the percentage of verb activation allocated to verbs which do *not* agree in number with their respective nouns. The agreement error for the first and second verbs were 1.00% and 16.85%, respectively. This result also follows from the earlier discussion of the GPE measure, establishing that a high degree of agreement error will result in high GPE scores. Moreover, note that the level of SRN agreement error is comparable with human performance: For example, Larkin & Burns (1977) found that when subjects were asked to paraphrase singly center-embedded constructions presented auditorily they made errors nearly 15% of the time.

average MSE found for the counting recursion network was 0.045 (vs. 0.080 for the previous 15HU SRN), 0.066 for the center-embedding net (vs. 0.092 for the previous 15HU SRN), and 0.073 for the cross-dependency net (vs. 0.079 for the previous 15HU SRN). Interestingly, although there were significant differences between the MSE scores for the SRNs and D2-SRNs trained on the counting recursion ($t(98) = 3.13, p < 0.003$) and center-embedding recursion ($t(98) = 3.04, p < 0.004$) languages, the difference between the two nets was not significant for the cross-dependency language ($t(98) = .97, p > 0.3$). The performance of the D2-SRNs thus appear to be somewhat better than the performance of the SRNs trained on the corpora of varying length—at least for the counting and center-embedding recursion languages. However, a closer look at the predictions that the nets made show that D2-SRNs are only slightly better than their counterparts trained on sentences of varying length.

Figure 8 shows GPE scores as a function of word position across doubly embedded complex recursive constructions from the three languages, averaged over 10 novel sentences. On the doubly embedded counting recursive sentences (top panel), both SRN and D2-SRN performed well, with a slight advantage for the D2-SRN on the last verb. Both networks obtained lower levels of GPE than the n-gram models which was relatively inaccurate, especially at the last two verbs. On doubly center-embedded sentences (middle panel), the two SRNs showed a gradual pattern of performance degradation across the sentence, but with the D2-SRN achieving somewhat better performance, especially on the last verb. The bigram and trigram models performed quite similarly, and again had great difficulty with the two final verbs. When processing doubly embedded cross-dependency sentences (bottom panel) the SRNs exhibited a pattern of performance resembling that found for counting recursion. The GPE scores for both SRNs increased gradually, and close to each other, until the first verb was encountered. At this point, the SRN GPE for the second verb dropped whereas the D2-SRN GPE continued to grow. At the third verb, the GPE for the D2-SRN dropped whereas the SRN GPE increased. Although the pattern of SRN GPE scores may seem puzzling at first, it appears to fit recent results concerning the processing of similar cross-dependency constructions in Dutch. Using a phrase-by-phrase self-paced reading task with stimuli adapted from Bach et al. (1986), Dickey & Vonk (1997) found a significant jump in reading times between the second and third verb, preceded by a (non-significant) decrease in reading times between the first and second verb. When the GPE scores for individual words are mapped onto reading times, the GPE pattern of the SRN, but not the D2-SRN, provides a reasonable approximation of the pattern of reading times found by Dickey & Vonk. Returning to Figure 8, the trigram model—although not performing as well as the SRN—displayed a somewhat similar pattern to the SRN, whereas the bigram model performed very poorly. Together, the results presented in Figure 8 reveal that despite being trained exclusively on doubly embedded complex recursive constructions and despite not having to acquire the regularities underlying the right-branching structures, the D2-SRN only performed slightly better on doubly embedded complex recursive constructions than the SRN trained on both complex and right-branching recursive constructions of varying length. This suggests that the performance of the SRN does not merely reflect the statistics of the training corpus, but that intrinsic architectural constraints also play a crucial role in determining prediction behavior.

————insert figure 8 about here————

An encouraging aspect of the simulations is that the SRNs performed a good deal better

than the n-gram based models. This is particularly important because the material that we have used in these studies is the most favorable possible for n-gram models, since there is no intervening material at a given level of recursion. In natural language, of course, there is generally a considerable amount of material between changes of depth of recursion, which causes problems for n-gram models because they concentrate on short-range dependencies. While n-gram models do not generalize well to more linguistically natural examples of recursion, SRN models, by contrast, do show a good level of performance on such material. We have found in other work (Christiansen, 1994, 1998; Christiansen & Chater, 1994) that the addition of intervening non-recursive linguistic structure does not appear to significantly alter the pattern of results found with the artificial languages reported here. Thus, we may conclude that SRNs are not merely learning bigram and trigram information, but appear to acquire grammatical regularities that, at least on a qualitative level, allow them to exhibit behaviors similar to humans. We now consider the match with human data in more detail.

5.4 Fitting Human Data

We have seen that the performance of the SRN in processing deeply embedded complex recursive structures appears to be limited by intrinsic constraints, which are independent of the number of hidden units. Moreover, we have shown that these limitations are not overcome even when the network is trained only on deeply embedded sentences.[17] As we have mentioned already, the level of embedding which the SRNs can process is roughly in line with human processing limitations. In this section, we consider in more detail the relation between the SRN's performance and the psychological data on processing recursive structures, using the same 15 hidden unit SRNs as in the previous sections.

5.4.1 Center-embedding vs. cross-dependency

In a classic study, Bach et al. (1986) found that cross-dependencies in Dutch were comparatively easier to process than center-embeddings in German. We have noted that this result is linguistically interesting and surprising, because cross-dependencies cannot be captured by phrase structure rules and are therefore typically viewed as more complex. Moreover, it is interesting from the point of view of language processing, because it suggests that the language processor cannot be primarily based on a stack-like memory store. This is because cross-dependencies, which require a queue, are easier to process than center embeddings, which require a stack.

Bach et al. had native Dutch speakers listen to sentences in Dutch involving varying depths of recursion in the form of cross-dependency constructions and corresponding right-branching paraphrases with the same meaning. Native German speakers were tested using similar materials in German, but with the cross-dependency constructions replaced by center-embedded constructions. Because of disagreement among German informants concerning whether the final verb should be in an infinitive form or in a past participle form, two versions of the

[17] Earlier work by Christiansen (1994) has additionally shown that these results are not significantly altered by training the SRNs exclusively on complex recursive structures of varying length (without interleaving right-branching constructions) or by using the back-propagation through time learning algorithm (Rumelhart, Hinton & Williams, 1986).

German materials were used. After the presentation of each sentence, subjects were asked to rate the comprehensibility of the sentence on a 9-point scale (1 = easy, 9 = difficult). Subjects were also asked comprehension questions after two-thirds of the sentences. In order to remove effects of processing difficulty due to length, Bach et al. subtracted the ratings for the right-branching paraphrase sentences from the matched complex recursive test sentences. The same procedure was applied to the error scores from the comprehension questions. The resulting difference should thus reflect the difficulty caused by the complex recursive constructions.

Figure 9 (left panel) shows the difference in mean test/paraphrase ratings for singly and doubly embedded cross-dependency sentences in Dutch and singly and doubly center-embedded sentences in German (with the final verb in a past participle form). We focus on the past participle German results because these were consistent across both the rating and the comprehension tasks, providing the same pattern of results in comparison with the Dutch data. As the mean GPE across a sentence reflects the processing difficulty that the SRN experiences across the sentence as a whole, we can map these scores onto the human rating data because the latter are thought to reflect the processing difficulty that people experience when processing a given sentence. We used the mean GPE scores from Figure 7 for the SRNs trained on the center-embedding and cross-dependency languages to model the Bach et al. results. For recursive depth 1 and 2, the mean GPE scores for the right-branching constructions were subtracted from the average GPE scores for the complex recursive constructions, and the differences plotted in Figure 9 (right panel).[18] The net trained on the cross-dependency language maps onto the Dutch data and the net trained on the center-embedding language maps onto the German (past participle) data. At a single level of embedding, Bach et al. found no difference between Dutch and German. This is also true of the SRN data ($t(18) = 0.36, p > 0.7$). However, at two levels of embedding Bach et al. found a significant difference between Dutch and German with the Dutch cross-dependency stimuli being rated better than their German counterparts. The SRN data also shows a significant difference between center-embedded constructions at depth 2 and their cross-dependency counterparts ($t(18) = 4.08, p < 0.01$). Thus, SRN performance on center-embedding and cross-dependency of depth 1 and 2 fits human performance on similar constructions quite closely. Next, the performance on doubly center-embedded sentences is studied in more detail.

————insert figure 9 about here————

5.4.2 Grammatical vs. ungrammatical double center-embeddings

The study of complex English sentences with multiple center-embeddings has long been an important source of information about the limits of human sentence processing (e.g., Blaubergs & Braine, 1974; Foss & Cairns, 1970; Marks, 1968; Miller, 1962; Miller & Isard, 1964; Stolz, 1967). A particularly interesting recent finding is due to Gibson & Thomas (1997). Their results from an off-line rating task suggest that some ungrammatical sentences involving doubly center-embedded object relative clauses may be perceived as grammatical.

[18] The human data presented here and in the next two sections involve three different scales of measurement (i.e., differences in mean test/paraphrase comprehensibility ratings, mean grammaticality ratings on a scale 1-7, and mean comprehensibility ratings on a scale 1-9). It was therefore necessary to adjust the scales for the comparisons with the mean GPE scores accordingly.

(14) *The apartment that the maid who the service had sent over was cleaning every week was well decorated.*

(15) * *The apartment that the maid who the service had sent over was well decorated.*

In particular, they found that when the middle VP was removed (as in 15), the resulting construction was rated no worse than the grammatical version (in 14). Gibson & Thomas interpreted this as an indication that people find doubly center-embedded relative clause structures just as acceptable when only two verb phrases are included instead of the grammatically-required three.

It is possible to investigate this result using the SRN trained on the center-embedding language. Within this abstract language, (14) corresponds to a grammatical 3VP ('NNNVVV') construction (with singular subject noun/verb agreements), whereas the ungrammatical (15) corresponds to a 2VP ('NNNVV') construction (again with singular subject noun/verb agreements). By looking at the output activation following 'NNNVV' we can determine whether the SRN can fit the Gibson & Thomas data. Figure 10 shows this activation averaged over 10 novel sentences and grouped into the four lexical categories and the end of sentence marker (EOS). It is clear that, in contrast to the results of Gibson & Thomas, the network demonstrated a significant preference for the ungrammatical 2VP construction over the grammatical 3VP construction, predicting that (14) actually should be rated worse than (15). This explains, in part, the high GPE score for the third verb in Figure 8 (middle panel). The erroneous activation of the nouns and the plural verbs also contribute to the high GPE.

————insert figure 10 about here————

The Gibson & Thomas study employed an off-line task which may explain why (14) was rated no worse than (15). Christiansen & MacDonald (1998) pursued this observation by conducting an on-line self-paced word-by-word (center presentation) grammaticality judgment task using the stimuli from Gibson & Thomas (1997). At each point in a sentence subjects were asked to use their intuition to judge whether what they had read so far was a grammatical sentence or not. Following the presentation of each sentence (whether accepted or rejected), subjects rated the sentences on a 7-point scale (1 = good, 7 = bad). Christiansen & MacDonald found that the grammatical 3VP construction was rated significantly worse than the ungrammatical 2VP construction.

One potential problem with this experiment is that the 2VP and 3VP stimuli were not of the same length, and hence that the result could be an artifact of mere length differences. In addition, the Gibson & Thomas stimuli also incorporated noun/verb semantic biases (e.g., *apartment/decorated, maid/cleaning, service/sent over* in (14)) which may make the 2VP stimuli more plausible than they would have been otherwise. Christiansen & MacDonald therefore replicated their first experiment using stimuli controlled for length and without noun/verb biases, such as (16) and (17):

(16) *The chef who the waiter who the busboy offended appreciated admired the musicians.*

(17) * *The chef who the waiter who the busboy offended frequently admired the musicians.*

Figure 11 shows the rating results from the second experiment in comparison with SRN predictions in terms of mean GPE. The GPE scores for the 2VP and 3VP constructions were recorded for 10 novel sentences with singular subject noun/verb agreements similarly to the agreement patterns in the human stimuli (and the lack of semantic noun/verb bias). To control for length the GPE scores for the 3VP constructions were only averaged over the first 6 words (i.e., for 'NNNVVV' vs. 'NNNVV#' for the 2VP constructions). As in the first study, Christiansen & MacDonald found in their second study that the grammatical 3VP constructions were rated significantly worse than the ungrammatical 2VP constructions. The SRN data fitted this pattern of human grammaticality ratings, with significantly higher GPE scores elicited by 3VP constructions compared with 2VP constructions ($t(18) = 2.34, p < 0.04$). In the next section, we investigate a possible match between human and SRN performance on right-branching structures.

