A Connectionist Model of Sentence Comprehension and Production

Thesis Proposal

Douglas L. T. Rohde

School of Computer Science, Carnegie Mellon University
and the Center for the Neural Basis of Cognition

July 8, 1999

Abstract

Linguists have historically favored symbolic, rule-based models to explain the human language faculty. Such models typically possess clear explanatory clarity and are well able to handle the apparently recursive structure of natural sentences. In fact, they tend to be too powerful in this respect and ad hoc external constraints and manipulations are often necessary to limit their performance to better match humans' imperfect abilities. Furthermore, due to their reliance on simple rules and structured knowledge, symbolic methods do not lend themselves to learning. A focus on symbolic models and an idealized notion of language processing has contributed to the widespread belief that language is primarily innately specified.

But over the past twenty years there has been increasing interest in connectionist models of language processing. Initially, these were relatively straight-forward implementations of symbolic methods using simple processing units. However, as the representational and learning abilities of connectionist networks were exploited to an increasing extent, connectionist systems have taken on a markedly different character than symbolic models. This has lead to a shift in our understanding of the critical sources of information in language and how they interact in processing and learning. The importance of connectionism is not merely that it links our understanding of cognition to the hardware of the brain but that its study leads to fundamental changes in our understanding of cognition.

It now seems quite reasonable to hypothesize that much of language could be learned by general processing mechanisms exposed to a natural environment. However, current connectionist models have only demonstrated limited abilities in simple domains and there remains considerable room for skepticism. This paper describes the development of a unified connectionist model of sentence comprehension and production. It extends previous work in its use of more sophisticated languages, word-level rather than phrase or word-class inputs, minimal symbolic control structures, and implicit prediction as a training method for production. The current proposal outlines progress to date on the model and discusses how it will be further extended and tested.

1 Introduction

A traditional approach to the study of language has been to focus on idealized models of its structure and to work backward to considerations of how humans might process language. Models arising from such a perspective have tended to assume that normal adult humans possess a processing mechanism that is theoretically able to perform quite perfectly on idealized, arbitrarily complex language but is hindered by incidental limitations on performance, such as memory restrictions. This separation of our purported knowledge of language and our ability to process language in practice has been termed the *competence/performance* distinction (Chomsky, 1957). An additional tenet introduced into the standard theory by Chomsky, is that statistical information, or the relatively frequency of occurrence of various components, has little or no relevance for the linguist.

This perspective has tended to yield language processing models that directly pattern after the competence/performance distinction. Because it now seems natural to characterize syntax using rule-based, algebraic grammars, a standard assumption has been that any system able to process language must be capable of manipulating information in a rule-based or algebraic way, much like a typical computer program. Such *symbolic* models are often able to perform quite well on the task for which they were designed. Performance factors tend to enter into symbolic models by the introduction of explicit hindrances. Although most symbolicists would agree that ultimately the system must be implemented in the neural hardware of the brain, they would argue that this does not significantly affect the symbolic algorithm embodied in the model.

However, while it is certainly an accomplishment to design a model able to solve a known problem, a full account of the language system must explain how it is able to *learn* the language to which it is exposed. It has proved quite difficult to learn the complex rules upon which symbolic models are based from the linguistic inputs to which we believe humans are exposed. Indeed, if statistical information is ignored, there appear to be theoretical limits on what a general mechanism could learn (Gold, 1967). This is compounded by the fact that many "rules" of language permit exceptions and the language to which we are exposed occasionally violates the rules we eventually learn. The apparent difficulty or even impossibility of learning language under the standard perspective, as well as the existence of universal properties of language, has lead to a widespread view that much of language, even very detailed rules governing complex aspects of syntax, is not learned but is innately predetermined (Chomsky, 1975; Pinker, 1989; Crain, 1991). This has been referred to as "Chomsky's wager" (Hahn & Adriaens, 1994) or the *nativist* perspective.

1.1 The connectionist revolution

Over the past twenty years an alternative to the traditional theory of language has been developing. This has grown largely, although not entirely, out of research in *connectionism*. Connectionism seeks to gain an understanding of the brain through the study of the computational properties of large networks of simple, interacting units, which are similar in many ways to neurons (see Medler (1998) for a review of early connectionist work). Some of the first applications of connectionist networks to language were simply localist implementations of symbolic models using simple, connected units.[1] The simplest forms of these networks do not make efficient, parallel use of their resources. Although these models tend to avoid the use of external memory stores, they share many of the properties of symbolic systems and remain a form of connectionism embraced by symbolists.

However, it was then discovered, or perhaps rediscovered, that some leverage could be gained by allowing the units to interact: exciting and inhibiting one-another over time (McClelland & Rumelhart, 1981). Such *interactive-activation* models were able to integrate soft constraints deriving from multiple sources, a process that appears to be important in many cognitive domains and that is not easily captured in rule-based systems. The next major step in the development of connectionism was the realization that concepts need not be represented using localist units but can be distributed patterns of activation over multiple units (Hinton, McClelland, & Rumelhart, 1986). This allows networks to represent graded degrees of similarity between concepts, to represent new concepts without the addition of units, and to generalize information about related concepts in productive ways. This further widened the gap between the behaviors characterizable in symbolic and connectionist systems.

The final major development of connectionism was the invention of learning rules, most importantly *backpropagation*, that allow networks to learn representations intermediate between inputs and outputs (Rumelhart, Hinton, & Williams, 1986). Such so called *hidden representations* are able to capture higher-order structure governing the regularities in the input/output mappings to which the network is exposed. This allows neural networks to perform significantly more complex tasks than simple one-layer networks and extends the range of properties the networks are able to display. For example, a network might pick up on strong regularities in a domain and appear to follow rule-like behavior, as do humans and symbolic models, but without explicitly implementing the rule anywhere in the network (Plaut, Seidenberg, McClelland, & Patterson, 1996). Thus, connectionist networks have begun to reveal alternative explanations for natural behaviors that are quite unlike those deriving from the symbolic tradition.

1.2 The connectionist view of language

The traditional approach to language research has focused largely on the task of parsing; that is, constructing a labeled hierarchical representation of the structure of a sentence, usually reflecting the grammatical rules used to generate it. Symbolic models excel at manipulating such representations, but they struggle to incorporate meaning into sentence processing. It is not easy to capture slight variations in meaning with variables and values. One aspect of language that is problematic in particular is the way in which subtle aspects of word meaning can have a significant influence on the correct syntactic interpretation. Furthermore, symbolic models tend to be brittle when facing slightly ungrammatical sentences or surviving mild damage. These and other problems encountered by the classical view of sentence processing are discussed further in McClelland, St. John, and Taraban (1989).

Human language users do not have quite the same limitations as symbolic models. We show remarkable sensitivity to meaning, being frequently unaware that seemingly clear sentences have multiple incorrect syntactic in-

[1] A *localist* model or representation is one in which individual units correspond directly to particular symbols or concepts. The alternative is a *distributed* representation, a many-to-many mapping in which a concept is represented as a pattern of activation over multiple units which themselves may participate in representing other concepts.

terpretations that might confuse a rule-based parser. And yet, we struggle with other aspects of language, particularly those best modeled by recursive, symbolic grammars. One example is our well-known difficulty understanding multiple center-embeddings such as, "The mouse the cat the dog bit chased ran away." The standard model of sentence processing suggests that we possess a symbolic parsing mechanism augmented by some form of memory store, perhaps in the form of a stack. Our difficulty with structures such as multiple center-embeddings is explained by postulating that the memory store is simply exhausted. Given that the human cortex contains at least 10 billion neurons, it seems suspect that there simply isn't enough capacity to remember three nouns and three verbs to enable these sentences to be parsed, even after training (Blaubergs & Braine, 1974). It is interesting to note that syntax and morphology, the most rule-like aspects of language, are also those that seem hardest for us to learn, both as children and as adult second-language learners.

On the other hand, the connectionist approach appears to offer a perspective that may better characterize the abilities and limitations of human language users. Modern connectionist networks are highly sensitive to the statistics of the environment and the interactions between constituents within it. Distributed representations provide a natural way to encode and manipulate word and utterance meanings. Although connectionist networks are less-well suited to manipulating symbolic information, it may be that these limitations mirror the same difficulties humans experience with deep structures. Furthermore, such restraints are inherent to the processing mechanism of the network, rather than easily manipulated and perhaps ad hoc external constraints imposed on a competent model. The study of connectionist models of language have even lead some to question the importance of truly recursive mechanisms in processing natural language (Christiansen, 1992).

Finally, neural networks have the ability to learn a wide variety of tasks, including many that appear germane to natural language. In contrast to the standard view, which emphasizes idealized, rule-based models and downplays the role of statistics and semantic influences, the connectionist approach focuses on the distributional properties of the input to which the learner is exposed and asks how language can be learned based on that input. While the language system may ultimately develop rule-like behavior, this behavior is not pre-specified but is learned on the basis of the regularities in the language. The development of rule-like mechanisms for solving regular mappings naturally arises in a reasonable learning mechanism that is pressured with limited resources. The rules of language should not be viewed as laws but as information-reducing conveniences.

With this altered focus, we are lead to question the utility of postulating an idealized, perfect language user. According to an alternate view, one is competent at language if one can communicate with others. The problem of language learning is not a matter of developing a perfect grammar but of learning to communicate effectively (Seidenberg & MacDonald, in press). Rather than focusing on the act of parsing sentences, which most humans may not even engage in at any explicit level, there ought to be renewed interest in the tasks in which we know humans engage: comprehending and producing utterances for the purpose of communicating ideas. Tasks such as judging the grammaticality of a sentence or performing lexical decision[2] are not natural to most language users but are built on top of the real language system and, while potentially revealing, are secondary aspects of language use.

Given the often impressive ability of neural networks to learn complex tasks with relatively generic initial structure, many researchers have begun to question the level of innate information that is really necessary for language learning. The old argument that a particular behavior must be innate because we cannot imagine how it could be learned is no longer valid. Many researchers are returning to the initial hypothesis that all aspects of language are learned by the most generic mechanism possible and searching for those mechanisms capable of learning in the environment to which we are exposed. Thus, one outcome of the connectionist revolution is, or ought to be, a renewed interest in accurately characterizing the distributional properties of the language learning environment. A second outcome is a renewed interest in accurately characterizing human language behavior to allow our models to be reasonably evaluated. In the past, our performance on various language tests has frequently been characterized as following very simple rules and the result has been models designed to implement those rules. However, work with connectionist models has lead to the realization that a model trained on one task may show particular behaviors on a related task or under damage that do not directly reflect the mechanisms built into the model. What appear to be rule-like behaviors, may not be exactly rule-like and may not reflect mechanisms that implement those rules in any obvious way.

Three critical questions at this juncture are:

1. What information is available in the language learner's environment?

2. What abilities does the language learner acquire and what behaviors does the learner show in exercising those abilities?

3. Can a system learn those abilities in an appropriate

[2] Lexical decision is the process of deciding whether a particular string of characters or sounds is, in fact, a word.

environment and what properties must the system have to demonstrate the same behaviors as human learners?

Despite decades of research on language, we are far from answers to any of these questions. Having only recently begun to see the true importance of statistical properties of the language environment, we are now faced with an unfortunate lack of sufficient data, particularly on the typical exposure of children in everyday life at various stages of development. Although we have acquired vast knowledge about such tests as priming and lexical decision, we have relatively little information about the major language abilities of comprehension and production.

1.3 The proposed research

Although work to date on connectionist networks has been very revealing and has had a growing impact on the study of language, the skeptic is quite justified in questioning the ability of networks to handle the full complexity of natural language. Largely due to technical limitations, connectionist models until now have mainly focused on peripheral aspects of language (such as operations on single words), have bypassed learning, or have used extremely simple surrogates for natural language. There has been an interest in applying models to the simple tasks on which many psycholinguists have focused, with relatively little work on comprehension and production. The conviction underlying the current proposal is that, to the extent it is technically feasible, we should first and foremost be focusing on the primary language abilities. Models should be trained on the tasks we believe humans are performing in everyday life with exposure to the same sorts of inputs humans experience. Studies of peripheral tasks, such as grammaticality judgement, priming, or lexical decision should be based on a model that has learned the more basic abilities. The behaviors observed in peripheral tasks are likely to have emerged from the mechanisms that have developed to handle communication. Unless we better understand the properties of such mechanisms, our search for explanations of behavior will be less grounded. Therefore, the primary goal of this project is to extend current connectionist models of sentence processing to the tasks of comprehension and production.

As a starting point, however, we begin with the somewhat simpler task of predicting the next word in the sentence. Prediction has been the focus of quite a bit of research in the connectionist community (Elman, 1991; Rohde & Plaut, in press) for several reasons. Learning prediction does not require an additional teacher or training signal: the correct next word is always observable in the input. Online prediction can be quite useful in language processing, helping to disambiguate noisy input or anticipate the speakers' intentions. However, prediction is not a simple task and accurate prediction requires a representational structure of a language equal in power to a grammar for generating the language. Nevertheless, prediction is attractive to connectionist researchers because, among other reasons, networks appear relatively well-suited to learning prediction.

A major proposal embodied in the current work is that prediction learned during the observation of speech may be the primary training mechanism of subsequent production. I suggest that the language user may be constantly engaged in formulating predictions while comprehending or merely observing language. These predictions are guided by some knowledge of the message the speaker is trying to convey, derived either from preceding utterances or experience with the messages likely in the current situation. Prediction in itself may aid comprehension as it sensitizes the learner to the structure of utterances. However, more profoundly, the ability to formulate accurate predictions in a strong message[3] context is very nearly equivalent to the planning abilities required in production. Essentially, the predictor has learned what a reasonable person would say to convey a particular message and has solved a significant proportion of the prediction problem.

The current plan of research will focus on four main areas of language processing—encoding messages, prediction, comprehension, and production—and will attempt to tell a plausible story about how they may interact in a connectionist system. Ultimately, the model should be judged by its ability to account for human behavior both developmentally and in the adult stage. Although significant progress has been made, many technical questions remain to be answered before the model will be ready for evaluation.

1.4 An outline of the current paper

Section 2 of this paper provides a brief overview of connectionist work on sentence processing to date in order to situate the current endeavor in a broader context. Section 3 discusses in greater depth work that I and others have done on the prediction task. This summarizes recent evidence that simple recurrent networks may not, as previously thought, gain significant advantage in starting with simple languages or reduced memory capacity. Section 4 describes how a prediction model can account for the "wanna problem", which is a much discussed linguistic phenomenon regarding the rare situations in English in which the phrase "want to" may not be contracted to "wanna". Although it has been argued that knowledge of this distinction must be innate, it is here demonstrated that

[3] The term *message* is used to refer to the meaning of a sentence. This differs from the sentence itself, which is language specific.

it is possible for a system sensitive to the statistics of its input to acquire knowledge of the distinction from exposure to a reasonable environment.

Section 5 discusses the general framework of the current project and its main goals. It also introduces the simplified English-like language used in the experiments. The next three sections describe the main components of the model. Section 6 discusses the encoding of messages in static representations. Two different approaches are investigated: the use of recursive auto-associative memories or RAAMs (Pollack, 1990) and query networks. Section 7 considers the problem of jointly learning comprehension and prediction using either the RAAM or query method of message encoding. One consideration is the effect prediction may have on comprehension. Section 8 discusses the addition of production to the comprehension and prediction network, although the initial investigations of this portion of the project are not yet complete. The final section reiterates the limitations and goals of the model and outlines the plan for the remainder of this thesis.

2 A Brief Overview of Connectionist Sentence Processing

This review considers the most significant, or at least well known, connectionist models of sentence processing to date. There have been a number of similar efforts in the past (Diederich, 1989; Sharkey & Reilly, 1992; Hahn & Adriaens, 1994; Wermter, Riloff, & Scheler, 1996; Christiansen & Chater, in press-a; Steedman, in press), but the focus here is on models that address the semantic and syntactic issues involved in handling multi-word utterances and will ignore most of the important applications of connectionist networks to other phenomena such as word reading, lexical decision, and past tense formation. The models illustrate a progression from localist implementations of symbolic systems, to systems of interacting localist units, to distributed representations and multi-layer learning rules, to recurrent learning systems. Although the early localist models are discussed, most of the later localist or hybrid symbolic/connectionist systems have been excluded as they typically differ from symbolic systems only in the implementational details. However, a number of hybrid systems are reviewed in Wermter, Riloff, and Scheler (1996). Most sentence processing models are designed to address one of four major language tasks: parsing, comprehension, word prediction, or production. I consider each of these categories in turn.

2.1 Parsing

Parsing, or producing a syntactic, structural description of a sentence from its surface form, is the sentence processing task that has received the most attention from the symbolic community. Indeed, given that the traditional approach to language has minimized all attention to semantics, parsing is one of the few behaviors left that, ostensibly, does not rely on semantics.[4] Thus, it should not be surprising that many of the connectionist parsing systems found in the literature are essentially symbolic models implemented transparently in connectionist hardware. Learning has not played a major role in most of these models for two reasons. Most connectionist parsing models have been localist. This architecture lends itself to hand-designed weight structures but not easy design of effective learning environments. But more critically, learning to produce an explicit parse of a sentence requires, for most systems, training data labeled with correct parsing information. However, few believe that this is actually available to the child so such models are of questionable relevance to human learning.

The first significant proposal for a connectionist model of parsing, Small, Cottrell, and Shastri (1982), does not actually fit the pattern of a transparently symbolic approach. Following McClelland and Rumelhart (1981), this model stresses the important of interaction between syntactic information and semantics or general memory skills. This interactive activation approach contrasts with more standard parsing theories that stress compartmentalism and serial processing (Frazier, 1979; Fodor, 1983). The Small et al. model, implemented in later work (Cottrell, 1985b), is not actually a full parser but is designed for word-sense disambiguation, which is arguably an important sub-task of parsing or comprehension. The model uses localist units to represent lexical items, individual word senses, and case roles. These units excite or inhibit one another through a set of hand-designed connections. Because of this, the model is not easily expandable to larger vocabularies or complex linguistic structures.

Cottrell (1985a) extended the earlier work with the addition of a full-fledged syntactic parsing network. The network can be generated automatically given a grammar, but still requires some weight-tweaking. Concepts are associated with case roles by means of localist *binder* units. Thus, there is a unit for each concept/role pair and they mutually inhibit one another. Units in the syntactic portion of the network represent the non-terminal symbols of the context-free grammar and their interconnections reflect the possible productions in the grammar.

The model is interesting in that it is able to process sentences presented in a temporal sequence and makes use of interacting top-down and bottom-up information. However, it has a number of limitations. As is a common

[4] The extent to which there exists an independent syntactic parsing module has been a matter of considerable debate, but most connectionists are increasingly skeptical of the existence of such a module (McClelland et al., 1989; Seidenberg & MacDonald, in press).

problem with other models that make use of case-roles, the model does not appear capable of handling sentences with multiple verbs. It is also only able to process fixed length sentences and requires constituent recognizers with duplicated and possibly non-connectionist control structures. Finally, some might complain that the model is not guaranteed to settle into a single, coherent interpretation of any sentence.

Several other connectionist parsing models appeared at the same time. Except where noted, they were localist, non-learning models. Because they also use a fixed-size network and a static input representation, rather than temporally coded inputs, these networks are only able to process sentences up to a finite length and often rely on redundant structure, as in the Cottrell (1985a) model. Waltz and Pollack (1985) presented an interactive activation model which differs from other work in that it does not consist of a single network. Rather, the network is generated based on the grammar and the sentence. This network is only able to represent the possible parses of the given sentence. A settling phase allows the network to settle into a particular interpretation. The model has a number of drawbacks, most significant of which is that it is not grammar-general but uses a program to produce a sentence-specific network. The networks also do not process words over time but use a static input representation and thus are not able to produce partial, online interpretations of sentences. Although the implemented model was purely localist, Waltz and Pollack proposed that concepts should not be represented by single nodes but by distributed patterns of "microfeatures", a suggestion that would be adopted in later connectionist modeling.

Fanty (1985, 1994) took a rather different approach. Aiming to produce a network that is deterministic, fast, and guaranteed to work, Fanty devised a way to implement the CYK dynamic-programming, context-free parsing algorithm (Younger, 1967) in a localist network. The network is able to handle sentences up to a fixed length. It essentially contains a unit for every non-terminal paired with each sub-sequence of the input. The network operates in two passes: a bottom-up phase and a top-down phase. In the bottom-up phase, units for increasingly longer sub-sequences become active if their non-terminal could have produced the words in that sub-sequence. In the top-down phase, units that don't fit within a coherent parse are silenced. In the end, only units that participate in a legal parse remain active. Thus, the model does not require an extended period of relaxation. It is interesting because it suggests that language could be parsed by a non-recursive procedure.

However, most natural language sentences have multiple possible syntactic parses and Fanty's basic model is not able to select the appropriate one. Fanty considered one way to bias the model toward selecting shallower parses, but did not entertain the integration of critical semantic information as in other models. The other major limitations of the model are that it can only handle fixed length sentences and it relies on redundant structure. Although the model is not able to learn entire grammars, Fanty discussed how small errors in the model could be corrected through learning. Rager (1992) described a localist model based on Fanty's but designed to handle "extragrammatical", or slightly incorrect, sentences.

Selman and Hirst (1985, 1994) presented a model that differs from other early connectionist parsers in that it uses a variation on the Boltzmann machine (Fahlman, Hinton, & Sejnowski, 1983) with non-deterministic units and a simulated annealing scheme to allow the network to settle gradually into a stable configuration. The rules of a context-free grammar are implemented in the network by means of syntactic binder units that inhibit one another and excite units for symbols that participate together in a production. The use of simulated annealing, while very slow, allows the network to settle into the correct parse with high probability. However, as in other localist models, this model requires sentences to be bounded in length and uses redundant structure. Due to the proliferation of binder units, the size of the network may grow intractable with more complex grammars. Furthermore, although it was suggested as the next step, this model does not incorporate semantic information and it is not clear how it would deal with syntactic ambiguity.

Charniak and Santos (1987) described another localist parsing model that differs from the others in that it uses a sliding input window. This allows the network to theoretically handle sentences of unbounded length but hinders the ability of the model to process long-distance dependencies, such as those surrounding center-embeddings. Although the model was successfully implemented for a very simple grammar, it is not clear that its parsing heuristics would be sufficient to handle more complex grammars. The model also uses parts of speech rather than lexical inputs and was thus clearly unable to incorporate semantics or resolve syntactic ambiguities.

Howells (1988) described an interactive activation parser known as VITAL. Like Waltz and Pollack (1985), Howells' networks were generated during parsing. However, it appears that the model could have been implemented as a single network. Given that the connection weights and unit thresholds in the model were carefully balanced, it is unclear how well it could be scaled to handle more complex languages. One interesting aspect of the model is that it makes use of the frequency with which productions in the grammar are used in sentences it has experienced. Thus, parsing can be biased toward more common interpretations and allows for a limited degree of learning. However, the model does not incorporate semantic information.

The model of Nakagawa and Mori (1988) also involves constructing the network on-the-fly, but rather than building a network for the entire sentence prior to parsing, it is generated sequentially, essentially implementing a left-corner parser. Sigma-pi units—those with multiplicative rather than additive inputs—are used to enforce ordering constraints on grammatical structures. Although the model can theoretically parse unbounded sentences, the copying mechanism used to construct the parse tree is not physiologically reasonable. The model also does not incorporate learning or semantic constraints.

The commonalities evident in these early connectionist parsing models leads to some generalizations about the limitations of the localist approach. With localist units, the representational capacity of the network is proportional to its size, leading to the inevitable problem that a fixed-size network can only handle inputs of bounded length or complexity. Like the limitations of symbolic models and unlike those of networks that use compositional distributed representations, this results in hard limits on the processing ability of localist networks. Such models do not exhibit a gradual degradation of performance with more difficult inputs and modeling of performance data generally requires ad hoc limitations. Learning is difficult in localist networks largely because of the problem of designing supervised training environments. This is compounded by the fact that large localist networks tend to require redundant structure and effective learning mechanisms ought to generalize what is learned across duplicated sub-networks. It is difficult or impossible to accomplish this in a reasonable manner. Finally, hand-wired networks do not allow easy incorporation of semantic information, which is necessary for parsing structurally ambiguous sentences, as aptly demonstrated in McClelland, St. John, and Taraban (1989). Aside from the ability to incorporate multiple sources of weak constraints, localist networks provide few advantages over symbolic models.