————insert figure 11 about here————

5.4.3 Right-branching subject relative constructions

Traditional symbolic models suggest that right-branching (and left-branching) recursion should not cause any processing problems. In contrast, as we mentioned in connection with SRN performance across depth of recursion on the center-embedding language shown in Figure 7, the SRN model suggests that some decrement in performance may occur. Unfortunately, this issue has not received much attention in the experimental literature—even though right-branching constructions are often used as control items in studies of center-embedded sentences. However, it is possible to glean some relevant information from some of these studies. Thus, Bach et al. (1986) report comprehensibility ratings for their right-branching paraphrase items. Figure 12 shows as a function of recursion depth the comprehensibility ratings for the German past participle paraphrase sentences and the mean GPE scores produced for right-branching constructions (from Figure 7) by the SRN trained on the center-embedding language. Both the human and the SRN data show the same pattern of increasing processing difficulty with increase in the depth of recursion.

————insert figure 12 about here————

A similar fit with human data can be found by comparing the human comprehension errors as a function of recursion depth reported in Blaubergs & Braine (1974) with mean GPE for the same depths of recursion (again for the SRN trained on the center-embedding language). Christiansen & MacDonald (1998) present on-line rating data concerning right-branching PP modifications of nouns in which the depth of recursion varied from 0 to 2 by modifying a noun by either one PP (18), two PPs (19), or three PPs (20):

(18) *The nurse with the vase says that the [flowers by the window] resemble roses.*

(19) *The nurse says that the [flowers in the vase by the window] resemble roses.*

(20) *The blooming [flowers in the vase on the table by the window] resemble roses.*

The stimuli were controlled for length and generally constructed to be of similar propositional and syntactic complexity. The results showed that subjects rated sentences with recursion of depth 2 (20) worse than sentences with recursion depth 1 (19), which, in turn, were rated worse than sentences with no recursion (18). Although these results do not concern subject relative constructions, they suggest together with data from the Bach et al. and the Blaubergs & Braine studies that the processing of right-branching recursive constructions is affected by recursion depth—albeit to a much lesser degree than for complex recursive constructions. Importantly, this dovetails with the SRN model of language processing that we have presented here and elsewhere (Christiansen, 1994, 1998; Christiansen & MacDonald, 1998). In contrast, traditional symbolic models of language (e.g., Church, 1982; Gibson, in press; Marcus, 1980; Stabler, 1994) do not predict an increase in processing difficulty for right-branching constructions as a function of depth of recursion, except perhaps for a mere length effect.

5.4.4 Counting recursion

In the final part of this section, we briefly discuss the relationship between counting recursion and natural language. We could find no experimental data which relate to natural language constructions corresponding to counting recursion. The good performance of the SRNs trained on counting recursion might suggest the prediction that people should be able to handle relatively deep embeddings of corresponding natural language constructions (e.g., the SRN handles doubly embedded structures successfully). However, we contend that, despite Chomsky (1957), such structures may not exist in natural language. Indeed, the kind of structures that Chomsky had in mind (e.g., nested *'if–then'* structures) may actually be closer to center-embedded constructions than to counting recursive structures. Consider the earlier mentioned depth 1 example (13), repeated here as (21):

(21) *If*$_1$ *if*$_2$ *the cat is in, then*$_2$ *the dog cannot come in then*$_1$ *the cat and dog dislike each other.*

As the subscripts indicate, the *'if–then'* pairs are nested in a center-embedding order. This structural ordering becomes even more evident when we mix *'if–then'* pairs with *'either–or'* pairs (as suggested by Chomsky, 1957: p. 22):

(22) *If*$_1$ *either*$_2$ *the cat dislikes the dog, or*$_2$ *the dog dislikes the cat then*$_1$ *the dog cannot come in.*

(23) *If*$_1$ *either*$_2$ *the cat dislikes the dog, then*$_1$ *the dog dislikes the cat or*$_2$ *the dog cannot come in.*

The center-embedding ordering seems necessary in (22) because if we reverse the order of *'or'* and *'then'* then we get the obscure sentence in (23). Given these observations, we can make the empirical prediction that human behavior on nested *'if–then'* structures are likely to follow the same breakdown pattern as observed in relation to the nested center-embedded constructions (perhaps with a slightly better overall performance).

5.5 Probing the Internal Representations

The intrinsic constraints of the SRN appear to provide a good qualitative match with the limitations on human language processing. We now consider how these constraints arise by conducting an analysis of the hidden unit representations with which the SRNs store information about previous linguistic material. We focus on the case of doubly embedded constructions, which represent the limits of performance for both people and the SRN. Moreover, we focus on what information the hidden units of the SRN maintain about the number agreement of the three nouns encountered in doubly embedded constructions (recording the hidden units' activations immediately after the three nouns have been presented).

Before giving our formal measure, we provide an intuitive motivation for our approach. Suppose that we aim to assess how much information the hidden units maintain about the number agreement of the last noun in a sentence; that is, the noun that the net has just seen. If the information is maintained very well, then the hidden unit representations of input sequences that end with a singular noun (and thus belong to the lexical category combinations: nn-**n**, nN-**n**, Nn-**n** and NN-**n**) will be well-separated in hidden unit space from the representations of the input sequences that end with a plural noun (i.e., NN-**N**, Nn-**N**, nN-**N** and nn-**N**). This means that we should be able to split the hidden unit representations *along* the plural/singular noun category boundary such that input sequences ending in plural nouns are separated from input sequences ending in singular nouns. It is important to contrast this with a situation in which the hidden unit representations instead retain information about the agreement number of individual nouns. In this case, we should be able to split the hidden unit representations *across* the plural/singular noun category boundary such that input sequences ending with particular nouns, say, N_1, n_1, N_2 or n_2 (i.e., nn-$\{N_1, n_1, N_2, n_2\}$,[19] nN-$\{N_1, n_1, N_2, n_2\}$, Nn-$\{N_1, n_1, N_2, n_2\}$ and NN-$\{N_1, n_1, N_2, n_2\}$) are separated from input sequences ending with remaining nouns N_3, n_3, N_4 or n_4 (i.e., nn-$\{N_3, n_3, N_4, n_4\}$, nN-$\{N_3, n_3, N_4, n_4\}$, Nn-$\{N_3, n_3, N_4, n_4\}$ and NN-$\{N_3, n_3, N_4, n_4\}$). Note that the above separation along lexical categories is actually a special case of across category separation in which input sequences ending with the particular (singular) nouns n_1, n_2, n_3 or n_4 are separated from input sequences ending with the remaining (plural) nouns N_1, N_2, N_3 or N_4. Only by comparing the separation along and across the lexical categories of singular/plural nouns can we assess whether the hidden unit representations merely maintain agreement information about individual nouns, or whether more abstract knowledge has been encoded pertaining to the categories of singular and plural nouns. In both cases, information is maintained relevant to the prediction of correctly agreeing verbs, but only in the latter case are such predictions based on a generalization from the occurrences of individual nouns to their respective categories of singular and plural nouns.

We can measure the degree of separation by attempting to split the hidden unit representations generated from the $(8 \times 8 \times 8 =)$ 512 possible sequences of three nouns into two equal groups. We attempt to make this split using a plane in hidden unit space; the degree to which two groups can be separated either along or across lexical categories therefore provides a measure of what information the network maintains about the number agreement of the last seen noun. A standard statistical test for the separability of two groups of items is

[19] We use curly brackets to indicate that any of the four nouns may occur in this position, thus creating the following four combinations: nn-N_1, nn-n_1, nn-N_2 and nn-n_2.

discriminant analysis (Cliff, 1987; see Bullinaria, 1994; Wiles & Bloesch, 1992; Wiles & Ollila, 1993 for earlier applications to the analysis of neural networks).

Figure 13(a) gives a schematic illustration of a separation along lexical categories with a perfect differentiation of the two groups, corresponding to a 100% correct classification of the hidden unit vectors. The same procedure can be used to assess the amount of information that the hidden units maintain concerning the number agreement of the nouns in second and first positions. We split the same hidden unit activations generated from the 512 possible input sequences into groups both along and across lexical categories. The separation of the hidden unit vectors along the lexical categories according to the number of the second noun shown in Figure 13(b) is also perfect. However, as illustrated by Figure 13(c), the separation of the hidden unit activations along the lexical categories according to the first encountered noun is less good, with 75% of the vectors correctly classified, because **N**-N**n** is incorrectly classified with the singulars and **n**-n**N** with the plurals.

————insert figure 13 about here————

We recorded hidden unit activations for the 512 possible noun combinations for both complex and right-branching recursive constructions of depth 2 (ignoring the interleaving verbs in the right-branching structures). Table 2 lists the percentage of correctly classified hidden unit activations for the 512 possible combinations of nouns. Classification scores were found for these noun combinations both before and after training, and both for separation along and across singular/plural noun categories. Scores were averaged over different initial weight configurations and collapsed across the SRNs trained on the three languages (there was no significant differences between individual scores). The results from the separations across singular/plural noun categories show that prior to any training the SRN was able to retain a considerable amount of information about the agreement number of individual nouns in the last and middle positions. Only for the first encountered noun was performance essentially at chance (that is, close to the level of performance achieved through a random assignment of the vectors into two groups). The SRN had, not surprisingly, no knowledge of lexical categories of singular and plural nouns before training, as indicated by the lack of difference between the classification scores along and across noun categories. The good classification performance of the untrained nets on the middle noun in the right-branching constructions is, however, somewhat surprising because this noun position is two words (a verb and a noun) away from the last noun. In terms of absolute position from the point where the hidden unit activations were recorded, the middle noun in right-branching constructions (e.g., '$N_1V_3-\mathbf{N_3}-V_2n_4$') corresponds to the first noun in complex recursive constructions (e.g., '$\mathbf{N_1}-N_3n_4$'). Whereas untrained classification performance for this position was near chance on complex recursion, it was near perfect on right-branching recursion. This suggests that in the latter case information about the verb, which occurs between the last and the middle nouns, does not interfere much with the retention of agreement information about the middle noun. Thus, prior to learning the SRN appears to have an architectural bias which facilitates the processing of right-branching structures over complex recursive structures (at least for the present implementation of the two kinds of recursion).

————insert table 2 about here————

After training, the SRNs retained less information in its hidden unit representations about individual nouns. Instead, lexical category information was maintained as evidenced by the big differences in classification scores between groups separated along and across singular/plural noun categories. Whereas classification scores along the two noun categories had increased considerably as a result of training, the scores for classifications made according to groups separated across the categories of singular and plural nouns had actually decreased—especially for the middle noun position. The SRN appears to have acquired knowledge about the importance of the lexical categories of singular and plural nouns for the purpose of successful performance on the prediction task, but at the cost of retaining information about individual nouns in the middle position.

We have suggested that SRNs embody intrinsic architectural constraints which make them suitable for the modeling of recursive structure—in particular the human limitations on complex recursion documented in many empirical studies. The results of the discriminant analyses suggest that the SRN is well-suited for learning sequential dependencies. Importantly, the feedback loop between the context layer and the hidden layer allows the net to retain information relevant to making appropriate distinctions between previously encountered plural and singular items even prior to learning. Of course, a net has to learn to take advantage of this initial separation of the hidden unit activations to produce the correct output, and this is a nontrivial task. Prior to learning, the output of an SRN consist of random activation patterns. Thus, it has to discover the lexical categories and learn to apply agreement information in the right order to make correct predictions for center-embedded and cross-dependency complex recursive structures. As a consequence of training, the SRN is able to retain a significant amount of information about even the first noun in complex recursive constructions, as well as exhibiting an output behavior very much in line with human data.

On a methodological level, the results from the discriminant analyses of the untrained networks suggests that when conducting analyses of hidden unit representations in recurrent networks after training it is advisable to make comparisons with the representations as they were prior to training. This may provide insight into which aspects of network performance are due to architectural biases and which arise due to learning. A network always has some bias with respect to a particular task, and this bias is dependent on a number of factors, such as, overall network configuration, the nature of the activation function(s), the properties of the input/output representations, the initial weight setting, etc. As evidenced by our discriminant analyses, even prior to learning hidden unit representations may display some structural differentiation, emerging as the combined product of this bias (also cf. Kolen, 1994) and the statistics of the input/output relations in the test material (also cf. Chater & Conkey, 1992). However, all too often hidden unit analyses—such as cluster analyses, multi-dimensional scaling analyses, principal component analyses—are conducted with no attention paid to the potential amount of structure that can be found in the hidden unit representations before any learning takes place. But by making comparisons with analyses of hidden unit patterns elicited prior to training, not only may over-interpretation of training results be avoided, but it is also possible to gain more insight into the kind of architectural constraints that a given network brings to a particular task.

5.6 Sentence Generation

We have so far considered how recursive structures are processed, and studied the hidden unit representations that the SRN has before and after training. We now briefly show how SRNs can also be used to model the *generation* of recursive structures. This provides additional insight into what the networks have learned, and also provides a possible starting point for modeling how people produce recursive constructions.