While the last decade has seen quite a few hybrid connectionist/symbolist parsing models, I will only mention two instances. The CDP model of Kwasny and Faisal (1990) is a modification of the PARSIFAL deterministic parser (Marcus, 1980). Several of the components of this rule-based parser were removed and replaced with a connectionist component. This network was trained to suggest actions to be taken by the symbolic components of the model based on the contents of the symbol stack and constituent inputs. The model was reportedly able to process ungrammatical and lexically ambiguous sentences in an appropriate way. However, it is not clear what effect the network component really had on the model. The primary reliance on a symbolic parsing mechanism is something most connectionist researchers would hope to avoid, and the authors did recognize the need for more fully connectionist parsers and discussed some of the hurdles involved.

Most grammar-based parsers suffer from an inability to parse sentences that fall outside of the given grammar. This can be a serious problem given the prevalence of pauses, false-starts, corrections, and word-substitutions in spoken language. Wermter and Weber (1994, 1997) and Weber and Wermter (1996) were interested in designing a model that was robust in the face of such problems. Their SCREEN model is a complex, highly modular, hybrid connectionist/symbolic system. While some of the modules are implemented in a symbolic manner, most are networks trained to perform a particular operation. Rather than producing full parse trees, the SCREEN model generates a *flat* syntactic and semantic parse. That is, the model labels the constituents by their syntactic class (e.g. noun or verb), their more abstract syntactic level (e.g. noun group or verb group), and some of their semantic properties including a few thematic roles (e.g. agent, action, or animate). The model was trained and tested on spontaneous spoken utterances and appears to work quite well. While the overall modular structure of the network is a symbolic design, the use of trainable, distributed networks allows for a certain level of generalization and fault tolerance. However, a serious limitation of the model, for many applications, is that the flat parse lacks much of the information necessary to construct a full parse tree. For example, the model does not appear to be capable of representing multiple interpretations of a prepositional phrase attachment ambiguity.

The Jain and Waibel (1990) model is essentially a localist, slot-based network but does incorporate learning and distributed representations at the word level. The model consists of a series of layers which essentially represent words, phrases, clauses, and inter-clausal relationships. These layers are trained independently with specified targets and therefore involve only limited learned, distributed representations. The model is interesting in its ability to process inputs over time, producing expectations of sentence structure, and dynamically revising hypotheses. However, it only has a fixed number of phrase and clause blocks and uses weight sharing to generalize learning across phrase blocks. This appears to cause a difficult trade-off between proper generalization and overgeneralization. It is not clear how well this model could make use of semantic information in resolving ambiguities.

Although several earlier connectionist models that were not purely parsers are described in Section 2.5, the XERIC model of Berg (1992) was one of the first distributed models that learns to parse. XERIC combines a simple-recurrent network (Elman, 1990) with a RAAM (Pollack, 1990) and is able to take words over time and produce a representation that can be decomposed into a parse tree whose structure is based on X-bar theory. This model has the advantage over localist methods that it can process unbounded sentences with only gradual degradation in performance. Although it was trained on a fairly simple grammar, the model is able to parse sentences with rather deep structure. While not included in the original work, it would be theoretically possible to introduce semantic information into this model by using a microfeatural encoding for words at the input. Despite its successes, XERIC might not be considered an adequate cognitive model as the hierarchical training procedure, like that for the RAAM, requires considerable memory and symbolic control. More crucial however, as with the Jain and Waibel (1990) model, is that the parsing information used to train the network is not available to the child.

Henderson (1994a, 1994b, 1996) described a localist, non-learning connectionist parser based on *temporal synchrony variable binding* (TSVB) and inspired by symbolic parsing theories. The main idea behind TSVB is that variable bindings, such as the bindings of constituents to thematic roles, can be represented by synchronous firing of constituent and role representations. The use of temporal synchrony, rather than something like binding units, reduces the need for duplicate structure and permits greater generalization. Henderson argued that the overall architecture is biologically well-motivated. The model, which is based on *structure unification grammar* (Henderson, 1990), does not itself construct an entire parse tree. Rather, it produces tree fragments with sufficient information that they could be combined into a complete tree. Because it is a deterministic parser, never backtracking on its commitments, and because it is unable to represent disjunctions of interpretations, it is likely that this model would have great difficulty with ambiguous sentences and suffer from an overly strong garden-path effect. The main drawback of the model is that it is primarily a connectionist implementation of a symbolic algorithm and lacks many of the advantages of connectionist networks, including the ability to learn and make use of multiple weak constraints.

Henderson and Lane (1998) and Lane and Henderson (1998) described an extension of the TSVB approach, known as a *simple synchrony network*, that can learn to parse sentences. The network took the part of speech tags for the sentence constituents as input and was trained to produce the parse tree fragment of any constituent seen so far when that constituent is queried. Although the network never produces a full parse tree, the tree fragments could be assembled into one. The network was able to learn to parse a corpus of written English to a reasonable degree of proficiency. However, this success is bounded by the limits of relying on parts of speech rather than actual words. This model might gain some advantage from using words rather than tags as input, but it would then encounter problems of lexical ambiguity. Nevertheless, the model is rather interesting, and could potentially have reasonable practical applications. I might note that, the name aside, TSVB seems to be identical in practice to the query mechanisms used in St. John and McClelland (1988, 1990, 1992) and in the work presented in this paper.

2.2 Comprehension

Although parsing models have sometimes been labeled comprehension models, I use the latter term here to refer to systems that aim to derive a meaning for an utterance that goes beyond its syntactic structure. There are, in fact, relatively few comprehension models in the literature. This may be due largely to the difficulty of representing and processing semantic information. Concept and phrase meanings involve subtle aspects that cannot easily be captured in a symbolic or localist system and do not interact in a cleanly combinatorial fashion. Furthermore, systems able to manipulate such information do not lend themselves to top-down design and are better constructed with learning methods. Therefore, comprehension has largely been the domain of distributed, connectionist models.

Hinton (1981) discussed one way in which semantic information and associations could be stored and recalled using distributed representations and pointed out some of the advantages this has over traditional localist semantic networks and over static distributed representations. One advantage is that associations formed between items may automatically generalize to semantically similar items. This work appears to have influenced, directly or indirectly, many subsequent connectionist models of semantics.

One such model is that of McClelland and Kawamoto (1986). While it does not derive fully structured representations of sentence meaning, this model produces thematic case role assignments, which are thought to be an important element of comprehension. Assigning case roles typically involves labeling the nouns in a sentence as agents, patients, instruments, and so forth. A key observation is that proper assignment of case roles does not simply depend on word order but also involves considerations of word meaning, inflectional morphology, and context. McClelland and Kawamoto hoped their model would be able to select the appropriate readings of ambiguous words, fill in missing arguments in incomplete sentences, and generalize its knowledge to handle novel words given their semantic properties.

The model uses stochastic units and a single layer of weights that is trained using the perceptron convergence rule. The inputs to the model consist of the semantic features of up to four main constituents of the sentence—

three nouns and a verb—which are then recoded in four larger sets of units that represent conjunctions of pairs of elements from the original arrays. The model is then trained to produce the semantic representations for the fillers of up to four thematic roles: agent, patient, instrument, and modifier. The model is able to satisfy many of the authors' goals, including resolving lexical and structural ambiguities, handling shades of meaning, and generalizing to novel words. However, as they acknowledge, this is just a first step and greatly simplifies the problem of sentence comprehension. The use of static input representations does not allow the network to process words over time and results in a hard limit on the complexity of sentences that can be handled. In particular, it would be unable to represent multi-clause sentences without considerable changes. The elimination of function words and the use of a fixed set of output slots limits the number of thematic roles that could be recognized by the model.

McClelland and Kawamoto suggested a number of ways in which these and other problems could be remedied and this was further fleshed out, though not implemented, in McClelland and St. John (1987) and McClelland (1989). Perhaps the best known model of sentence comprehension is the later work of St. John and McClelland (1988, 1990, 1992) and McClelland, St. John, and Taraban (1989). These papers described a model that shares many of the goals of the McClelland and Kawamoto (1986) work but extends the framework to produce a changing interpretation as each constituent is received and to allow the learning of distributed hidden representations of phrase and sentence meaning. The input end of the model is a simple-recurrent network (Elman, 1990) that learns to take a sequence of phrase components and compile a single message representation, known as the sentence gestalt, in the form of a trainable hidden layer. The phrase components are either a simple noun phrase, prepositional phrase, or verb. The output end of the model was trained to answer questions about the sentence in the form of a probe. When probed with a constituent, the network is to respond with the thematic role played by that constituent. When probed with a role, the network produces the constituent that fills that role. During training, the error that derives from these probes is backpropagated through the network to affect the formation of the sentence gestalt.

The St. John and McClelland model successfully exhibited its desired behaviors, including the ability to:

- make use of both syntactic and semantic clues to sentence meaning
- revise its interpretations online and produce expectations in the absence of complete information
- infer missing constituents, for example that eating soup is probably done with a spoon
- infer properties of vague constituents, such as "person", based on context
- handle both active and passive sentences
- use variable verb syntactic frames
- generalize its abilities to novel sentences

A major limitation of the model is that it is not able to processes multi-clause sentences, which are of considerable interest in the study of language. Other limitations include the representational inadequacy of a small number of fixed thematic roles and the lack of extra-sentential context. Nevertheless, the St. John and McClelland model remains a key inspiration for the work discussed in this paper.

One hindrance to the development of sentence comprehension models has been the difficulty of specifying adequate meaning representations of concepts and sentences. One solution adopted by Allen (1988), St. John (1992a) and Noelle and Cottrell (1995) is to avoid specifying meanings by focusing on language learning in the service of a task. By grounding language in this way, the model can be trained to respond to linguistic inputs by performing an appropriate action. Allen (1988) described a model which takes as input a coded microworld and sequential questions about that world. The simple-recurrent network was trained to answer questions with either yes/no or single constituent responses. In similar work, St. John (1992a) trained a simple-recurrent network to take a description of a scene and a sentence identifying a particular block in the scene, such as "the big blue block that is on the right of the left page," and output the block to which the sentence refers. The model is able to handle fairly complex inputs including relative clauses and prepositional phrases and can even handle human-produced sentences moderately well, but is otherwise severely limited in its scope.

Noelle and Cottrell (1995) were interested in the ability to perform a task immediately after receiving some instructions on how to perform it. This is described as "learning by being told". The framework of their model was inspired by the sentence gestalt network of St. John and McClelland (1990). The *plan* component of the network receives instructions over time and produces a plan that guides the performance of the *domain task* portion of the network. In this way, the sentence gestalt model might be viewed as one in which the input sentence instructs the model how to act appropriately in the domain of answering queries about that sentence. Although Noelle and Cottrell did not phrase the instructions to their model in natural language, that would be a simple extension. The suggestion that much of language is learned in the service of various tasks is a reasonable one. However, it seems unlikely that all of language is learned through direct, action-based feedback in this way.

Miikkulainen and Dyer (1989a) trained a backpropagation network on the same sentences used in the McClelland and Kawamoto (1986) study. The network learned to map from a static representation of the words in the sentence to a representation of the case role assignments. The principle difference between this and the earlier study is that McClelland and Kawamoto hand-designed feature-based distributed representations for words while the Miikkulainen and Dyer network learned the word representations, using the *FGREP-method*. In the FGREP-method, word representations are initially randomized. Error is propagated all the way back to the input units and the word representations are updated as if they were weights on links feeding the input group. The revised representations are then used as training targets on subsequent sentences. This method seems to be an effective one in practice for learning representations when they must appear at both the input and output of a network. However, it is not clear what prevents the representations from degenerating into, for example, all zeros, nor how it could be implemented without a symbolic controller. The task performed by the system is simpler due to the fact that words maintain the same representations in the input and output. There is no distinction between phonological and semantic representations and the meaning of a sentence is treated quite literally as the concatenation of its parts. The method was later extended to a simple-recurrent network which accepts the same sentences encoded sequentially (Miikkulainen & Dyer, 1990).

Miikkulainen and Dyer (1989b, 1990, 1991) further extended their model to the comprehension and production of script-based stories from a limited set of domains. The stories consisted of a series of simple sentences describing activities such as eating in a restaurant or shopping. The system involves four modular networks which all share the same word representations due to the FGREP mechanism. One network maps a sequence of words into a slot-filler representation of the case-roles of the sentence. The next module maps a sequence of sentence representations to a slot-filler story representation. Two other modules are trained on the inverse mappings. The networks are able to comprehend and reproduce the stories and can fill in missing details from partial stories. However, the true generalization abilities of the system are questionable given that the stories are drawn from a very restricted set of possibilities. While the use of modules improves the ability of the network to solve this task, the method relies on encoding sentences and stories with visible, slot-based representations. This does not extend easily to the more complex and subtle aspects of natural language.

Miikkulainen (1990b) applied the modular architecture to comprehending and producing sentences with relative clauses. The network is rather similar to that used to process stories. The sentences were composed of noun-verb or noun-verb-noun clauses, separated by commas. The first module maps from a sequence of words drawn from a single clause, or part of a clause if it contains an embedding, to a slot-based representation of the meaning. A second network maps from a sequence of clause frames to a static representation of all the frames in the sentence. Two other networks perform the inverse mappings. The system was trained on a set of 388 sentences with up to 3 clauses utilizing just 3 different verbs and 4 nouns and was able to reproduce the sentences quite well. The use of a slot-filler representation for sentence meaning places a hard constraint on the complexity of sentences that could be represented by this system. Another limitation is that it relies on markers to distinguish clause boundaries, thus preventing it from handling reduced-relative constructions, which lack relative pronouns. Nevertheless, aside from the current work, this appears to be the only connectionist comprehension model able to process complex sentences.

Two other connectionist comprehension models, Miikkulainen (1990a) and St. John (1992b), also address the problem of comprehending stories with multiple sentences. Both use sequences of propositions encoded in thematic role frames, rather than actual sentences, as input. For example, `(agent=person1, predicate=drove, patient=vehicle, destination=airport)`. The Miikkulainen model uses self-organizing feature maps to form an unsupervised classification of stories based on the type of event being described. The St. John model, known as the *story gestalt*, is quite similar in design to the earlier sentence gestalt models (St. John & McClelland, 1990). However, it was trained to answer queries about entire stories rather than individual sentences. The main issues addressed by the model are the representation of multiple propositions, resolution of pronouns, revision of on-going interpretations and inferences, and generalization, under the hypothesis that graded constraint satisfaction plays a primary role in these processes. The model was quite successful although its generalization abilities leave something to be desired.

2.3 Word prediction

Some of the most successful connectionist models of sentence processing are those that perform word prediction. Word prediction is a surprisingly useful ability. It can be the foundation for a *language model* which determines the likelihood that a particular utterance will occur in the language. This is a principal component of most speech recognition systems as it helps in resolving ambiguous inputs. The ability to predict accurately is sufficient to generate the language, and thus indicates knowledge of the grammar underlying the language. As a result, prediction networks are sometimes labeled *parsers.* However that term is reserved here for a model that produces an explicit representation of the syntactic structure of the sentence.

The best known connectionist prediction models are those of Elman (1990, 1991, 1993), who pioneered the use of simple-recurrent networks (SRNs), also called Elman networks. Elman (1990) applied an SRN to letter prediction in a concatenated sequence of words, demonstrating that the network could potentially learn to detect word boundaries by identifying locations of high entropy, where the prediction is difficult. Prediction might thus be a primary mechanism used by infants to learn word segmentation. Elman extended the model to word prediction in a language of simple sentences. Representations of words that developed at the network's hidden layer could be clustered to produce a reasonable classification of words syntactically and semantically. This indicates that much of basic knowledge required for parsing and comprehension could be extracted by a prediction mechanism from the child's input.

Elman (1991) further extended the model to process sentences that potentially involve multiple embedded clauses. The main goal of this work was to demonstrate that networks are capable of learning to represent complex, hierarchical structure. This is clearly a critical question if one is concerned with their ability to process natural language. As Elman put it, "The important result of the…work is to suggest that the sensitivity to context which is characteristic of many connectionist models, and which is built-in to the architecture of [SRNs], does not preclude the ability to capture generalizations which are at a high level of abstraction" (p. 220). However, a second major outcome of the work was the finding that the networks were only able to learn corpora of mostly complex sentences if they first began training on simple sentences before gradually advancing. This was developed further in Elman (1993), where it was shown that the networks could also learn well if their memory spans were initially hindered and then gradually allowed to improve.

These results appeared to have important implications for any language models based on recurrent networks. However, Rohde and Plaut (1997, in press) re-examined these findings and discovered that manipulating the training environment or memory span of the networks does not always facilitate learning and can, in fact, be harmful. These issues are revisited in greater depth in Section 3.1.

The other connectionist prediction models are all based more or less directly on (Elman, 1991). Weckerly and Elman (1992) focused specifically on the issue of the difficulty of right-branching versus center-embedded sentences. They found that, in accordance with behavioral data, the SRN showed a preference for sentences involving a double center-embedding versus a double right-branching structure. Furthermore, the network was able to make use of semantic constraints to facilitate word prediction in center-embedded sentences. However, these issues deserve further inquiry. The sentences used in training and testing the network, as well as those in most empirical studies, confound a number of factors with the center versus right distinction. These include subject relatives versus active and passive object relatives, frequency effects, semantic constraints, and the inconsistent use of reduced relatives. Finally, we might question the relevance results from a prediction network have in a comprehension task. A predictor has the advantage that it can forget information once it becomes irrelevant. This is the principal explanation for why such a model prefers right-branching structures. However, a comprehender must remember all important information at least until the end of the sentence. This would tend to weaken any preference for right-branching sentences. These issues are discussed further in Section 7.3.

Chater and Conkey (1992) compared Elman's SRN training procedure to a more complicated variant, backpropagation through time (Rumelhart et al., 1986), which extends the propagation of error derivatives back to the beginning of the sentence. Not surprisingly, they found that backpropagation through time, which is slower and considerably less "biologically plausible" produces better results. Bounded backpropagation through time is used in many of the experiments reported in this paper.

Christiansen (1994) tested the ability of SRNs to learn simple languages exhibiting three types of recursion: counting recursion[5], center-embeddings, and cross-dependencies, which exceed the power of a context-free grammar. However, these experiments resulted in rather poor results, with networks not even performing as well as statistical bigram models and sometimes worse than unigrams. It would be worth re-examining the methods used to train those networks. In a second experiment, Christiansen extended the language used by Elman (1991) to include prepositional phrases, left recursive genitives, conjunction of noun phrases, and sentential complements. One version of the grammar could produce center-embedded sentences and a second version cross-dependencies. In general the networks performed rather well on these languages and exhibited behaviors that largely reflect human comprehension performance on similar sentences. Christiansen and Chater (in press-b) extended these results and provided more detailed comparisons with human performance.

Finally, Tabor, Juliano, and Tanenhaus (1997) performed a number of experiments comparing human and network reading times on sentences involving structural

[5] Counting recursion involves sentences composed of a sequence of symbols of one type followed by an equivalent number of symbols of a second type without any further agreement constraints.

ambiguities. Although the network used in these studies was just a simple-recurrent prediction network, reading times were elicited using a novel "dynamical system" analysis. Essentially, the hidden representations that appear in the network at various stages in processing sentences are plotted in a high dimensional space. These points are treated as masses that exhibit a certain gravitational force. To determine the reading time of the network on a particular word, the network's hidden representation is plotted in the high dimensional space and then allowed to gravitate among the attractors until a stable state is reached. The settling time is taken as a proxy for reading time. Although this contrived test-mass settling was intended to be a proxy for a true dynamical system that actually settles into a stable state, no experiments were performed to demonstrate that this is a reasonable simplification.

2.4 Production

Sentence production has received far less attention than parsing or comprehension in the symbolist community. This may be largely due to the fact that simple production, if just viewed as the inverse of parsing, or deriving a sequence of words from a higher-order representation of sentence structure, can potentially be accomplished in a symbolic framework through the application of a few deterministic rules. The interesting challenge of parsing is resolving ambiguity, but that is not an issue in simple production. On the other hand, human-like production based on meaning is a very hard problem. Proper phrasing depends on nuances of meaning that are difficult to capture in a symbolic system (Ward, 1991). Thus, modelers have begun turning to connectionist approaches. However, most connectionist language production models have so far been restricted to the word level, dealing with lexical access and phoneme production rather than sentence-level phenomena (Dell, 1986; O'Seaghdha, Dell, Peterson, & Juliano, 1992; Harley, 1993; Dell, Juliano, & Govindjee, 1993). This section considers the most notable sentence production networks.

Kalita and Shastri (1987, 1994) focused on the problem of producing the words in a sentence given the thematic role fillers and indications of the desired voice and tense. Their model, which is a rather complex localist network, is able to produce simple SVO sentences in active or passive voice and in several tenses. In order to ensure that constituents are produced in the proper order, the model uses *sequencer* units to inhibit nodes once they have performed their duty. A special mechanism is included to allow the noun-phrase production component to be reused. Because of the complexity of hand-designing a localist network of this type and of representing thematic roles in multi-clause sentences, it is unlikely that this model could easily be extended to more complex sentences, particularly those with recursively nested structures. The model does not seem to exhibit any properties that transcend those of symbolic systems.

Gasser (1988) (see also Gasser & Dyer, 1988) described a significantly more ambitious localist production model that produces sentences using elaborate event schemas. The model, known as the Connectionist Lexical Memory, is based on interactive-activation principles. Bindings to syntactic roles are encoded with synchronized firing, as in temporal synchrony variable binding (Henderson, 1994a). Sequencing is accomplished using start and end nodes for each phrase structure, which are somewhat similar to the sequencer units in Kalita and Shastri's model. Gasser's model is designed to account for a wide range of phenomena, including priming effects, speech errors, robustness given incomplete input or linguistic knowledge, flexibility in sequencing, and transfer of knowledge to a second language. The model is also able to parse sentences using the same sequencing mechanism as for generation but may not be able to handle lexical ambiguities or garden paths. However, the model was only applied to simple clauses and noun phrases and does not produce recursive structures involving long-distance dependencies. Again, it is not clear whether such a localist model could be scaled up to handle more complex language.

The third major localist production model was by Ward (1991). This model was intended to be "more connectionist" than the previous attempts, relying on a truly interactive settling process and avoiding the need for binder units. Ward described previous models as essentially serial in their processing. His model, like Gasser's, was designed to handle both Japanese and English. One major limitation of the model, which may apply to the others as well, is that the network structures used to represent the intended meaning of the utterance are built on a sentence-by-sentence basis. Although the model is apparently able to produce a broader range of sentences than the previous attempts, it is still unable to handle agreement, anaphor, and relative clauses. Ward acknowledged that a primary drawback of the model is the difficulty of extending it in all but the most trivial ways and he recognized the need for a learning mechanism.

The inability to learn or to handle complex structure appears to be inherent in localist production models, which should not be surprising since these systems tend to be rather transparent implementations of classical finite state machines. However, while not truly context-free, language is pseudo-context-free or even pseudo-context-sensitive in that it allows a limited amount of recursion. For a simple, localist finite state machine to capture such recursion, it would require replicated structure, which would presumably be a serious hindrance to generalization. We therefore turn to models that make use of distributed representations with the hope of overcoming these problems.

Kukich (1987) was interested in the ability of a network to learn to produce stock market reports given the day's activity. He doubted the ability of a single network to learn the entire task and thus trained one network to associate units of meaning, sememes, to morphemes and another to re-order morphemes. Sememes were represented as a series of slot fillers encoding such information as the type of trading activity, and the direction, and duration of any change. The output of the first network was an unordered set of word stems and suffixes, which could be produced accurately 75% of the time. The morpheme ordering network did not actually produce morphemes sequentially but used a slot-based encoding of order. The results of these simulations left considerable room for improvement but were encouraging given the early state of connectionism.