The basic idea is to interpret the output of the SRNs not as a set of predictions, but as a set of possible sentence continuations. One of these possible *continuations* can then be chosen stochastically, and fed back as the next input to the SRN. This is illustrated in Figure 14. The process starts from a randomly chosen noun given as input to the network. The network then produces a distribution of possible successors. The stochastic selection process (SSP) first normalizes the outputs so that they sum to 1 (and hence can be interpreted as probabilities), and then chooses one of the outputs randomly, according to these probabilities. This item is given as the next input to the network, and the process is repeated. Eventually, the end of sentence marker will be selected, and a sentence will be completed. However, the generation process need not be halted at this point, as the end of sentence marker can serve as an input from which the first word of the next sentence can be produced. In this way, the generation process can be continued indefinitely to produce an arbitrarily large corpus of sentences from the SRN. A similar approach was used by Mozer & Soukup (1991) to generate musical sequences.[20]

————insert figure 14 about here————

Table 3 presents the distribution of the grammatical sentences obtained from a sample of 100 sentences generated for each language by the SRNs with 15 hidden units. The counting recursion net generated 67% grammatical sentences, the center-embedding net 69% grammatical sentences, and the cross-dependency net 73% grammatical sentences. Thus, once again the cross-dependency net performed better than the center-embedding net (and the counting recursion net). The table is further divided into three subgroups, depending on whether the constructions are of depth 0, complex recursive or right-branching. Across the three languages there was a larger proportion of grammatical sentences of depth 0 (35–42%) than found in the training corpora (30%), suggesting a weak tendency to generate shorter strings. There was also a some tendency towards producing more grammatical right-branching sentences (50–67%) than complex recursive sentences (especially for the counting recursion net) despite the fact both kinds of recursion occurred equally often in the training corpora. For complex recursion, both the counting recursive net and the cross-dependency net generated several structures of depth 2, whereas the center-embedding net generated none. Thus, the center-embedding net appeared to have acquired a slightly stronger bias toward shorter strings than the two other nets. In the case of right-branching recursion, all nets were able to generate at least two sentences of depth 2, again indicating that the SRN found these structures easier to deal with than the complex recursive structures.

————insert table 3 about here————

[20] We thank Paul Smolensky for bringing this work to our attention.

The ungrammatical strings from the 100 sentence samples are listed in Table 4. Agreement errors accounted for less than a quarter of the ungrammatical strings: 24% for counting recursion, 13% for center-embedding recursion, and 19% for cross-dependency recursion. The ungrammatical strings were divided into four subgroups on the assumption that the combination of a single noun and a single verb counted as depth 0, the initial occurrence of two or more nouns as complex recursion, and the initial occurrence of a noun and a verb followed by other material as right-branching recursion. The fourth subgroup, "Other", consisted of strings which either started with a verb or were null strings (i.e., just an end of sentence marker). Few errors were made on depth 0 constructions. The counting recursion net made more errors on right-branching structures than on complex recursive structures, whereas the opposite is true of the center-embedding net. The cross-dependency net made about same number of errors on both kinds of constructions. Whereas many of the non-agreement errors are hard to interpret, the nets did make some interesting errors involving a combination of both a complex recursive construction and a right-branching construction, whose individual parts were otherwise grammatical (counting recursion: 'NnvVnv'; center-embedding recursion: 'NVNNVV' and 'NVNNNVVV'; cross-dependency recursion: 'nvnNvV' and 'NVnnvv').

————insert table 4 about here————

On the whole, the networks performed reasonably well on the stochastic sentence generation task; that is, their acquired knowledge of the structural regularities provided a good basis for the probabilistic generation of sentences—though performance does not reach human levels of production. Nonetheless, given these encouraging initial results, we can speculate that the representations that the SRNs acquire through training may form a good common representational substrate for both sentence recognition and production. That is, knowledge acquired in the service of comprehension may form the basis for production (see Dell, Chang & Griffin, in press, for a similar perspective on SRN sentence production). It is worth noting that viewed in this way, the SRN embodies the asymmetry typically found between human language comprehension and production. The nets predominantly generated sentences of depth 0 and 1, but are able to process sentences of depth 2 (albeit to a very limited degree). Thus, the nets have a comprehension basis which is wider than their productive capabilities. Of course, sentence generation in these nets is not driven by semantics contrary to what one would assume to be the case for people. Adding semantics to guide the generation process may help eliminate many of the existing ungrammatical sentences because the selection of words would then be constrained not only by probabilistic grammatical constraints but also by semantic/contextual constraints.

6 General Discussion

We have shown that an SRN can be trained to process recursive structures with similar performance limitations regarding depth of recursion as found in human language processing. The limitations of the network do not appear sensitive to the size of the network, nor to the frequency of deeply recursive structures in the training input. The qualitative pattern of results from the SRN for center-embedding, cross-dependency and right-branching recursion match human performance on natural language constructions with these structures. The

SRNs trained on center-embedded and cross-dependency constructions performed well on singly embedded sentences—although, as for people, performance was by no means perfect (Bach et al., 1986; Blaubergs & Braine, 1974; King & Just, 1991). Of particular interest is the pattern of performance degradation on sentences involving center-embeddings and cross-dependencies of depth 2, and its close match with the pattern of human performance on similar constructions.

Overall, the qualitative match between the SRN performance and human data is encouraging. These results suggest a reevaluation of Chomsky's (1957, 1959) arguments that the existence of recursive structures in language rules out finite state and associative models of language processing. These arguments have been taken to indicate that connectionist networks, which learn according to associative principles, cannot in principle account for human language processing. But we have shown that this in principle argument is not correct: Connectionist networks can learn to handle recursion with a comparable level of performance to the human language processor. The simulations that we have provided are, of course, small scale, and we have not demonstrated that this approach could be generalized to model the acquisition of the full complexity of natural language. Note, however, that this limitation applies equally well to symbolic approaches to language acquisition (e.g., Anderson, 1983), including parameter-setting models (e.g., Gibson & Wexler, 1994; Niyogi & Berwick, 1996), and other models which assume an innate universal grammar (e.g., Berwick & Weinberg, 1984).

Turning to linguistic issues, the better performance of the SRN on cross-dependency recursion compared with center-embedding recursion may reflect the fact that the difference between learning limited degrees of context-free and context-sensitive structure may be very different from the problem of learning the full, infinite versions of these languages; a similar conclusion with respect to processing is reached by Vogel, Hahn & Branigan (1996) from the viewpoint of formal language computation and complexity. Within the framework of Gibson's (in press) Syntactic Prediction Locality Theory, center-embedded constructions (of depth 2 or less) are harder to process than their cross-dependency counterparts because center-embedding requires holding information in memory over a longer stretch of intervening items than cross-dependency. Put simply, the information about the first noun has to be kept in memory over minimally $\sim 2D$ items for center-embedding, where D corresponds to depth of recursion, but only over minimally $\sim D$ items for cross-dependency. Although a similar kind of analysis is helpful in understanding the difference in SRN performance on the two types of complex recursive constructions, this cannot be the full explanation. Firstly, this analysis incorrectly suggest that singly embedded cross-dependency structures should be easier to process than comparable center-embedded constructions. As illustrated by Figure 9, this is not true of the SRN predictions, nor does it fit with the human data from Bach et al. (1986). Secondly, the above analysis would predict a flat or slightly rising pattern of GPE across the verbs in a sentence with two cross-dependencies. In contrast, the GPE pattern for the cross-dependency sentences (Figure 8) is able to fit the reading time data from Dickey & Vonk (1997) because of a *drop* in the GPE scores for the second verb. Even though there are several details still to be accounted for, the current results suggest that we should be wary of drawing strong conclusions for language processing behavior, in networks and perhaps also in people, from arguments concerning idealized infinite cases.

A related point touches on the architectural requirements for learning languages involving, respectively, context-free and context-sensitive structures. In the simulations reported here,

the same network (with initial random weights held constant across simulations) was able to learn the three different artificial languages to a degree similar to human performance. To our knowledge, no symbolic model has been shown to be able to *learn* these three kinds of recursive structures given *identical initial conditions*. For example, Berwick & Weinberg's (1984) symbolic model of language acquisition has a built-in stack (as well as other architectural requirements for implementing a Marcus-style parser, Marcus,1980) and would therefore not be able to learn languages involving cross-dependencies because the latter are beyond the capacities of simple stack memory structures. It is, of course, true that if one builds a context-sensitive parser then it can also by definition parse context-free strings. However, the processing models that are able to account for the Bach et al. (1986) data (Gibson, in press; Joshi, 1990; Rambow & Joshi, 1994) do not incorporate learning theories specifying how knowledge relevant to the processing of center-embedding and cross-dependency could be acquired. Connectionist networks therefore present a learning-based alternative to the symbolic models because, as we have shown, the same network is able to develop representations necessary for the processing of both center-embedding and cross-dependency structures (as well as counting recursive constructions). Recent simulations involving more natural language-like grammars, incorporating significant additional complexity in terms of left and right recursive structures, suggest that this result is not confined to the learning of the artificial languages presented in this paper (see Christiansen, 1994, 1998, for details).

In this paper, we have presented results showing a close qualitative similarity between the breakdown patterns in human and SRN processing when faced with complex recursive structures. This was achieved without assuming that the language processor has access to a competence grammar which allows unbounded recursion, subject to performance constraints. Instead, the SRN account suggests that the recursive constructions that people actually say and hear may be explained by a system in which there is no representation of unbounded grammatical competence, and performance limitations arise from intrinsic constraints on the processing system. If this hypothesis is correct, then the standard distinction between competence and performance, which is at the center of contemporary linguistics, may need to be rethought.

References

Allen, J. & Christiansen, M.H. (1996). Integrating multiple cues in word segmentation: A connectionist model using hints. In *Proceedings of the Eighteenth Annual Cognitive Science Society Conference* (pp. 370–375). Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, J.R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Bach, E., Brown, C. & Marslen-Wilson, W. (1986). Crossed and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes*, **1**, 249–262.

Batali, J. (1994). Artificial evolution of syntactic aptitude. In *Proceedings from the Sixteenth Annual Conference of the Cognitive Science Society* (pp. 27–32). Hillsdale, NJ: Lawrence Erlbaum Associates.

Berwick, R.C & Weinberg, A.S (1984). *The grammatical basis of linguistic performance: Language use and acquisition.* Cambridge, MA: MIT Press.

Blaubergs, M.S. & Braine, M.D.S. (1974). Short-term memory limitations on decoding self-embedded sentences. *Journal of Experimental Psychology*, **102**, 745–748.

Brill, E. Magerman, D. Marcus, M. & Santorini, B. (1990). Deducing linguistic structure from the statistics of large corpora. In *DARPA Speech and Natural Language Workshop.* Hidden Valley, Pennsylvania: Morgan Kaufmann.

Bullinaria, J.A. (1994). Internal representations of a connectionist model of reading aloud. In *Proceedings from the Sixteenth Annual Conference of the Cognitive Science Society* (pp. 84–89). Hillsdale, NJ: Lawrence Erlbaum Associates.

Chalmers, D.J. (1990). Syntactic transformations on distributed representations. *Connection Science*, **2**, 53–62.

Charniak, E. (1993). *Statistical language learning.* Cambridge, MA: MIT Press.

Chater, N. & Conkey, P. (1992). Finding linguistic structure with recurrent neural networks. In *Proceedings of the Fourteenth Annual Meeting of the Cognitive Science Society* (pp. 402–407). Hillsdale, NJ: Lawrence Erlbaum Associates.

Chater, N. & Oaksford, M. (1990). Autonomy, implementation and cognitive architecture: A reply to Fodor and Pylyshyn. *Cognition*, **34**, 93–107.

Chomsky, N. (1957). *Syntactic structures.* The Hague: Mouton.

Chomsky, N. (1959). Review of Skinner (1957). *Language*, **35**, 26–58.

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

Christiansen, M.H. (1992). The (non)necessity of recursion in natural language processing. In *Proceedings of the Fourteenth Annual Meeting of the Cognitive Science Society* (pp. 665–670). Hillsdale, NJ: Lawrence Erlbaum Associates.

Christiansen, M.H. (1994). *Infinite languages, finite minds: Connectionism, learning and linguistic structure.* Unpublished doctoral dissertation, University of Edinburgh.

Christiansen, M.H. (1998). *Intrinsic constraints on the processing of recursive sentence structure.* Manuscript in preparation, University of Southern California.

Christiansen, M.H., Allen, J. & Seidenberg, M.S. (in press). Learning to segment speech using multiple cues: A connectionist model. *Language and Cognitive Processes*

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

Christiansen, M.H. & MacDonald, M.C. (1998). *Processing of recursive sentence structure: Testing predictions from a connectionist model.* Manuscript in preparation, University of Southern California.

Church, K. (1982). *On memory limitations in natural language processing.* Bloomington, IN: Indiana University Linguistics Club.

Cleeremans, A., Servan-Schreiber, D. & McClelland, J. L. (1989). Finite state automata and simple recurrent networks. *Neural Computation*, **1**, 372–381.

Cliff, N. (1987). *Analyzing multivariate data.* Orlando, FL: Harcourt Brace Jovanovich.