I have already mentioned the comprehension and production models of Miikkulainen (1990b) and Miikkulainen and Dyer (1991). These were trained to produce either sequences of sentences based on a slot-filler representation of a story or multi-clause sentences based on a slot-filler representation of its clauses. So far this work has been restricted to fairly simple domains. The nature of the representations used appear to limit the ability of the system to be scaled up to more natural languages.

Finally, Dell, Chang, and Griffin (in press) were specifically interested in the phenomenon of structural priming, which leads speakers to preferentially produce sentences of a particular form, such as passive rather than active voice, if they have recently heard or produced sentences of similar form. Dell et al. hypothesized that the mechanism that results in structural priming is the same procedure used to learn production. Their model takes a representation of the sentence's propositional content and produces the words in the sentence sequentially. While it was intended to be an SRN, the recurrent portion of the model was not actually implemented, but was approximated by a symbolic procedure. Propositional content was encoded using a slot based representation consisting of localist representations of the agent, patient, recipient, location, and action. Therefore, the model was only able to produce simple sentences with a limited range of prepositional phrases.

Based on whether the agent or patient recieved greater emphasis, the model was trained to produce either active or passive sentences. It was also able to convey recipients using a prepositional phrase or a dative. The model learned to produce sentences with 94% of the words correct. Based on an average sentence length of 4.8 words, we might estimate that this translates to about 74% of sentences being produced correctly. The model was able to match human structural priming data quite well. The main limitations of this model are that it was only applied to simple sentences, didn't produce sentences as accurately as one might hope, and did not learn distributed context representations. The model presented in the current paper is rather similar to that used by Dell et al. but amends some of its limitations.

2.5 Other language processing models

There have been a few additional connectionist investigations of language that do not fit cleanly in one of the above categories.

Hanson and Kegl (1987) trained an auto-encoder network, known as PARSNIP, to compress sentences drawn from the Brown corpus (Francis & Kucera, 1979). Words were replaced by one of 467 syntactic categories, each encoded using 9 bits. Only sentences with fewer than 15 words were selected, eliminating relative clauses. The input and output representations for the network comprised 15 slots, holding the syntactic categories of all of the words in the sentence at once. PARSNIP was trained using backpropagation to map from the input to the identical output through a smaller layer of 45 units. When trained on 10,000 sentences, the network was able to reproduce about 85% of the word categories correctly. The network performed at about the same level on novel sentences, indicating robust generalization. PARSNIP was reportedly able to fill in missing sentence constituents and correct bad constituents, and did so in a way that often went against first-order statistics. It could handle single embeddings, despite their not having been trained, but not double embeddings or some sentences that violate English word order constraints. Although Hanson and Kegl acknowledge that auto-association is not a reasonable model for language acquisition, the importance of this work was its demonstration that distributed networks can learn to be sensitive to higher-order structure merely through exposure to surface forms and can generalize that knowledge in productive ways.

Allen (1987) performed a number of small studies of language using backpropagation networks. In one experiment, a network was presented sentences containing pronouns referring to nouns appearing earlier in the sentence and was trained to identify the location of the original noun. Although it is not clear how well the network could really perform the task, it was able to make use of semantic properties and gender in resolving some references. A second experiment involved training a network to translate from English to Spanish surface forms. The sentences dealt with a single topic, were limited to 11 words in length, and were presented to the network statically. A multi-layer feed-forward network was able to translate the sentences on a novel transfer set with an average of just 1.3 incorrect words. Although these early experiments were relatively simple, they are indicative of the ability of networks to learn complex language-related tasks.

Finally, Chalmers (1990) demonstrated that connectionist networks, while able to construct compositional representations through mechanisms such as the RAAM (discussed in Section 6.1), can also operate directly on those representations in a holistic fashion without first decomposing them. Chalmers first trained a RAAM to encode simple active and passive sentences and then trained a second network to transform the structural encodings of an active sentence to that for the corresponding passive sentence. The transformation network was found to generalize quite well to novel sentences. This simple experiment demonstrated that networks can perform structure-sensitive operations in a manner that is not simply an implementation of symbolic processes.

In summary, other than prediction networks, no connectionist sentence processing models have exhibited all of the main properties necessary to provide a plausible account of natural language acquisition. These include the ability to learn a grammar, to process a sentence sequentially, to represent complex, multi-clause sentences, and to be naturally extendable to languages outside of the domain originally addressed by the designer.

3 Word Prediction Revisited

The ability to predict utterances in a language is quite powerful. Accurate word prediction is quite powerful, entailing knowledge sufficient to produce a language or to decide the grammaticality of any sentence. Prediction is the role of the *language model*, which has been found to be essential in many forms of automated natural language processing, such as speech recognition (Huang, Ariki, & Jack, 1990), and is believed to play a role in human comprehension (Marslen-Wilson & Tyler, 1980). The benefit of prediction is largely that it enables the system to disambiguate noisy input on the basis of the likelihood that a particular utterance was intended and because it allows the system to prepare itself to better handle expected inputs. In learning complex, goal-directed behavior, prediction can provide the feedback necessary to learn an internal *forward model* of how actions relate to outcomes (Jordan & Rumelhart, 1992). Such a model can be used to convert "distal" discrepancies between observable outcomes and goals into the "proximal" error signals necessary for learning, thereby obviating the need for externally provided error signals. Two important additional features of prediction are that no external training signal is needed other than the input itself and feedback is available immediately; the learner need not perform a re-analysis of previously observed positive evidence (cf. Marcus, 1993). I raise the issue of prediction here because, as we will see in Sections 7 and 8, it may play a fundamental role in learning language production.

It is important to clarify that the type of predictions I hypothesize the language system might be engaged in are not necessarily consciously accessible, nor must predictions be over a small set of alternatives. Nor, for that matter, is prediction restricted to a probability distribution over localist lexical units—it is likely that linguistic predictions occur on many levels of representation, across phonemic features, across semantic and syntactic features of words, and across semantic and syntactic features of entire phrases.

We might view prediction as involving the operation of standard processing mechanisms which embody the general computational principle, in interpreting linguistic utterances, of going as far beyond the literal input as possible in order to facilitate subsequent processing (see McClelland et al., 1989). One version of this approach would be to propose that the system maintains and updates in real time a probability distribution over words reflecting the likelihood that each word is the one being heard. Such a distribution is exactly what would emerge from attempting to predict the current word as early as possible. More generally, accurate prediction need not and should not be based on the preceding surface forms alone, as in a k-limited Markov source. In order to make accurate predictions and to generalize to novel combinations of surface forms, the system must learn to extract and represent the underlying higher-order structure of its environment.

3.1 On the importance of starting small

In order to demonstrate the ability of neural networks to learn to encode hierarchical constituent structure, Elman (1991, 1993) investigated the ability of a neural network to learn a language prediction task. He trained a simple recurrent network (Elman, 1990, sometimes termed an "Elman" network) to predict the next word in sentences generated by an artificial grammar exhibiting number agreement, variable verb argument structure, and embedded clauses. Elman found that the network was unable to learn the prediction task—and, hence, the underlying grammar—when presented from the outset with sentences generated by the full grammar. The network was, however, able to learn if it was trained first on just simple sentences (i.e., those without embeddings) followed by an increasing proportion of complex sentences. The explanation of these findings was that focusing on simple structures initially allowed the network to learn the important noun-verb relationships before applying that knowledge to more complex constructions.

Additionally, Elman (1993) found that networks were also able to learn when faced with the complete language from the start but with their memory span initially limited and allowed to improve gradually. The explanation in this case was that the memory-limited network was only able to learn the simple relations first but that this helped it to subsequently learn more complex structures. The fact that learning was successful only under conditions of restricted input or restricted memory is what Elman referred to as “the importance of starting small.”

Elman’s finding that simplifying a network’s training environment or limiting its computational resources was necessary for effective language learning was interesting largely because it accords well with Newport’s “less is more” proposal (Newport, 1990; Goldowsky & Newport, 1993)—that the ability to learn a language declines over time as a result of an *increase* in cognitive abilities. This hypothesis is based on evidence that early and late learners seem to show qualitative differences in the types of errors they make. It has been suggested that limited abilities may force children to focus on smaller linguistic units which form the fundamental components of language, rather than memorizing larger units which are less amenable to recombination. In terms of Elman’s network, it is possible that staged input or limited memory similarly caused the network to focus early on simple and important features, such as the relationship between nouns and verbs. By “starting small,” it is believed, the network had a better foundation for learning the more difficult grammatical relationships which span potentially long and uninformative embeddings.

However, in a reexamination of these findings (Rohde & Plaut, 1997, in press), we explored the hypothesis that the importance of starting small might be less critical with a somewhat more natural language than the one used by Elman. To this end, we introduced a training language that was similar to Elman’s but added semantic constraints limiting the nouns that could act as subjects and objects of each verb. The extent to which these semantic constraints were applied between constituents of a clause and those of its sub-clause could be parametrically controlled. Somewhat surprisingly, we found that there was, in fact, a significant advantage to starting with the full language. As illustrated in Figure 1, this advantage increased as between-clause semantic constraints became stronger.

We then attempted a more direct replication of Elman’s work in order to better understand the discrepancy in our results. Using our training methods on Elman’s grammar again resulted in a significant disadvantage for starting small. However, when we adopted similar parameters, we found neither the simple nor complex training regimes led to successful learning, although the complex regimen was worse. Experimenting with the role of various parameters we discovered that the range of initial random connection weights has the greatest effect. In particular, if the range is close to the value of ±0.001 chosen by Elman, progress can be severely delayed or even impossible. This is especially true of networks in the complex regimen. However, if the range is closer to the ±1.0 we adopted, learning is initially quite rapid in both regimes but ultimately more successful for networks that saw a stable, complex input.

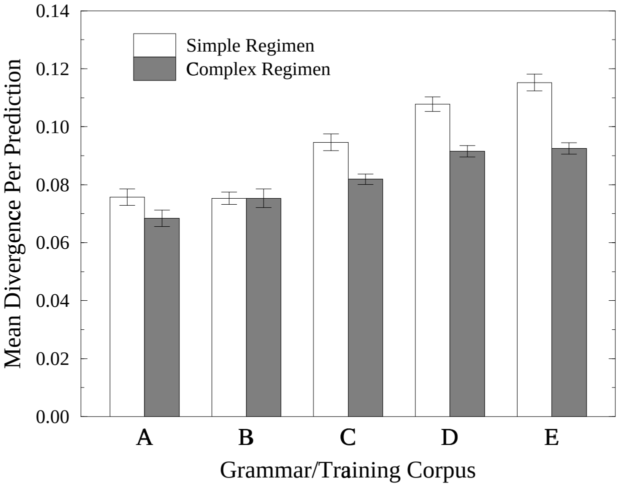

Figure 1: Mean divergence per word prediction over the 75% complex testing corpora for the simple and complex training regimes. Lower values indicate better performance. Grammar classes A through E vary in the percentage of sentences that are forced to obey semantic constraints, from 0% to 100%. Means and standard errors were computed over the best 16 of 20 trials in each condition.

Our studies of memory limitations found that they seem to have no significant effect on learning. Initially, the network has not yet learned to use its memory and periodically interfering with it has no effect. As the network develops representations that facilitate reliance on memory, the limitations are relaxed and thus continue to have little effect. Starting small in prediction tasks does not seem to be a fundamental necessity for recurrent networks because networks tend to naturally start by picking out simple correlations in the input. The danger of starting with simplified inputs is that, early on, the network may commit all of its resources to solving the apparently simple task and will subsequently be unable to restructure its representations once the demands of the task change. It appears that Elman’s difficulty with the complex regimen may have been largely due to an unfortunate choice of initial random weights.

Although we believe starting small to be of dubious import in prediction, it may have significant benefits in learning comprehension. This is especially true because the targets for comprehension are not immediately provided as in prediction. A child presumably learns comprehension by inducing the intended meaning of its parent's utterance from its world knowledge and the current situation. It is unlikely that the child will be capable of inducing or even representing complex messages. Therefore, complex sentences may largely serve as noise until the child is capable of representing the thoughts expressed by them and it may be advantageous to simplify our utterances to children, as we appear to do (Newport, Gleitman, & Gleitman, 1977). Nevertheless, we fully agree with Elman in the more important result of these experiments: that recurrent networks appear capable of learning to represent the hierarchical constituent structures necessary for processing natural language without relying on extensive innate information.

4 The Wanna Problem

Prediction networks have been an important tool in providing new explanations for our linguistic abilities and challenging claims against the learnability of language. This sections describes a recent experiment along these lines. One aspect of English that appears rather difficult for a learner to acquire is knowledge of the often rare circumstances in which certain lexical contractions are prohibited. This is especially insidious in the case of the contraction from "want to" to "wanna", as illustrated in the following sentences:

(1) a. You want to play with Stan?
b. Who do you want to play with t?
c. Who do you wanna play with t?

(2) a. You want Stan to play with you?
b. Who do you want t to play with you?
c. *Who do you wanna play with you?

Under most circumstances, English permits the wanna contraction, as in the *object extraction* question (1c) in which, according to transformational grammar, "who" has been moved from the object position at the end of the sentence to the front, leaving behind a "Wh-trace", identified with a t. On the other hand, sentence (2b) involves *subject extraction* in which the Wh-trace falls between "want" and "to". Most English speakers judge contraction in this case to sound unnatural. One explanation is that the rules of English prohibit contraction across a Wh-trace. While it is debatable whether the existence of a Wh-trace is truly the reason that contraction is awkward in subject-extraction questions, a more important issue is whether this aspect of English could be learned from the input available to the child or whether it reflects an innate rule, as has been maintained by some theorists (Chomsky, 1976; Thornton, 1999; Crain, 1991).

The problem here arises from the fact that the traditional view of language discounts the role of the statistical properties of the input. Accordingly, the language input is considered to be a sequence of valid sentences drawn with no guarantee that any particular sentence, grammatical structure, or rule will be observed in any fixed period of observation. The only guarantee is that every legal sentence will eventually be heard. Therefore, the learner, following any finite exposure, is never justified in concluding that a particular structure is not part of the language simply because it has not been observed. As far as the learner knows, that structure may be observed at any moment. Likewise, in such an environment it is not possible to recover from overgeneralization. For example, if the learner adopted the over-general rule that "want to" may always be contracted to "wanna", it will never discover that it is wrong because the lack of sentences like (2c) does not constitute evidence that such sentences will never appear. Thus, Gold (1967) has proved theoretically that most interesting language learning tasks are impossible when only very weak constraints are placed on the input.

As Crain (1991) argues, assuming children are only exposed to examples of valid sentences without negative feedback:

> It is difficult to see how knowledge about the ungrammaticality of sentences like [2c] could have been acquired through exposure to environmental input at any age. It is accordingly important to ask when children know that a trace blocks contraction. The logic of the situation would suggest they must know it innately… If corrective feedback is not available to children who err in this way, …children who make the false generalization would not be informed of their mistake, and would not attain the adult grammar… Here, then, is a partial syntactic generalization of grammar that clearly calls for assistance from innate linguistic principles. (p. 603)

Rather than resorting to innate knowledge, an alternative perspective is offered by accepting a more restrictive model of the language environment. We might replace the weak guarantee that each legal sentence will eventually be observed with a stochastic guarantee that each legal sentence will be observed during any fixed period of observation with some non-zero probability. As a result, the probability that a common, and thus important, sentence or sentence structure goes unobserved after a lengthening period of time grows ever smaller. If a learner does not observe a structure such as (2c) after considerable observation, the learner can assume, with small chance of being incorrect, that the structure is not part of the language.

I hypothesized that, given such a stochastic environment, the proper use of "wanna" could be learned by a system sensitive to the statistical properties of its input when exposed to a language approximating that experienced by a child. In particular, I trained a simple-recurrent network on a prediction task and then elicited grammaticality judgements from the network. The result is that the network became quite sensitive to the proper use of “wanna” despite any negative examples, despite the lack of any semantic information, and despite the very rare occurrence of either object- or subject-extraction questions in the language.

4.1 The *wanting* language

It was my intention that the language used in these experiments reflect, as accurately as possible, the distribution of inputs children are likely to experience while they themselves are learning the proper use of contractions. Because it was not feasible to model the entire language, the input was restricted to those sentences dealing with “wanting”, which is a fairly common topic of conversation with children. To obtain an approximate model of this portion of English, I analyzed approximately 1,000 child-directed sentences from the CHILDES database (MacWhinney, 1999) and classified them into 54 sentence frames with associated frequencies of occurrence, which are displayed in Table 4.1.

In the artificial language, 15.7% of the utterances are statements and the rest questions. This turned out slightly less than the 17.2% in the CHILDES data, the frequency of questions being particularly high for sentences involving wanting. 24.0% of the artificial sentences use the reduced form, wanna, which is quite close to the 24.4% in the CHILDES data. It is interesting to note that, in the data, the contracted form was used in 40% of the cases in which it is appropriate. However, we cannot rely too much on these exact proportions because it is likely that transcribers varied in their sensitivity to whether a contraction was used. Somewhat surprisingly, object- and subject- extraction questions were actually quite rare in the data. Of the 22,418 sentences involving wanting, only 37 were “who” questions. Of these, 7 were uncontracted object-extractions of the form (1b), 4 were contracted object-extractions of the form (1c), 9 were uncontracted subject-extractions of the form (2b), and 2 were of the supposedly illegal form (2c). However, we shouldn’t conclude too much from these findings, given the sparsity of the data and the possibly non-uniform transcriptions.

Although subject-extraction questions were quite rare overall, this may partially be an effect of the way in which the CHILDES data was collected. Many of the observations involved a parent interacting with their child in a closed environment. In a situation with no other participants, it may be common to ask “What do you want to play with?” but not “Who do you want to play with?” or “Who do you want to play with you?”. In the artificial language, 0.12% (just over a tenth of one percent) of the sentences were of the form (1b), 0.08% were of the form (1c), and 0.2% were of the form (2b). There were no sentences of the form (2c).

Sentences were completed from the frames using a fixed set of noun and verb phrases. 33 noun phrases were able to serve as direct objects and 15 as subjects, not including the very common “you” and “I” that were included in the frames. The nouns were selected from the most frequent words in the CHILDES data. Some examples of noun phrases are: “it”, “Mommy”, “the baby”, “your toys”, and “something”. 26 of the most common verbs were selected, forming approximately 80 verb phrase structures, including: “read READ_OP”, “feed FEED_OP to FEED_RE”, “feed FEED_RE FEED_OP”, “sit with SIT_OP”. Each verb permitted only certain noun phrases in each of its thematic role slots, as were semantically appropriate. All told, the language used 90 lexical items.

Rather than giving the prediction network localist representations of the lexical items, as in Elman (1991) and Rohde and Plaut (in press), words were presented one syllable at a time using a phonological encoding. This is because contracted forms are not merely arbitrary replacements for the original words, but are phonologically quite similar. I hypothesized that this similarity could be quite important as it makes the generalization to the contracted form easier and the non-generalization required to rule out sentences of the form (2c) correspondingly harder. Syllables were encoded using one unit for each phoneme in the onset, vowel, and coda. 20 units were required for the onsets, 15 for the vowels, and 16 for the codas.

At each step, the network was trained to predict both the next syllable and the word that would contain the next syllable. It was also trained to predict the end of the sentence and indicate whether it had been a question or a statement. However, at the start of the sentence the network was told which it would be. This was meant to reflect the fact that, in English, the speaker’s tone often distinguishes statements from questions even from the very start. The mean sentence length was approximately 6 syllables (5.4 words), or 7 including the initial punctuation.

4.2 Methods and results

The network used in these experiments is illustrated in Figure 2. It is a simple-recurrent or Elman network (Elman, 1990) which uses logistic activation functions on all of the units but a soft-max constraint (Luce, 1986) on the lexical-level output units which normalizes their total activation to 1.0. A cross-entropy error measure was used on the syllable output group and divergence on the lexical output group. Initial weights were randomly generated in

Table 1: Sentence frames used in the "wanting language" with associated frequencies.

Frame	Frequency
? you want OP ?	0.1230
? DO1 you want to VP ?	0.1230
? you want to VP ?	0.1230
? DO1 you want OP ?	0.0840
? you wanna VP ?	0.0760
? wanna VP ?	0.0500
? want to VP ?	0.0500
? DO1 you wanna VP ?	0.0420
? what DO1 you want ?	0.0380
? you want OP2 to VP ?	0.0380
. you want to VP .	0.0280
? DO1 you want OP2 to VP ?	0.0260
? want OP ?	0.0260
. you want OP .	0.0220
. I want to VP .	0.0200
. I wanna VP .	0.0180
. NP WANT to VP .	0.0180
? want OP2 to VP ?	0.0140
? what DO1 you want OP2 to VP_WHAT ?	0.0140
? what DO1 you wanna do ?	0.0140
. I want OP .	0.0120
. you wanna VP .	0.0120
? which O DO1 you want to VP_WHAT ?	0.0100
. you want OP2 to VP .	0.0080
. I want OP2 to VP .	0.0070
? which O DO1 you want ?	0.0070
. we wanna VP .	0.0070
? what DO1 you wanna VP_WHAT ?	0.0070
? why DO1 you want OP ?	0.0060
? why DO NP want to VP ?	0.0060
? what DO1 you want to VP ?	0.0050
? where DO1 you want to VP ?	0.0050
. NP WANT you to VP .	0.0050
? NP WANT to VP ?	0.0050
? which O DO1 you want OP2 to VP_WHAT ?	0.0040
? how DO1 you want OP ?	0.0040
? why DO1 you wanna VP ?	0.0040
? who DO1 you wanna be ?	0.0040
? why DO1 you want OP2 to VP ?	0.0040
? I wanna VP ?	0.0030
? I want to VP ?	0.0030
? where DO1 you want OP ?	0.0030
? what DO NP want ?	0.0030
? what DO1 you want to do ?	0.0030
? DO NP want OP ?	0.0030
? DO NP want to VP ?	0.0030
? who wants to VP ?	0.0030
? what DO1 you want to VP_WHAT ?	0.0020
? which O DO1 you want to do ?	0.0020
? which O DO1 you wanna VP_WHAT ?	0.0020
? how DO1 you want to VP ?	0.0020
? who DO1 you want to VP_WHO ?	**0.0012**
? who DO1 you wanna VP_WHO ?	**0.0008**
? who DO1 you want to VP ?	**0.0020**

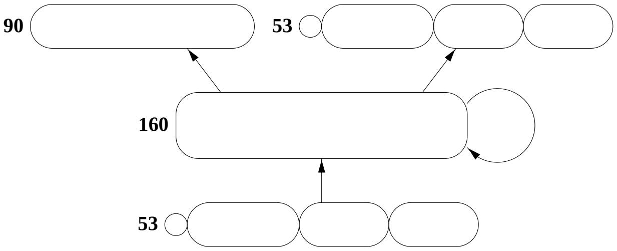

Figure 2: The simple-recurrent prediction network used in the "wanna" experiment.

the range ± 0.3. The network was trained on 1 million sentences using the LENS neural network simulator (Rohde, 1999a). Sentences were not drawn from a fixed corpus but were generated on-the-fly from the grammar using a program called the Simple Language Generator (Rohde, 1999b). Weight updates were performed every ten sentences. Bounded momentum descent was used with a momentum of 0.9 and a learning rate initialized to 0.2, annealed gradually to 0.04, and then reduced to 0.0005 for the last 50,000 sentences.[6]

When tested on 5,000 sentences produced in the same way as the training corpus, the average error per prediction (compared against the theoretically correct prediction distribution) was 0.0056 for the lexical prediction and 0.0217 for the syllable prediction, compared with 3.93 and 42.6 for a random network. To test whether the network had learned the proper usage of wanna, four sets were constructed each containing 500 sentences from one of the four extraction-question sentence types: (1b), (1c), (2b), and (2c). In generating these, questions that were ambiguously subject- or object- extractions, such as "Who do you want to help?" were eliminated. A prediction network that has learned the appropriate use of wanna should perform poorly on the illegal sentences of type (2c). Indeed, this is reflected in the overall lexical prediction error which, when compared against the actual next word, was 1.45 for sentences (1b), 1.44 for (1c), 1.34 for (2b), but 1.97 for (2c). However, a significant portion of this error is due to the difficult prediction of "who" at the beginning of all of these sentences. When this is factored out, the error scores are (1b):0.922, (1c):0.917, (2b):0.870, (2c):1.503. Thus, the illegal subject-extraction wanna sentences produce significantly greater prediction error overall.