Cottrell, G.W. & Plunkett, K. (1991). Learning the past tense in a recurrent network: Acquiring the mapping from meanings to sounds. In *Proceedings of the Thirteenth Annual Meeting of the Cognitive Science Society* (pp. 328–333). Hillsdale, NJ: Lawrence Erlbaum Associates.

Dell, G.S., Chang, F. & Griffin, Z.M. (in press). Connectionist models of language production: Lexical access and grammatical encoding. *Cognitive Science.*

Dickey, M.W. & Vonk, W. (1997). Center-embedded structures in Dutch: An on-line study. Poster presented at the Tenth Annual CUNY Conference on Human Sentence Processing. Santa Monica, CA, March 20–22.

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

Elman, J.L. (1991). Distributed representation, simple recurrent networks, and grammatical structure. *Machine Learning*, **7**, 195–225.

Elman, J.L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, **48**, 71–99.

Fanty, M. (1985). *Context-free parsing in connectionist networks* (Tech. Rep. No. TR-174). Rochester, NY: University of Rochester, Department of Computer Science.

Foss, D.J. & H.S. Cairns (1970). Some effects of memory limitations upon sentence comprehension and recall. *Journal of Verbal Learning and Verbal Behavior*, **9**, 541–547.

Frazier, L. & Fodor, J.D. (1978). The sausage machine: A new two stage parsing model. *Cognition*, **6**, 291-325.

Gale, W. & Church, K. (1990). Poor estimates of context are worse than none. In *Proceedings of the June 1990 DARPA Speech and Natural Language Workshop.* Hidden Valley, PA.

Gazdar, G. & Pullum, G.K. (1985). *Computationally relevant properties of natural languages and their grammars* (Tech. Rep. No. CSLI-85-24). Palo Alto, CA: Stanford University, Center for the Study of Language and Information.

Gibson, E. (in press). Linguistic complexity: Locality of syntactic dependencies. *Cognition.*

Gibson, E. & Thomas, J. (1997). *Memory limitations and structural forgetting: The perception of complex ungrammatical sentences as grammatical.* Unpublished manuscript, Cambridge, MA: MIT.

Gibson, E. & Thomas, J. (1996). The processing complexity of English center-embedded and self-embedded structures. In C. Schütze (Ed.) *Proceedings of the NELS 26 sentence processing workshop* (pp. 45-71). Cambridge, MA: MIT Occasional Papers in Linguistics 9.

Gibson, E. & Wexler, K. (1994). Triggers. *Linguistic Inquiry*, **25**, 407–454.

Giles, C. & Omlin, C. (1993). Extraction, insertion and refinement of symbolic rules in dynamically driven recurrent neural networks. *Connection Science*, **5**, 307–337.

Giles, C., Miller, C., Chen, D., Chen, H., Sun, G., & Lee, Y. (1992). Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Computation*, **4**, 393–405.

Green, D.M. & Swets, J.A. (1966). *Signal detection theory and psychophysics.* New York: Wiley.

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

Hanson, S.J. & Kegl, J. (1987). PARSNIP: A connectionist network that learns natural language grammar from exposure to natural language sentences. In *Proceedings of the Eight Annual Meeting of the Cognitive Science Society* (pp. 106–119). Hillsdale, NJ: Lawrence Erlbaum Associates.

Jelinek, F., Lafferty, J.D. & Mercer, R.L. (1990). *Basic methods of probabilistic context-free grammars* (Tech. Rep. RC 16374 (72684)). Yorktown Heights, NY: IBM.

Joshi, A.K. (1990). Processing crossed and nested dependencies: An automaton perspective on the psycholinguistic results. *Language and Cognitive Processes*, **5**, 1–27.

Just, M.A. & Carpenter, P.A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, **99**, 122-149.

Kimball, J. (1973). Seven principles of surface structure parsing in natural language. *Cognition*, **2**, 15–47.

King, J. & Just, M.A. (1991). Individual differences in syntactic processing: The role of working memory. *Journal of Memory and Language*, **30**, 580–602.

Kolen, J.F. (1994). The origin of clusters in recurrent neural network state space. In *Proceedings from the Sixteenth Annual Conference of the Cognitive Science Society* (pp. 508–513). Hillsdale, NJ: Lawrence Erlbaum Associates.

Larkin, W. & Burns, D. (1977). Sentence comprehension and memory for embedded structure. *Memory & Cognition*, **5**, 17–22.

Luce, D. (1959). *Individual choice behavior.* New York: Wiley.

MacDonald, M.C. & Christiansen, M.H. (1998). *Individual differences without working memory: A reply to Just & Carpenter and Waters & Caplan.* Manuscript submitted for publication.

Marcus, M. (1980). *A theory of syntactic recognition for natural language*. Cambridge, MA: MIT Press.

Marks, L.E. (1968). Scaling of grammaticalness of self-embedded English sentences. *Journal of Verbal Learning and Verbal Behavior*, **7**, 965–967.

McClelland, J.L. & Kawamoto, A.H. (1986). Mechanisms of sentence processing. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing, Vol. 2* (pp. 272–325). Cambridge, MA.: MIT Press.

Miller, G.A. (1962). Some psychological studies of grammar. *American Psychologist*, **17**, 748–762.

Miller, G.A. & Isard, S. (1964). Free recall of self-embedded English sentences. *Information and Control*, **7**, 292–303.

Miyata, Y., Smolensky, P., & Legendre, G. (1993). Distributed representation and parallel distributed processing of recursive structures. In *Proceedings of the Fifteenth Annual Meeting of the Cognitive Science Society* (pp. 759–764). Hillsdale, NJ: Lawrence Erlbaum.

Mozer, M.C. & Soukup, T. (1991). Connectionist music composition based on melodic and stylistic constraints. In R.P. Lippmann, J.E. Moody & D.S. Touretzky (Eds.), *Advances in Neural Information Processing Systems 3* (pp. 789–796). San Mateo, CA: Morgan-Kaufmann.

Newell, A. and Simon, H.A. (1976). Computer science as empirical inquiry. *Communications of the ACM*, **19**, 113–126.

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

Niyogi, P. & Berwick, R.C. (1996). A language learning model for finite parameter spaces. *Cognition*, **61**, 161–193.

Norris, D.G. (1990). A dynamic net model of human speech recognition. In G. Altmann (Ed.), *Cognitive models of speech processing: Psycholinguistic and cognitive perspectives*. Cambridge, Mass.: MIT Press.

Pollack, J.B. (1988). Recursive auto-associative memory: Devising compositional distributed representations. In *Proceedings of the Tenth Annual Meeting of the Cognitive Science Society* (pp. 33–39). Hillsdale, NJ: Lawrence Erlbaum Associates.

Pollack, J.B. (1990). Recursive distributed representations. *Artificial Intelligence*, **46**, 77–105.

Pullum, G.K. & Gazdar, G. (1982). Natural languages and context-free languages. *Linguistics and Philosophy*, **4**, 471–504.

Pulman, S.G. (1986). Grammars, parsers, and memory limitations. *Language and Cognitive Processes*, **2**, 197–225.

Rambow, O. & Joshi, A.K. (1994). A processing model for free word-order languages. In C. Clifton, L. Frazier & K. Rayner (Eds.), *Perspectives on sentence processing* (pp. 267–301). Hillsdale, NJ: Lawrence Erlbaum Associates.

Redington, M., Chater, N. & Finch, S. (in press). Distributional information: A powerful cue for acquiring syntactic categories. *Cognitive Science.*

Reich, P. (1969). The finiteness of natural language. *Language*, **45**, 831–843.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986). *Learning internal representations by error propagation.* In McClelland, J.L. & Rumelhart, D.E. (Eds.) *Parallel distributed processing, Vol. 1.* (pp. 318–362). Cambridge, MA: MIT Press.

Schütze, C.T. (1996). *The empirical base of linguistics: Grammaticality judgments and linguistic methodology.* Chicago, IL: The University of Chicago Press.

Servan-Schreiber, D., Cleeremans, A. & McClelland, J. L. (1991). Graded state machines: The representation of temporal contingencies in simple recurrent networks. *Machine Learning*, **7**, 161–193.

Shieber, S. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, **8**, 333–343.

Shillcock, R., Levy, J. & Chater, N. (1991). A connectionist model of word recognition in continuous speech. In *Proceedings from the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 340–345). Hillsdale, NJ: Lawrence Erlbaum Associates.

Small, S.L., Cottrell, G.W. & Shastri, L. (1982). Towards connectionist parsing. In *Proceedings of the National Conference on Artificial Intelligence.* Pittsburgh, PA.

Smolensky, P. (in press). Grammar-based connectionist approaches to language. *Cognitive Science.*

Stabler, E.P. (1994). The finite connectivity of linguistic structure. In C. Clifton, L. Frazier & K. Rayner (Eds.), *Perspectives on sentence processing* (pp. 303–336). Hillsdale, NJ: Lawrence Erlbaum Associates.

Steijvers, M. & Grünwald, P. (1996). A recurrent network that performs a context-sensitive prediction task. In *Proceedings from the Eighteenth Annual Conference of the Cognitive Science Society* (pp. 335–339). Mahwah, NJ: Lawrence Erlbaum Associates.

Stolcke, A. (1991). Syntactic category formation with vector space grammars. In *Proceedings from the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 908–912). Hillsdale, NJ: Lawrence Erlbaum Associates.

Stolz, W.S. (1967). A study of the ability to decode grammatically novel sentences. *Journal of Verbal Learning and Verbal Behavior*, **6**, 867–873.

Vogel, C., Hahn, U. & Branigan, H. (1996). Cross-serial dependencies are not hard to process. In *Proceedings of COLING-96, The 16th International Conference on Computational Linguistics* (pp. 157–162), Copenhagen, Denmark.

Wanner, E. (1980). The ATN and the sausage machine: Which one is baloney? *Cognition*, **8**, 209–225.

Weckerly, J. & Elman, J. (1992). A PDP approach to processing center-embedded sentences. In *Proceedings of the Fourteenth Annual Meeting of the Cognitive Science Society* (pp. 414–419). Hillsdale, NJ: Lawrence Erlbaum Associates.

Wiles, J. & Bloesch, A. (1992). Operators and curried functions: Training and analysis of simple recurrent networks. In J.E. Moody, S.J. Hanson & R.P. Lippmann (Eds.), *Advances in Neural Information Processing Systems 4*. San Mateo, CA: Morgan-Kaufmann.

Wiles, J. & Elman, J. (1995). Learning to count without a counter: A case study of dynamics and activation landscapes in recurrent networks. In *Proceedings of the Seventeenth Annual Meeting of the Cognitive Science Society* (pp. 482–487). Hillsdale, NJ: Lawrence Erlbaum Associates.

Wiles, J. & Ollila, M. (1993). Intersecting regions: The key to combinatorial structure in hidden unit space. In S.J. Hanson, J.D. Cowan & C.L. Giles (Eds.), *Advances in Neural Information Processing Systems 5*, (pp. 27–33)). San Mateo, CA: Morgan-Kaufmann.

7 Author Notes

We would like to thank James Greeno, Paul Smolensky and an anonymous reviewer for their valuable comments on an earlier version of this manuscript, and Joe Allen, Jim Hoeffner and Mark Seidenberg for discussions of the issues involved.

TABLE 1
The Distribution of Embedding Depths in Training and Test Corpora

	Embedding Depth			
Recursion Type	0	1	2	3
Complex	15%	27.5%	7%	.5%
Right-Branching	15%	27.5%	7%	.5%
Total	30%	55%	14%	1%

Note. The precise statistics of the individual corpora varied slightly from this ideal distribution.

TABLE 2
Percentage of Cases Correctly Classified given Discriminant Analyses of Network Hidden Unit Representations

	Recursion Type			
	Separation Along Singular/Plural Noun Categories		Separation Across Singular/Plural Noun Categories	
Noun Position	Complex	Right-Branching	Complex	Right-Branching
	Before Training			
First	62.60	52.80	57.62	52.02
Middle	97.92	94.23	89.06	91.80
Last	100.00	100.00	100.00	100.00
Random	56.48	56.19	55.80	55.98
	After Training			
First	96.91	73.34	65.88	64.06
Middle	92.03	98.99	70.83	80.93
Last	99.94	100.00	97.99	97.66
Random	55.99	55.63	54.93	56.11

Notes. Noun position denotes the left-to-right placement of the noun being tested, with Random indicating a random assignment of the vectors into two groups.

TABLE 3
The Distribution of Grammatical Sentences Generated by the Nets Trained on the Three Languages

	Language					
	Counting		Center-		Cross -	
Construction	Recursion		Embedding		Dependency	
Depth 0	nv	(16)	nv	(15)	nv	(15)
	NV	(8)	NV	(14)	NV	(12)
Complex	*NNVV*	(11)	nNVv	(7)	nnvv	(5)
Recursion	*NNNVVV*	(6)	nnvv	(6)	NnVv	(5)
			NNVV	(5)	NNVV	(5)
			NnvV	(2)	nNvV	(3)
					nnnvvv	(1)
					nNnvVv	(1)
					NnnVvv	(1)
Right-	NVnv	(8)	NVNV	(7)	nvNV	(7)
Branching	NVNV	(6)	nvnv	(5)	nvnv	(6)
Recursion	nvNV	(4)	nvNV	(4)	NVnv	(5)
	nvnv	(2)	NVnv	(1)	NVNV	(5)
	nvNVnv	(2)	nvNVnv	(1)	nvnvnv	(1)
	nvnvnv	(1)	NVNVnv	(1)	NVNVNV	(1)
	nvnvNV	(1)	NVnvNV	(1)		
	NVnvnv	(1)				
	NVNVNV	(1)				

Notes. The number of instances of each construction is indicated in parentheses. Capitalization indicates plural agreement—except in the case of complex recursive structures generated by the SRN trained on the counting recursion language where the letters stand for both singular and plural. The end of sentence marker ('#') is omitted for expositional purposes.

TABLE 4
The Distribution of Ungrammatical Strings Generated by the Nets Trained on the Three Languages

	Language					
Construction	Counting Recursion		Center-Embedding		Cross - Dependency	
Depth 0	Nv	(3)				
Complex Recursion	nnvVV	(1)	NNVv	(3)	nnvvV	(1)
	nnnVnv	(1)	nnvvv	(2)	nnvnvVv	(1)
	nnVvV	(1)	NNVNVv	(2)	nnvVvNV	(1)
	nNv	(1)	nnvvV	(1)	nnv	(1)
	nNVnv	(1)	nNVvv	(1)	nnNvVv	(1)
	nNNVv	(1)	nNVV	(1)	nNnVv	(1)
	NnvVV	(1)	Nnvnvvv	(1)	nNNVvv	(1)
	NnVVV	(1)	NnvVV	(1)	NnVvn	(1)
	NNnvV	(1)	Nnv	(1)	NnVVv	(1)
	NnvVnv	(1)	NnNV	(1)	NnVNvVVnN	(1)
			NNnvvnvv	(1)	NNVNV	(1)
			NNnv	(1)	NNV	(1)
			NNVnv	(1)	nNnNvNvVVNVNVNV	(1)
			NNVNV	(1)		
			nNNVV	(1)		
Right-Branching Recursion	nvV	(3)	nvV	(2)	NVV	(4)
	nvv	(2)	NVv	(2)	nvvV	(1)
	nvNv	(2)	nvv	(1)	nvnvnV	(1)
	NVv	(2)	nvnvVV	(1)	nVnv	(1)
	NVV	(2)	nVV	(1)	nVNvVV	(1)
	nvNVVV	(1)	Nvv	(1)	NvnVvv	(1)
	nVnVNV	(1)	NVNNVV	(1)	NvVV	(1)
	nVNVNVNV	(1)	NVNNNVVV	(1)	NVvV	(1)
	nVNV	(1)			NVv	(1)
	NVvv	(1)			nvnNvV	(1)
	NVvV	(1)			NVnnvv	(1)
Other	vnvNVNV	(1)	{}	(1)		
	vnvNV	(1)	Vnv	(1)		
	V	(1)				

Notes. The number of instances of each construction is indicated in parentheses. Capitalization indicates plural agreement. The end of sentence marker ('#') is omitted for expositional purposes.

Figure Captions

Figure 1: A recursive set of phrase structure rules which can be used to assign syntactic structure to sentences involving right-branching relative clauses.

Figure 2: The basic architecture of a simple recurrent network (SRN). The rectangles correspond to layers of units. Arrows with solid lines denote trainable weights, whereas the arrow with the dashed line denotes the copy-back connections.

Figure 3: The performance for complex recursive structures of nets of different sizes as a function of number of epochs trained on the counting recursion language (top panel), the center-embedding recursion language (middle panel), and the cross-dependency recursion language (bottom panel).

Figure 4: The performance for right-branching structures of nets of different sizes as a function of number of epochs trained on the counting recursion language (top panel), the center-embedding recursion language (middle panel), and the cross-dependency recursion language (bottom panel).

Figure 5: The performance averaged across epochs on complex recursive constructions (left panels) and right-branching constructions (right panels) of nets of different sizes as well as the bigram and trigram models trained on the counting recursion language (top panels), the center-embedding recursion language (middle panels), and the cross-dependency recursion language (bottom panels). Error bars indicate the standard error of the mean.

Figure 6: The performance averaged across epochs on doubly embedded complex recursive constructions of nets of different sizes as well as the bigram and trigram models trained on the counting recursion language (top left panel), the center-embedding recursion language (top right panel), and the cross-dependency recursion language (bottom panel). Error bars indicate the standard error of the mean.

Figure 7: The mean grammatical prediction error on complex (C) and right-branching (RB) recursive constructions as a function of embedding depth (0-4). Results are shown for the SRN as well as the bigram and trigram models trained on the counting recursion language (top left panel), the center-embedding recursion language (top right panel), and the cross-dependency recursion language (bottom panel).

Figure 8: Grammatical prediction error for each word in doubly embedded sentences for the net trained on constructions of varying length (SRN), the net trained exclusively on doubly embedded constructions (D2-SRN), and the bigram and trigram models. Results are shown for counting recursion (top panel), center-embedding recursion (middle panel), and cross-dependency recursion (bottom panel). Subscripts indicate subject noun/verb agreement patterns.

Figure 9: Human performance (from Bach et al., 1986) on singly and doubly center-embedded German (past participle) sentences compared with singly and doubly embedded cross-dependency sentences in Dutch (left panel), and SRN performance on the same kinds of constructions (right panel). Error bars indicate the standard error of the mean.

Figure 10: The mean output activation for the four lexical categories and the end of sentence marker (EOS) given the context 'NNNVV'. Error bars indicate the standard error of the mean.

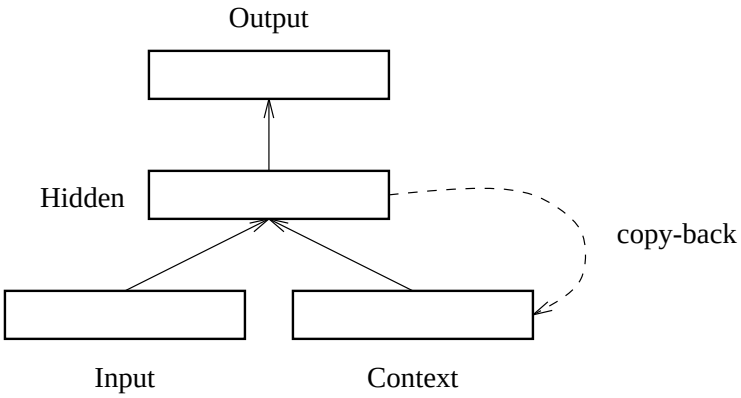
Figure 11: Human ratings (from Christiansen & MacDonald, 1998) for 2VP and 3VP center-embedded English sentences (left ordinate axis) compared with the mean grammatical prediction error produced by the SRN for the same kinds of constructions (right ordinate axis). Error bars indicate the standard error of the mean.

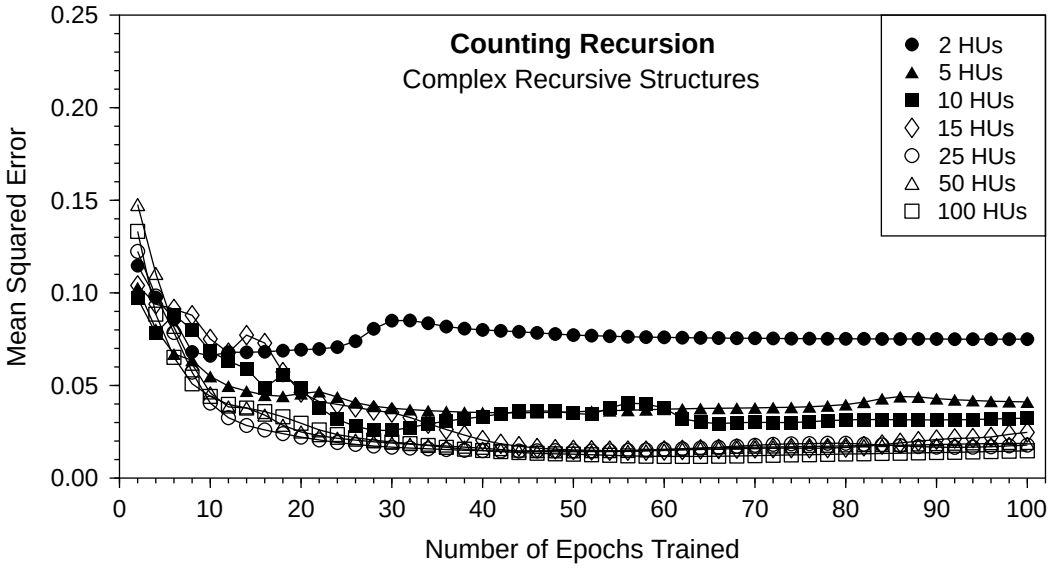
Figure 12: Human comprehensibility ratings (left ordinate axis) from Bach et al. (1996: German past participle paraphrases) compared with the average grammatical prediction error for right-branching constructions produced by the SRN trained on the center-embedding language (right ordinate axis), both plotted as a function of recursion depth.

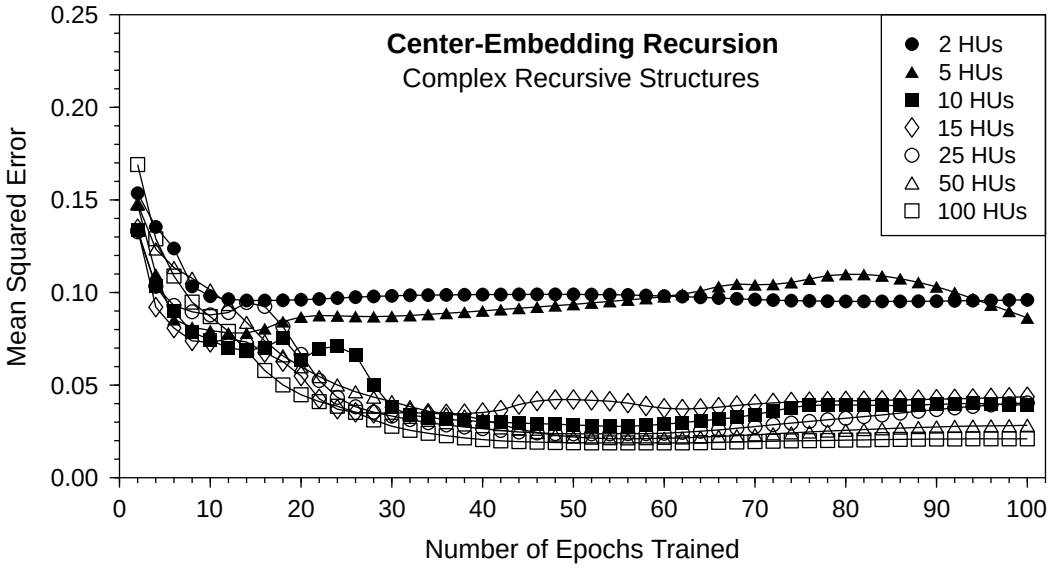
Figure 13: Schematic illustration of hidden unit state space with each of the noun combinations denoting a cluster of hidden unit vectors recorded for a particular set of agreement patterns (with 'N' corresponding to plural nouns and 'n' to singular nouns). The straight dashed lines represent three linear separations of this hidden unit space according the number of (a) the last seen noun, (b) the second noun, and (c) the first encountered noun (with incorrectly classified clusters encircled).

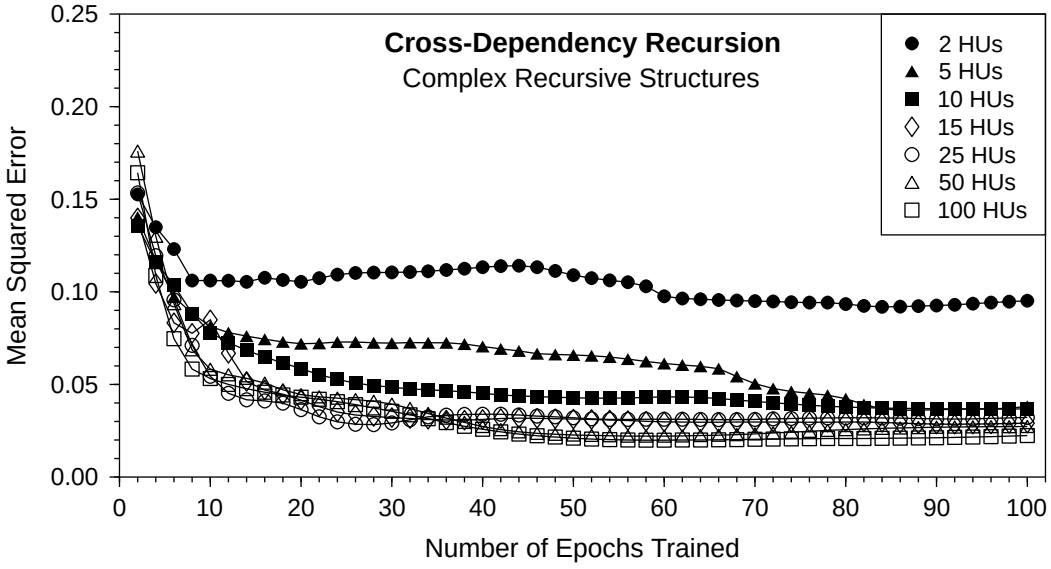
Figure 14: The architecture of a simple recurrent network using a stochastic selection process (SSP) to generate sentences. Arrows with solid lines between the rectangles (corresponding to layers of units) denote trainable weights, whereas the arrow with the dashed line denotes the copy-back connections. The solid arrows to and from the SSP do not denote weights.