However, average prediction error is perhaps not the best measure of whether the sentences have violated the network's language model. Therefore, we formulated a measure of grammaticality based on the network's lex-

[6] Bounded momentum is a method I developed that bounds the length of the weight adjustment step to be no greater than the learning rate, prior to the addition of the momentum term. In most cases this makes learning more stable and reduces the need to adjust the learning rate or momentum.

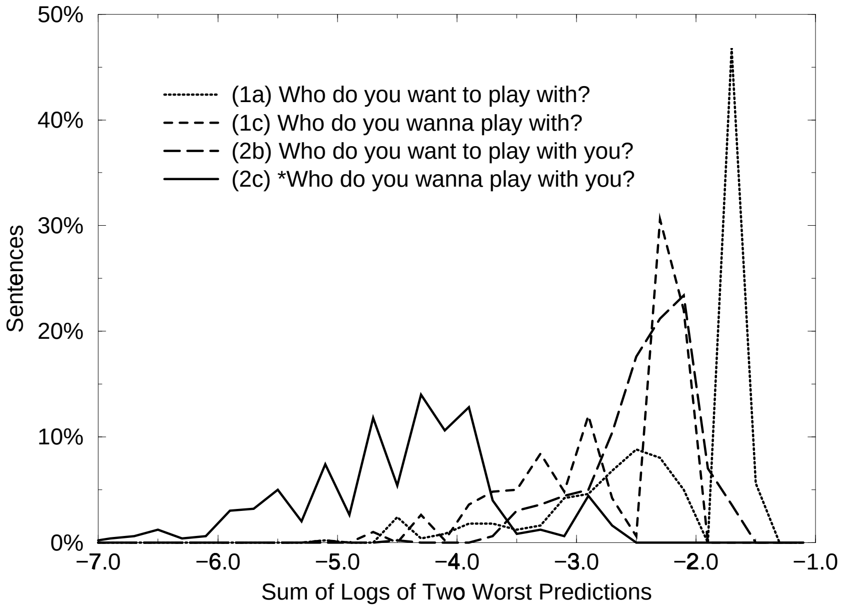

Figure 3: Grammaticality ratings of the wanna problem test sentences. Values to the left are less grammatical.

ical predictions. Borrowing from Rohde and Plaut (in press), to assess the "goodness" of the sentence for the purpose of judging its grammaticality we selected the two words that were most "surprising" to the network (those to which the network assigned the least likelihood, other than the initial "who") and took the log of the product of the two likelihoods. Figure 3 shows a histogram of the grammaticality ratings of the 500 sentences in each of the four extraction-question types. Note that the distribution for the sentences of type (2c) is shifted to the left, indicating that these sentences are less grammatical according to the measure used. Of particular interest is the fact that the distributions for (2b) and (2c), which differ only in the use of "wanna", are quite well separated. If a value of -3.72 were selected as the decision point between grammatical and ungrammatical sentences, the network would make just 10.6% errors on (2c), 10.4% on (1c), 6.6% on (1b), and 0.2% on (2b).

4.3 Conclusion

The network in this experiment was trained to perform word and syllable prediction on a language that models the experience children have in English with sentences involving wanting. Despite any negative examples, the lack of any semantic information, and the very rare occurrence of extraction questions in the language, the network became quite sensitive to the prohibition against using the contracted form of "want to" in subject-extraction questions. I do not intend to suggest that the learner of English makes no use of semantic clues in solving the wanna problem. In fact, I believe that transfer of knowledge from familiarity with statements such as (2a) to questions such as (2b) may play a major role. The conclusion is merely that there does not appear to be any principled reason to believe that the wanna problem could not be solved by a fairly general learner based on nothing more than sensitivity to the observable properties of the input.

These results do not shed light on whether the invalid uses of "wanna", as far as the network or human learner is concerned, have anything to do with the possible existence of a Wh-trace. However, other contracted forms in English have proscribed usages that do not involve Wh-traces. For example, although "should have" can usually be contracted to "shoulda", one could say "I should have some coffee," but not, "*I shoulda some coffee". Although we have limited ourselves to the "wanna" contraction in these experiments, the model could easily be applied to other cases. It may be that a systematic relationship between these various forms and the underlying semantics makes it even easier to learn when contractions are permitted and when they are proscribed.

5 An Integrated Framework for Sentence Processing

A primary goal of many researchers involved in the connectionist approach to language is to investigate whether language could be learned by a fairly general mechanism not reliant on detailed, innate knowledge. The principle language tasks that must be addressed are comprehension and production. However, as discussed in Section 2, there has so far been relatively little work on these fundamental problems. Most connectionist learning models to date have focused on single-word tasks, prediction, parsing, and limited forms of comprehension such as thematic role assignment. Comprehension and production models, with possibly one or two exceptions, have not been designed to handle complex sentences with embedded structure.

The development of a production model is particularly important because much of the behavioral data that we might hope to explain with a connectionist system comes from production. While it is often difficult to assess the comprehension abilities of children or second language learners, their errors in production are much easier to document. However, language production is undoubtedly intimately tied to comprehension and is dependent on knowledge gained from observing the speech of others, so we may be misguided in training a production system in isolation. Therefore, the primary aim of this project is to develop a connectionist model of language that integrates the tasks of comprehension and production in a single, coherent framework.

The first goal is largely a technical one: to produce a model that can be trained in reasonable time and learns to perform comprehension and production with some degree of proficiency. Although it will be trained on simplified subsets of English, the model must be capable of handling sentences with embedded clausal structure in order for the model to be extensible to the full language. To the extent that it is feasible, the language should include as many potentially difficult aspects of English as possible. Because we believe that so much of the linguistic behavior of humans can be explained by sensitivity to the statistical properties of our languages, we should not expect a model to exhibit those same behaviors unless it is trained on a language that is sufficiently similar to English. The ability of a network to master a reasonable approximation to English may help to dispel the conviction that language could not be learned without reliance on detailed innate constraints, beyond the general forms of constraint implicit in pre-trained networks.

A secondary goal of the project, assuming the model is able to achieve reasonable performance, will be to evaluate its ability to replicate the progression of errors demonstrated by first- and second-language learners and the problems adult speakers have with various aspects of language. Two such areas of investigation will be the relative difficulty of center-embedded versus right-branching structures and the ability of the model to make use of semantic constraints in resolving ambiguous or garden-path sentences. Models introduced to a second language ought to show increased difficulty attaining native-like performance with later exposure and should have more or less difficulty with various structures as reported in, for example, Johnson and Newport (1989).

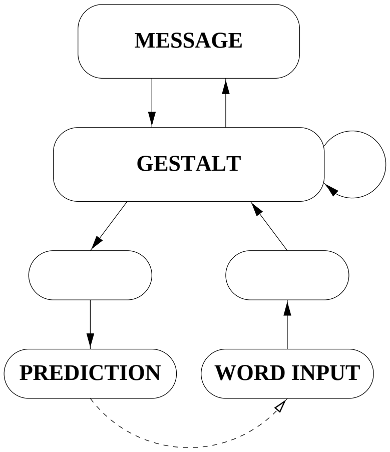

Figure 4: A simplified view of the comprehension/prediction/production model.

Assuming the model can be sufficiently validated using behavioral data, an eventual goal will be to map the network onto the brain, explaining the possible role played by each of the brain areas known to be involved in language. Ideally, lesions to the network should produce similar deficits to those produced by damage to the corresponding brain region. Traditional characterizations of these regions tend to draw clean distinctions between areas involved in comprehension and those for production. Although damage to different brain regions has varying effects on receptive and productive tasks, known aphasias have proven difficult to fully explain with traditional models. Virtually all aphasias involve impairments of both comprehension and production, lending credence to the view that the mechanisms underlying these abilities are very much integrated. It is possible that this model could yield a better understanding of the neurobiology of language.

5.1 An outline of the model

The basic design of the model is illustrated in Figure 4. The simple-recurrent network processes one word per time-step. During comprehension, words are presented at the *word input* layer. The network is trained to map the word sequence to the message at the *message* layer. In doing so, the *gestalt* layer integrates each word into the sentence context represented by the previous activations at the message and gestalt layers. Thus, the gestalt representation will presumably integrate syntactic and semantic information.

Developing appropriate representations for complex messages is actually quite a hard problem. Although one could hand-design a representation, this is very difficult without placing strict bounds on the level of syntactic complexity of the language. It is also difficult to design distributed encodings which are more efficient than localist representations and better support generalization. Therefore, I have studied ways of training the network to develop its own representations of sentence meaning using either a RAAM (Pollack, 1990) or query framework (St. John & McClelland, 1988, 1992), which are discussed in Section 6.

This model of comprehension presupposes that the system is able to derive the messages of observed sentences from some additional source of information. This may be a profoundly simplifying assumption but perhaps one that is not entirely unreasonable. There are a number of sources from which messages could be derived. If statements pertain to a visually observable scene, such as, "Look at the dog chase the cat," the child could derive the meanings of those sentences from context. The child may not be able to predict the entire utterance in advance, but given the context and a vague notion of the words, the child could induce one or two plausible messages. If the child fails to understand a command, such as, "Time for bed," or "Eat your peas," the meaning may be made readily apparent. Given that a parents' goal is not usually to teach their child language but to communicate with the child (Newport et al., 1977), it will be quite common for the parent to use gesture, repetition, and rephrasing to help the child obtain the correct message. Pinker (1990) has argued that it may be a reasonable first idealization to assume that messages are available to the learner. At issue, perhaps, is how accurate induced meanings can be and when they become available in the course of hearing the utterance.

The current model makes the assumption that messages are available to provide error feedback throughout comprehension. It is not a big step to suppose that these messages are also able to help guide comprehension. Various versions of the model may assume that either whole or partial messages are available as input. These inputs may be noisy or only available rarely and it is likely that manipulating this parameter will have a major influence on the behavior of the network as it develops.

While performing comprehension, the network will also be trained to predict the next word in the sentence. Although this may cause interference by using resources at the gestalt level that could otherwise serve comprehension, it is possible that prediction may actually aid comprehension. Prediction requires the development of abstract structural representations and the network may be able to map these representations more or less directly onto messages. If message is not available as input, prediction will be based just on the observed beginning of the sentence and is likely to be inaccurate to the extent that the sentence could be completed in multiple ways. However, if the message is available in advance, either fully, partially, or noisily, prediction could theoretically be quite accurate.

The main motivation for the inclusion of prediction in the model is that a network that has learned to do prediction in the context of a known message is well on its way to production. In essence, the language producer is just saying what it expects another speaker of the language to say to convey the same message. This would explain how much of the knowledge required for production can actually be gained not by trial and error, during which children receive little feedback, but by observing the speech of others. However, it may be that production, beyond simply the motor aspects, is also refined through self-monitoring.

The proposed model will be given the message it should convey at the start of production. Two ways of presenting the message are discussed later. The network will then try to predict the first word of the sentence. This will produce a probability distribution at the prediction layer, although hopefully one that is heavily weighted on a single word. A word will be selected from that distribution and effectively produced. That word will then be fed to the network as input and the network will be asked to predict the next word. As the network listens to itself, it will be deriving a representation at the message level just as in comprehension. The difference between the intended and derived messages will produce an error signal that can be backpropagated through the network and through the word-selection mechanism operating on the prediction layer on the previous time-step. Thus, the network can learn to refine its productions, or even learn production completely, by self-monitoring. An important parameter of the model will be the extent to which it learns production through actual practice or implicitly during comprehension.

The current model, by design, has several potentially important limitations. First of all, it does not go below the lexical level in either the input or the output. Thus, it does not provide an account of morphological effects and ignores the phonological and motor levels in both comprehension and production. By this we do not intend to suggest that these are modularly separable from the sentence-level system, we have merely truncated the model at a certain point for simplicity. In theory, the addition of more refined input and output representations should not be a problem and adding them may be necessary to model some interesting data. Another simplification is the mechanism for choosing words during production. Although this is currently a symbolic procedure that selects a word based on the prediction distribution, this is just a surrogate for an interactive procedure in which the network would settle on the appropriate word or syllable representation and then drive the articulators to produce the word.