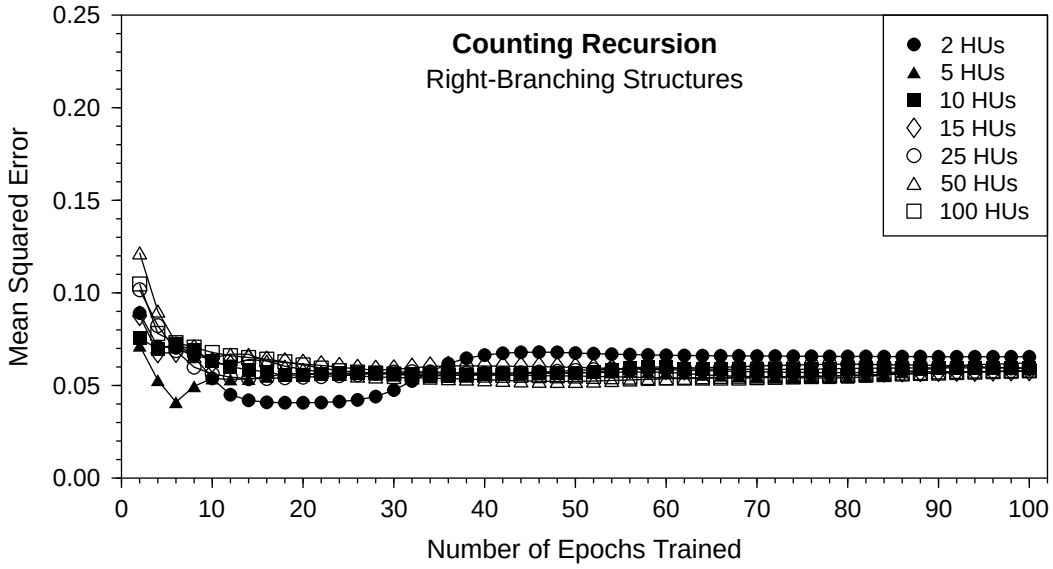
S	$\rightarrow$	NP VP
NP	$\rightarrow$	N (comp S)
VP	$\rightarrow$	V (NP)

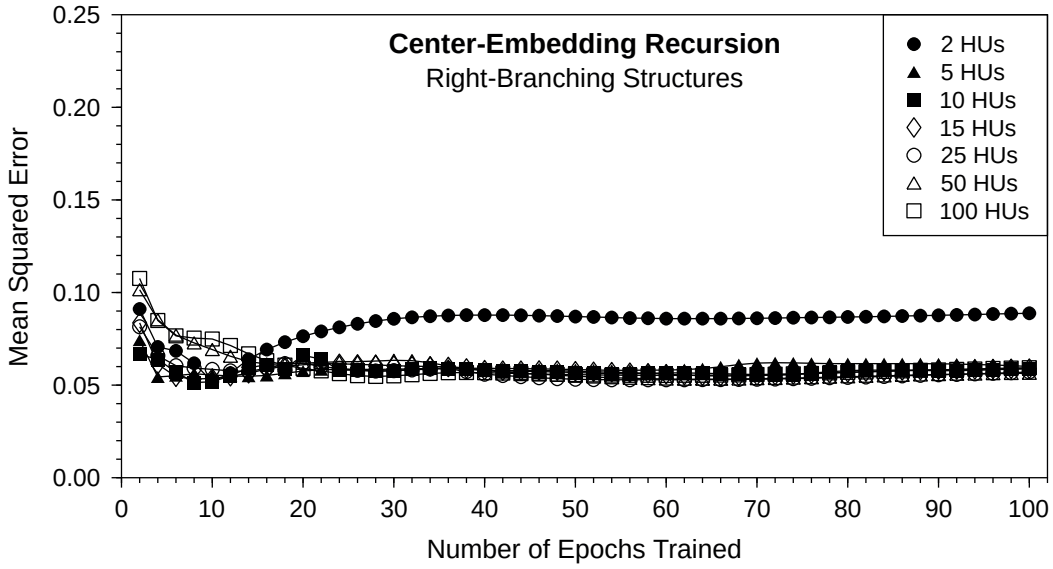

Output
Hidden
copy-back
Input
Context

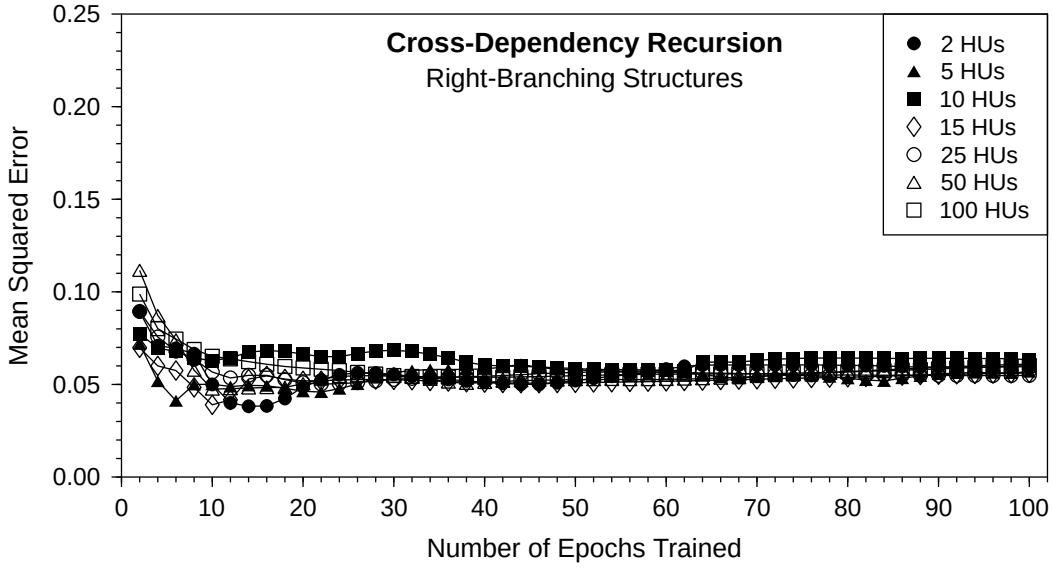

Counting Recursion
Complex Recursive Structures
2 HUs
5 HUs
10 HUs
15 HUs
25 HUs
50 HUs
100 HUs
Mean Squared Error
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60
70
80
90
100
Number of Epochs Trained

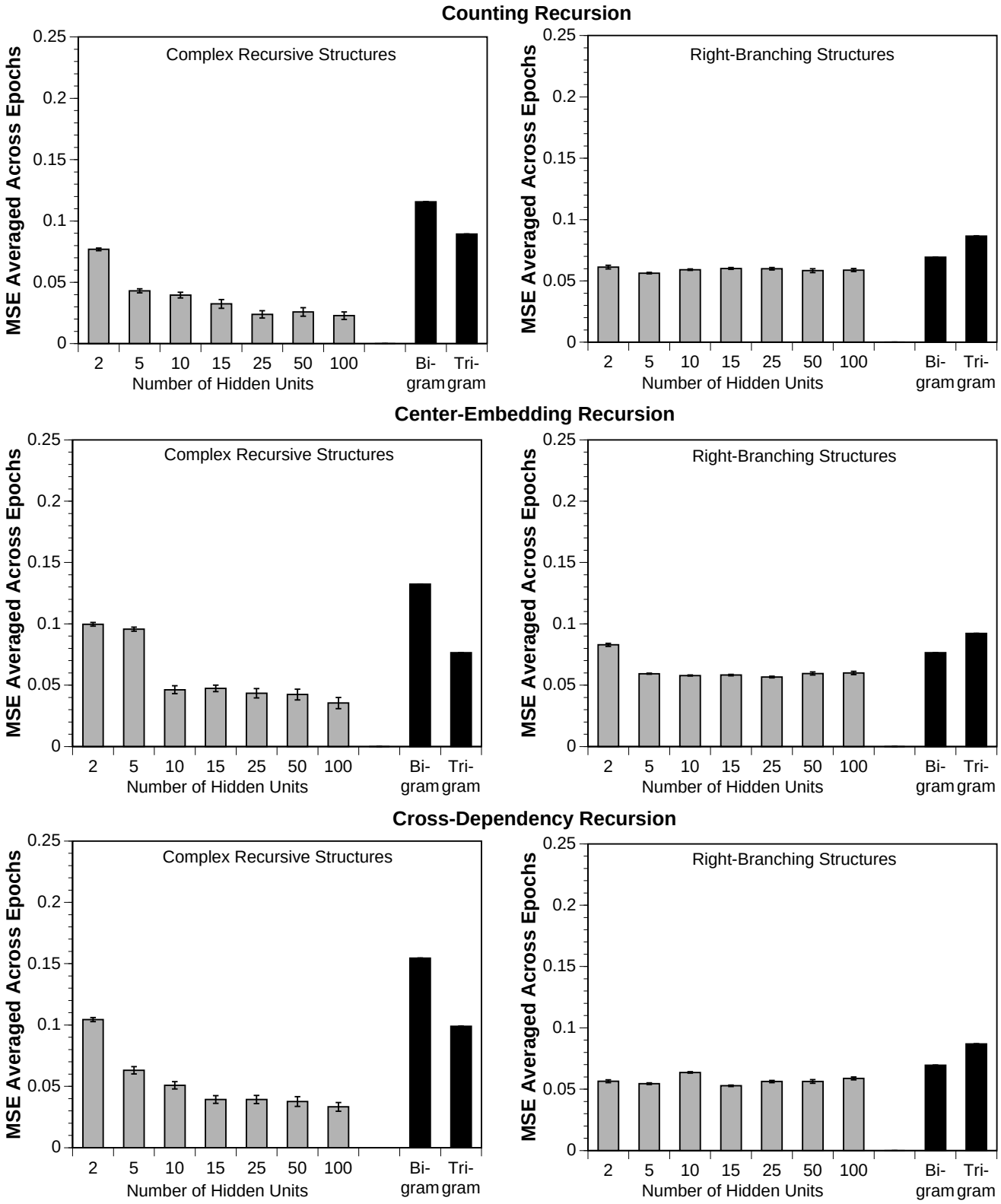

Center-Embedding Recursion
Complex Recursive Structures
2 HUs
5 HUs
10 HUs
15 HUs
25 HUs
50 HUs
100 HUs
Mean Squared Error
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60
70
80
90
100
Number of Epochs Trained

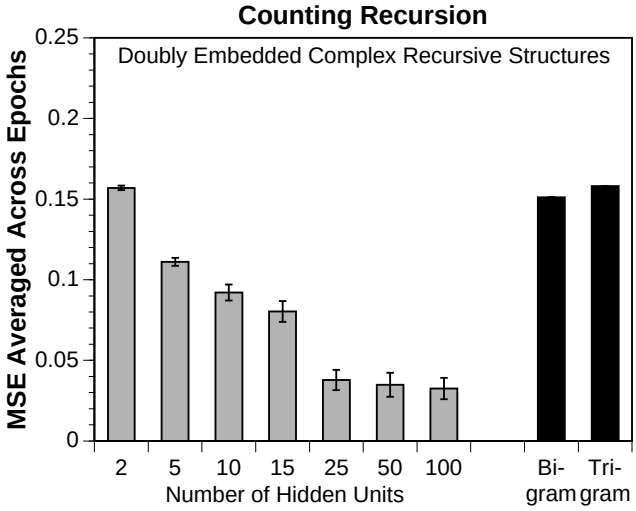

Cross-Dependency Recursion
Complex Recursive Structures
2 HUs
5 HUs
10 HUs
15 HUs
25 HUs
50 HUs
100 HUs
Mean Squared Error
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60
70
80
90
100
Number of Epochs Trained

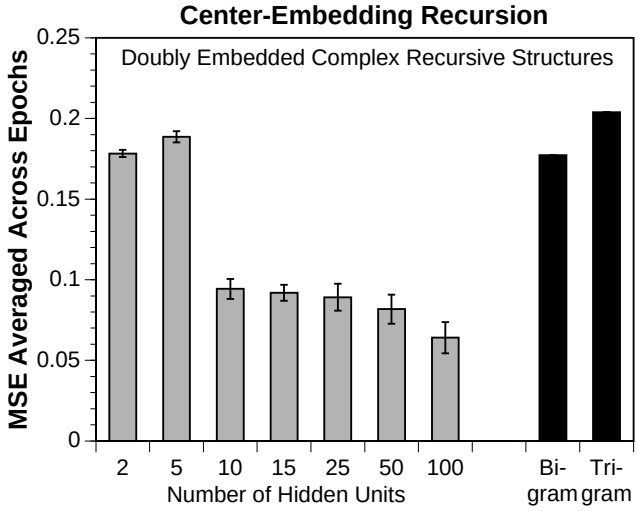

Counting Recursion
Right-Branching Structures
2 HUs
5 HUs
10 HUs
15 HUs
25 HUs
50 HUs
100 HUs
Mean Squared Error
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60
70
80
90
100
Number of Epochs Trained

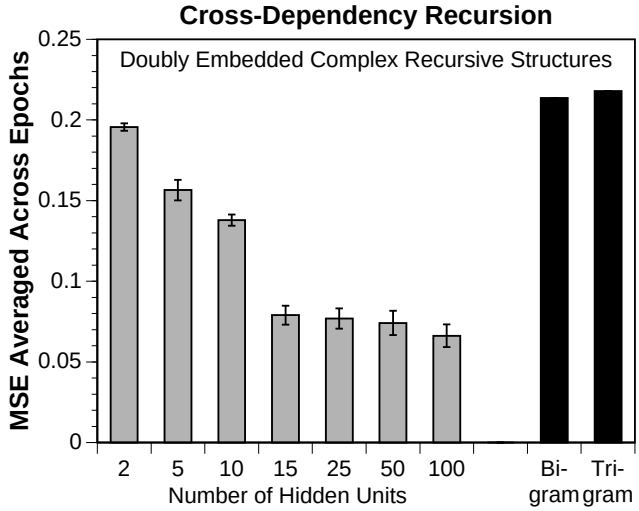

Center-Embedding Recursion
Right-Branching Structures
2 HUs
5 HUs
10 HUs
15 HUs
25 HUs
50 HUs
100 HUs
Mean Squared Error
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60
70
80
90
100
Number of Epochs Trained

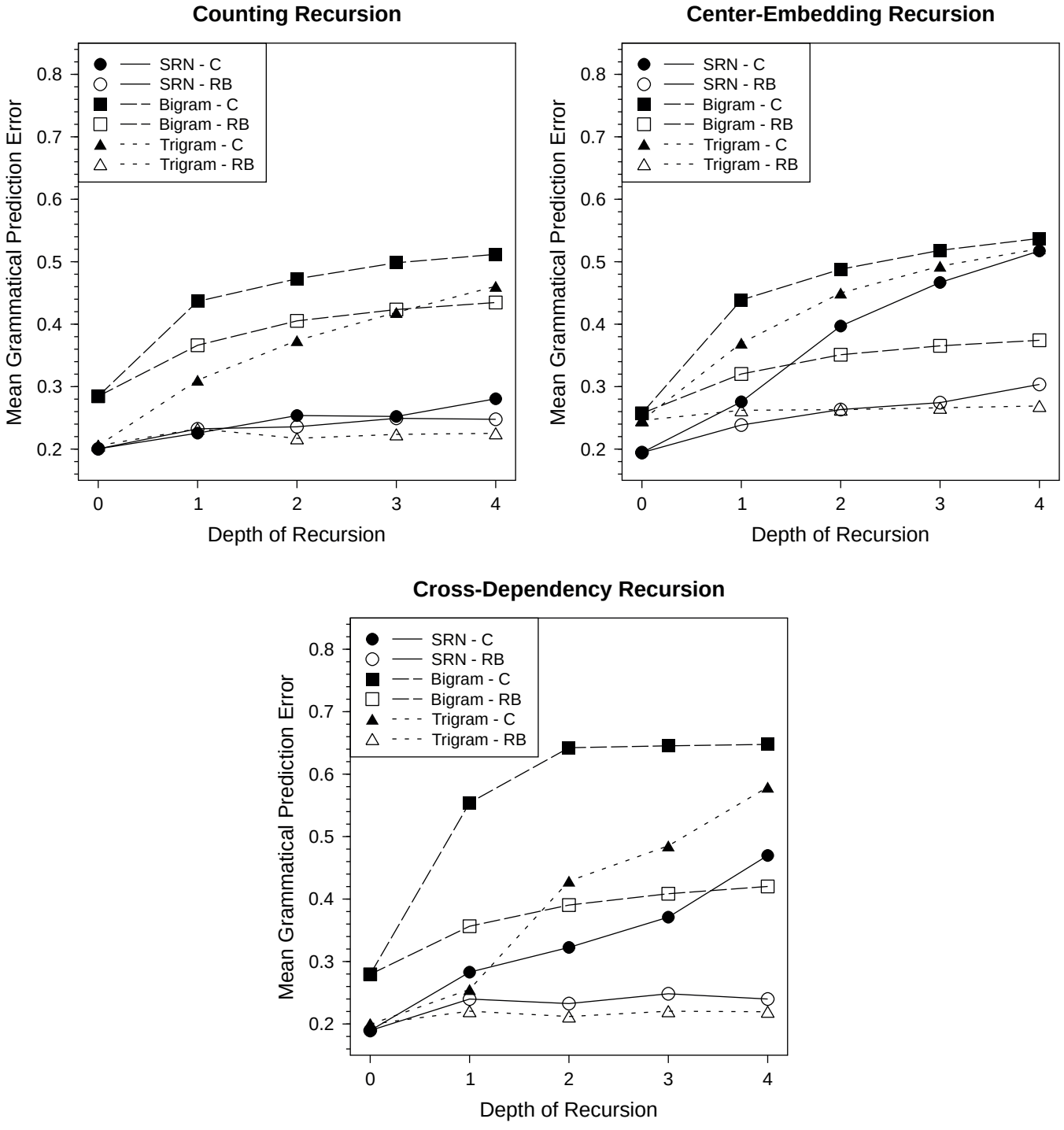

Cross-Dependency Recursion
Right-Branching Structures
2 HUs
5 HUs
10 HUs
15 HUs
25 HUs
50 HUs
100 HUs
Mean Squared Error
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60
70
80
90
100
Number of Epochs Trained

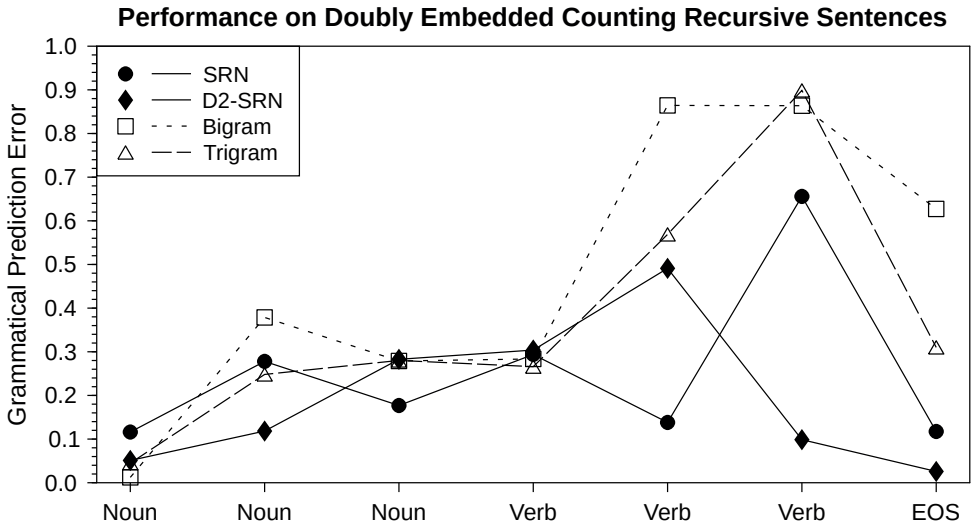

Counting Recursion
Complex Recursive Structures
Right-Branching Structures
Center-Embedding Recursion
Complex Recursive Structures
Right-Branching Structures
Cross-Dependency Recursion
Complex Recursive Structures
Right-Branching Structures
MSE Averaged Across Epochs
0.25
0.2
0.15
0.1
0.05
0
2
5
10
15
25
50
100
Bi-gram
Tri-gram
Number of Hidden Units

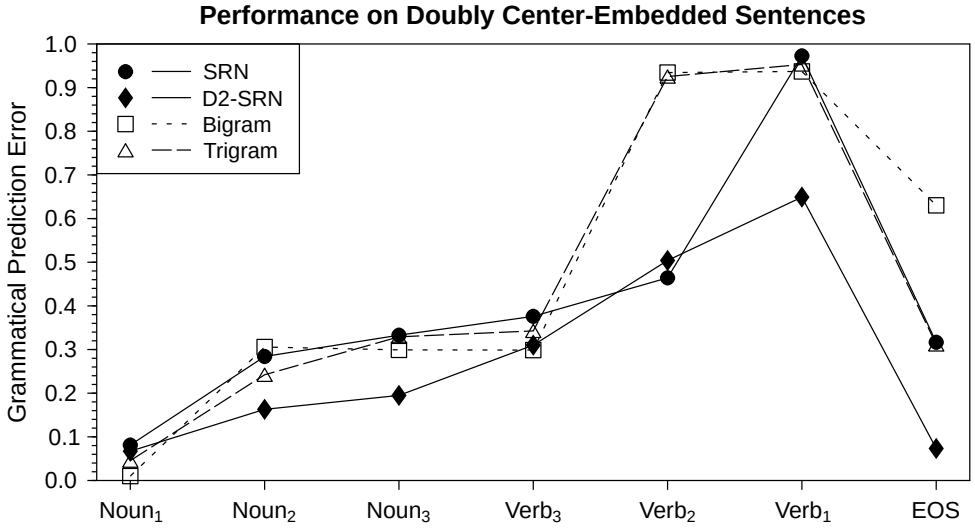

Counting Recursion
Doubly Embedded Complex Recursive Structures
MSE Averaged Across Epochs
0.25
0.2
0.15
0.1
0.05
0
2
5
10
15
25
50
100
Bi-gram
Tri-gram
Number of Hidden Units

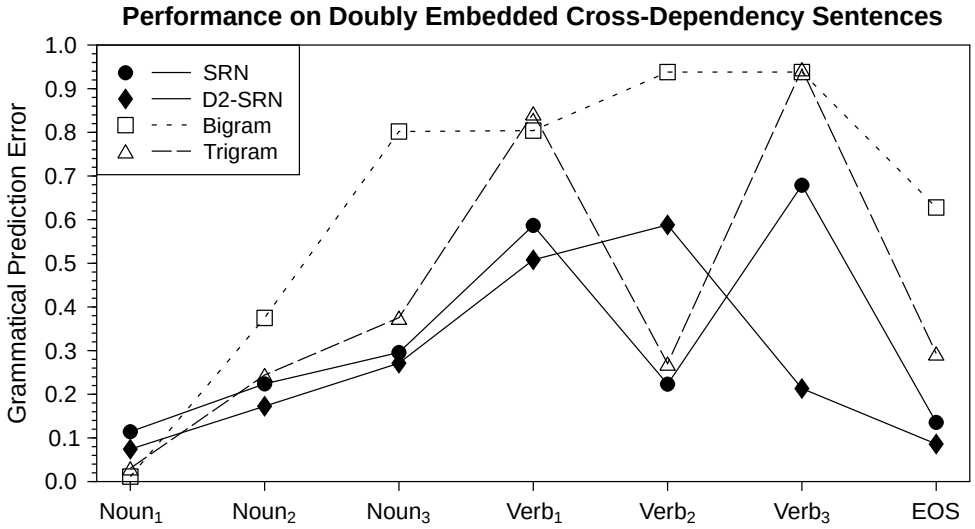

Center-Embedding Recursion
Doubly Embedded Complex Recursive Structures
MSE Averaged Across Epochs
0.25
0.2
0.15
0.1
0.05
0
2
5
10
15
25
50
100
Bi-gram
Tri-gram
Number of Hidden Units

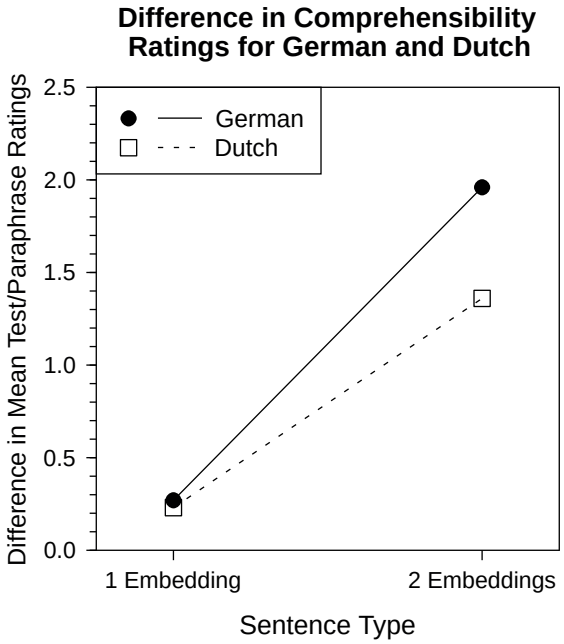

Cross-Dependency Recursion
Doubly Embedded Complex Recursive Structures
MSE Averaged Across Epochs
0.25
0.2
0.15
0.1
0.05
0
2
5
10
15
25
50
100
Bi-gram
Tri-gram
Number of Hidden Units

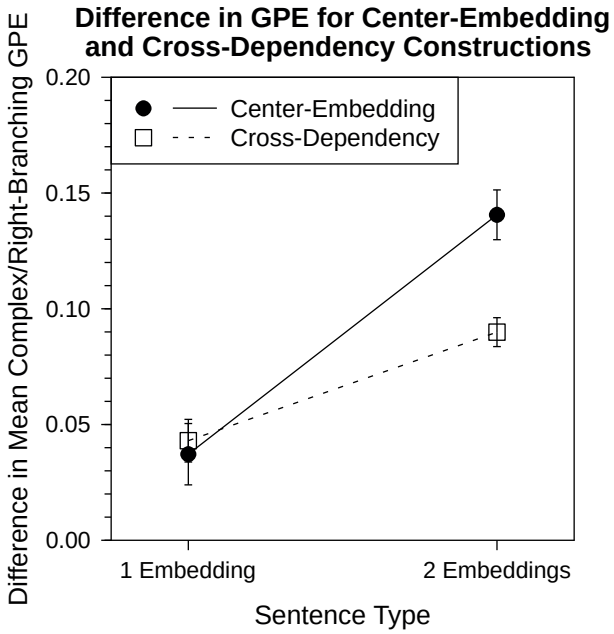

Counting Recursion
Center-Embedding Recursion
Cross-Dependency Recursion
SRN - C
SRN - RB
Bigram - C
Bigram - RB
Trigram - C
Trigram - RB
Mean Grammatical Prediction Error
Depth of Recursion
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
1
2
3
4

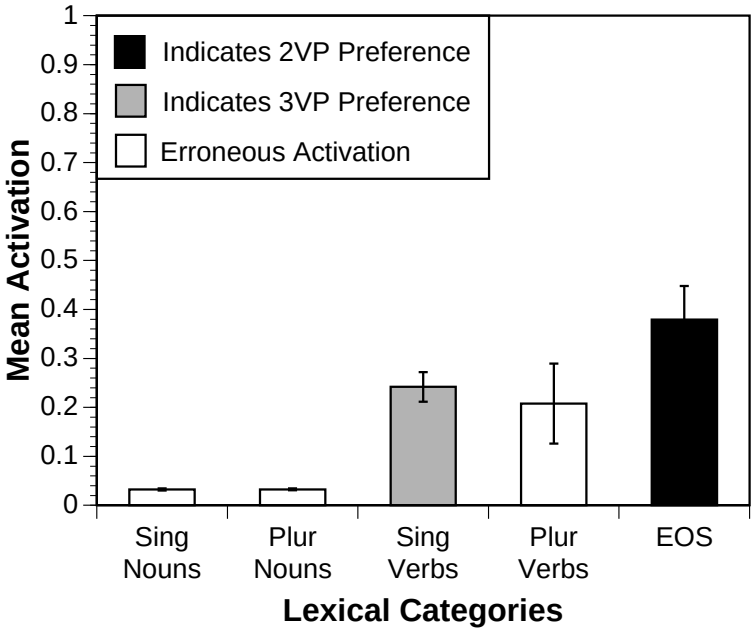

Performance on Doubly Embedded Counting Recursive Sentences
Grammatical Prediction Error
SRN
D2-SRN
Bigram
Trigram
Noun
Noun
Noun
Verb
Verb
Verb
EOS

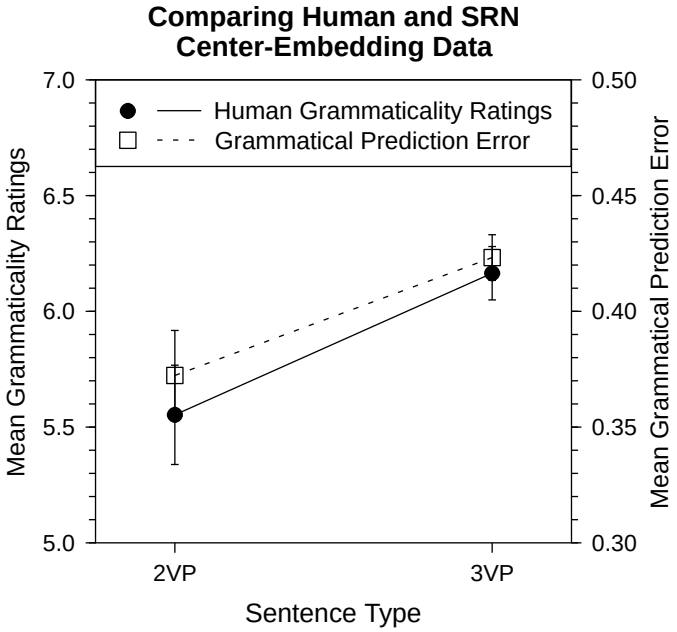

Performance on Doubly Center-Embedded Sentences
Grammatical Prediction Error
SRN
D2-SRN
Bigram
Trigram
Noun1
Noun2
Noun3
Verb3
Verb2
Verb1
EOS

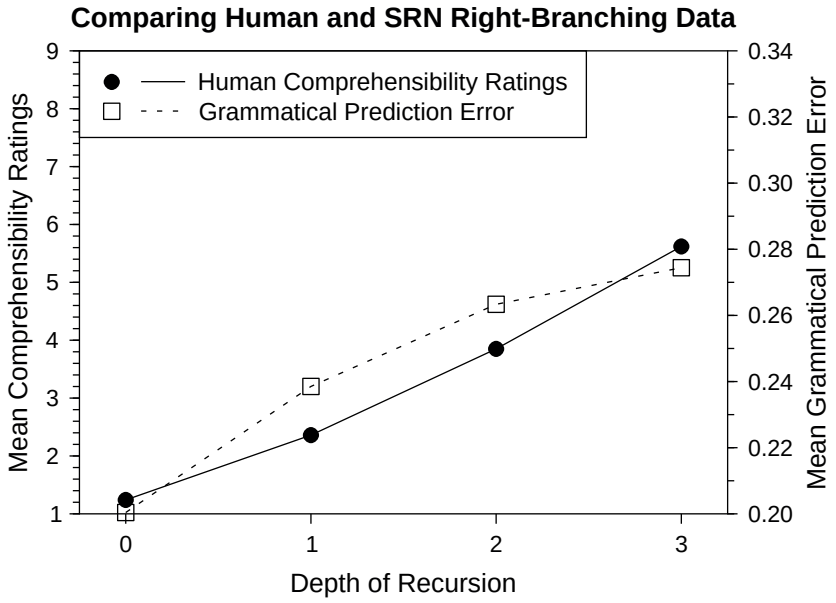

Performance on Doubly Embedded Cross-Dependency Sentences
Grammatical Prediction Error
SRN
D2-SRN
Bigram
Trigram
Noun1
Noun2
Noun3
Verb1
Verb2
Verb3
EOS

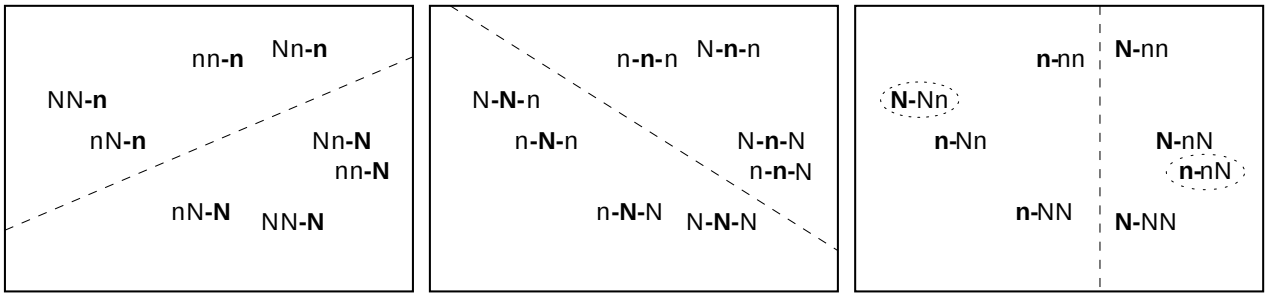

Difference in Comprehensibility Ratings for German and Dutch
Difference in Mean Test/Paraphrase Ratings
2.5
2.0
1.5
1.0
0.5
0.0
German
Dutch
1 Embedding
2 Embeddings
Sentence Type


Difference in GPE for Center-Embedding and Cross-Dependency Constructions
Difference in Mean Complex/Right-Branching GPE
0.20
0.15
0.10
0.05
0.00
Center-Embedding
Cross-Dependency
1 Embedding
2 Embeddings
Sentence Type

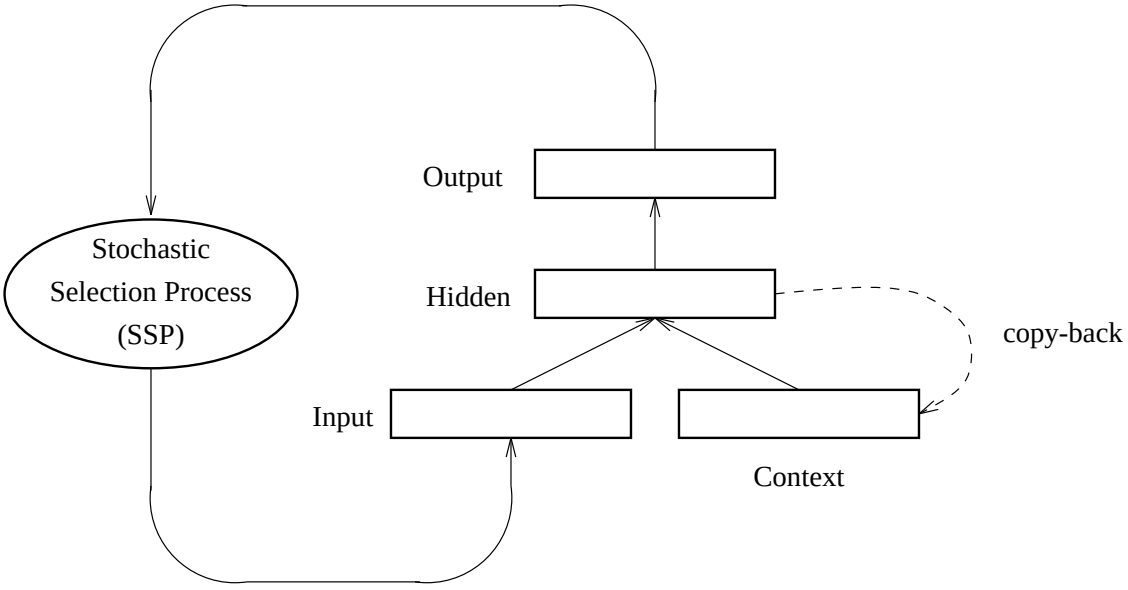

Indicates 2VP Preference
Indicates 3VP Preference
Erroneous Activation
Mean Activation
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Sing Nouns
Plur Nouns
Sing Verbs
Plur Verbs
EOS
Lexical Categories


Comparing Human and SRN
Center-Embedding Data
Human Grammaticality Ratings
Grammatical Prediction Error
Mean Grammaticality Ratings
Mean Grammatical Prediction Error
7.0
6.5
6.0
5.5
5.0
0.50
0.45
0.40
0.35
0.30
2VP
3VP
Sentence Type


Comparing Human and SRN Right-Branching Data
Human Comprehensibility Ratings
Grammatical Prediction Error
Mean Comprehensibility Ratings
Mean Grammatical Prediction Error
Depth of Recursion
9
8
7
6
5
4
3
2
1
0.34
0.32
0.30
0.28
0.26
0.24
0.22
0.20
0
1
2
3


nn-n
Nn-n
NN-n
nN-n
Nn-N
nn-N
nN-N
NN-N
n-n-n
N-n-n
N-N-n
n-N-n
N-n-N
n-n-N
n-N-N
N-N-N
n-nn
N-nn
N-Nn
n-Nn
N-nN
n-nN
n-NN
N-NN

(a) (b) (c)


Output
Stochastic
Selection Process
(SSP)
Hidden
copy-back
Input
Context