More serious limitations are that the model does not go beyond processing of sentences in isolation and does not incorporate associative or propositional knowledge of the world except to the extent that knowledge can be derived from the training sentences. A complete system should contain associative knowledge, such as "pianos are heavy", that would be available for explicit reference or access with non-linguistic mechanisms, but would still enable the model to correctly interpret such sentences as, "You run like you've got a piano on your back." This body of associative knowledge might be termed the "world model". The current proposal also eliminates knowledge of the current linguistic and physical situation, which is crucial for properly interpreting many aspects of language, most notably pronoun reference. This "situational model" ought to aid comprehension and to be updated with information gained from comprehension.

Nevertheless, the currently proposed model already pushes the limits of our computational hardware and involves sufficient complexity that it should support a number of interesting experiments as well as extend our understanding of the language processing abilities of neural networks.

5.2 The language

The first language used to develop the model was not carefully controlled for the purpose of performing any particular experiments. The goal in designing this language was to produce a grammar complex enough that the ability to learn the language would be indicative of the ability to learn closer approximations to natural language. Therefore, several potentially difficult features were included, such as variable verb tenses and argument structures, passive constructions, relative clauses and reduced relatives, noun- and verb-modifying prepositional phrases, articles and adjectives, singular, plural, and mass nouns with number agreement, dative shift, and semantic plausibility constraints.

The base grammar for this language is displayed in Table 2. Where a production is followed by a number in parentheses, the number is the probability that production is used. The probabilities for unlabeled productions are dependent on semantic constraints. All sentences in this language are statements and begin with a start symbol, >, and end with an end-of-sentence marker. The basic sentence structure, S2, permits an intransitive, transitive, or passive main clause. The intransitive verb phrase, VP_INTRANS, can either be a standard verb with an optional prepositional phrase or the verb-to-be followed by a predicate adjective. The transitive verb phrase, VP_TRANS, permits either a direct object phrase with an optional prepositional phrase (which could be of the form "to recipient") or a dative recipient followed by the direct object. Verbs for which a recipient is not appropriate do not use this option.

In order to avoid recursively embedded relative clauses in this simplified language, the VP_TRANS2 verb phrase is used within relative clauses. It does not permit the direct object to be modified by an additional relative clause or prepositional phrase. The VERB used in the active constructions can have one of six tenses, including the simple and progressive forms of past, present, and future. The S_INFIN and INFIN are simple present tense forms of the verb which may have different singular and plural aspects, e.g. "runs" and "run". VP_PASSV is a passive construction with an optional agent. It too can be in the past, present, or future.

A relative clause can either be subject relative, object relative, passive, or a reduced passive which lacks the introductory relative pronoun. The SP and OP are really full noun phrases which may include a relative clause or prepositional phrase. NP is just the head of the noun phrase which includes the noun and optional article and adjective. The verb may be modified by an adverb or one of five prepositional phrases describing manner, instrument, recipient, destination, or location. The only noun-modifying prepositional phrase at the moment describes possession. However, this does permit the language, in theory, to produce such attachment ambiguities as, "The boy saw the girl with binoculars."

On top of this base are a number of lexical constraints which can eliminate certain sentences or modify the frequency of others. Nouns constrain their articles, adjectives, and possessions, and agents constrain their verbs and patients. Verbs constrain a number of other constituents and structures including their patients, adverbs, recipients, and other elements in prepositional phrases. All of these constraints ensure that the language only produces sentences that syntactically and semantically are plausibly English. They were incorporated into the context-free base of the language using the Simple Language Generator (Rohde, 1999b), which is then able to produce or analyze sentences from the language.

The initial vocabulary used in these experiments is rather small, with 12 noun stems, 12 verb stems, 6 adjectives, and 6 adverbs. Including the function words and morphological variants, the language uses 111 different words. However, it is likely that the networks will have less trouble with an expanded vocabulary than with an expanded grammar.

The following are some examples of sentences generated by the language:

> an apple will be stolen by the dog .
> mean cops give John the dog that was eating some food .

Table 2: The context-free base for the grammar used in the following experiments.

S	→	> S2 .
S2	→	SP VP_INTRANS (0.3) | SP VP_TRANS (0.5) | OP VP_PASSV (0.2)
VP_INTRANS	→	VERB V_PREP (0.8) | IS ADJ (0.2)
VP_TRANS	→	VERB OP V_PREP (0.7) | VERB RECP OP (0.3)
VP_TRANS2	→	VERB NP V_PREP (0.7) | VERB RECP NP (0.3)
VP_PASSV	→	P_VERB V_PREP (0.5) | P_VERB V_PREP by NP (0.5)
VERB	→	PAST | WAS PRES_PART | S_INFIN | INFIN | IS PRES_PART | will INFIN | will be PRES_PART
P_VERB	→	WAS PASSIVE | IS being PASSIVE | will be PASSIVE
RC	→	THAT VP_INTRANS (0.25) | THAT VP_TRANS2 (0.3)| THAT NP VERB (0.15) | THAT VP_PASSV (0.2) | PASSIVE by NP (0.1)
SP | OP INSTR | RECP | DEST | LOC |	→	NP (0.7) | NP RC (0.2) | NP N_PREP (0.1)
NP	→	ART N (0.6) | ART ADJ N (0.4)
V_PREP	→	ϵ (0.6) | ADVERB | with MANNER | with INSTR | to RECP | to DEST | in LOC
N_PREP	→	with NP
WAS	→	was | were
IS	→	is | are
THAT	→	that | which | who

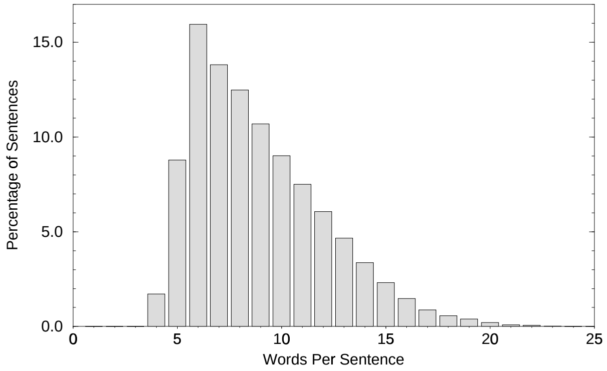

Figure 5: A histogram of sentence lengths in the training language.

> boys will give cats some apples .
> the cat with a picture is being taken by a dog .
> a cop was giving the boys a picture .
> cats are chasing the mean boy in a house .
> the boy drove .
> John who is being chased by the fast cars is stealing an apple which was had with pleasure .

The average sentence length is 8.9 words, including start and end symbols. A histogram of sentence lengths is shown in Figure 5. Networks were trained on a fixed set of 50,000 sentences generated from the language. This included 37,477 unique sentences. A testing set of 5,000 sentences was generated in the same manner as the training set, comprising 4,475 different sentences, 3,452 (77%) of which did not appear in the training set. Because of difficulties some of the semantic encoding networks have had with the full language, we also generated corpora composed of sentences restricted to at most 10 words in length, eliminating 28% of the sentences. The training set for the simple version has 31,210 unique sentences and the testing set 4,164, 2,699 (65%) of which did not appear in the training set. This *simple corpus* was used for the comprehension and production experiments reported here.

A few of the potentially important features of English that are not included in the current grammar are questions, imperatives, pronouns, compound sentences, modal verbs, idiomatic constructions, and various phrase and clause types including gerund phrases (“*Using foul language* is bad”), infinitive phrases (“I have a bone *to pick with you*”), and sentential complements (“I know *that she is leaving*”). However, once a reasonable network architecture is in place, it will be a relatively simple matter to expand the language or adapt it to a particular experiment.

6 Encoding Messages

Comprehension, as viewed in the current model, is the task of mapping a series of words to a static representation of the message. Of crucial importance, then, is how messages are to be encoded. One approach might be to attempt to hand-code a representation. However, this would be quite difficult without using very large and redundant structures. Another approach is to train a network to develop its own representations of sentence structure. The ability of the network to take advantage of redundancy should result in more compact encodings. I expect that learned representations will be more similar to those that underlie language in the brain and will thus contribute to the predictive power of the model.

The notion of the meaning of a sentence, or message, used here is based on what one might consider its *semantic parse tree.* Although this is a recursively composed structure, it should not be confused with the syntactic parse tree. The semantic parse tree is not based on the words of the sentence but the constituent concepts and their relationships. A message has a single representation regardless of the language in which it is expressed. Figure 6 illustrates a syntactic parse tree for the sentence, "The dogs that were chasing a car chased the boy with some fast cars in a big house," and Figure 7 illustrates its semantic tree.

The semantic tree, as defined here, is a ternary tree. In other words, each non-terminal has exactly three children. The left child of every non-terminal defines a relationship that exists between the middle child and the right child. Consider, for example, the non-terminal labeled K-M. The relationship encoded in the left child is that of a property. The middle child represents the concept of multiple cars. The definiteness property of the cars is not specified and, in this language, that will translate into "some cars". The right child of the non-terminal is the property "fast". Therefore, this non-terminal represents the concept of "some fast cars". The I-M non-terminal represents the concept of a particular boy who owns the fast cars. Thus, this portion of the tree might be translated as, "the boy with some fast cars".

For the purposes of constructing the semantic tree, we treat a transitive verb as a relationship between the subject and the object. For example, non-terminal F-H indicates that the dogs were chasing a car. The *past* and *extend* features of the action indicate that it is in past progressive tense. However, because the *that* feature of the action has been activated in this case, it indicates that the action is a subordinate one, and is expressed in a relative clause. Therefore, this portion of the tree can be translated as the noun phrase, "the dogs that were chasing a car."

We define the *head* of a sub-tree in the semantic network to be the terminal node directly under the root of the tree, following the chain of middle children. Therefore, the head of the A-E sub-tree is terminal B, which represents the action associated with the verb "chased". This sub-tree, then, itself serves as an action or a verb phrase. The head of the entire tree will always be the subject, which is terminal G, representing "the dogs" in this case.

One important consideration in the design of the semantic parse tree is how passive constructions are to be represented. There are two possible interpretations for the action relationship. We could either think of there being an (*action*, *agent*, *patient*) triple or an (*action*, *subject*, *object*) triple. In the case of an active sentence, such as "The boy chased the dog", these yield the same result: (chased, boy, dog). However, the passive sentence, "The dog was chased by the boy", could either be represented as (was chased, boy, dog) or (was chased, dog, boy). There is good reason to prefer the former, which follows the rule (*action*, *agent*, *patient*). In this case, the semantic representations of sentences such as "The boy chased the dog" and "The dog was chased by the boy" differ only minimally. However, using that encoding would violate the rule that the head of the sub-tree always falls at the bottom of the middle path. For example, the phrase "dogs who were chased by boys" would be encoded as (who were chased, boys, dogs) and the head of this phrase, dogs, would be relegated to the right-hand position.

Therefore, the subject or focus of a relationship is currently placed in the middle position. In an active construction, the agent takes the middle position and in a passive construction the agent fills the right-hand position. This may need to be reconsidered when we attempt to model human data on the processing of passives. However, it is interesting to note that RAAM networks, which will be discussed in Section 6.1, appear to be significantly more successful at learning to compress the semantic trees when the subject, rather than the agent, is always placed in the middle position.

A major limitation of the current semantic tree representation is that it has only been designed to encode declarative sentences. However, questions and commands will be of critical importance in modeling early development as they constitute the majority of utterances addressed to children, according to the CHILDES database (MacWhinney, 1999). There are several ways in which these could be handled. Wh-questions might be encoded using a similar representation to the corresponding statement but with a special code in place of the missing constituent. For example, if, "Newt chased the dog," is encoded with the relation (chased, Newt, dog), then, "What did Newt chase?," might be encoded (chased, Newt, ?). The representation of the verb might also be altered slightly in this case. Imperative commands could be represented by explicitly placing a "you" in the subject

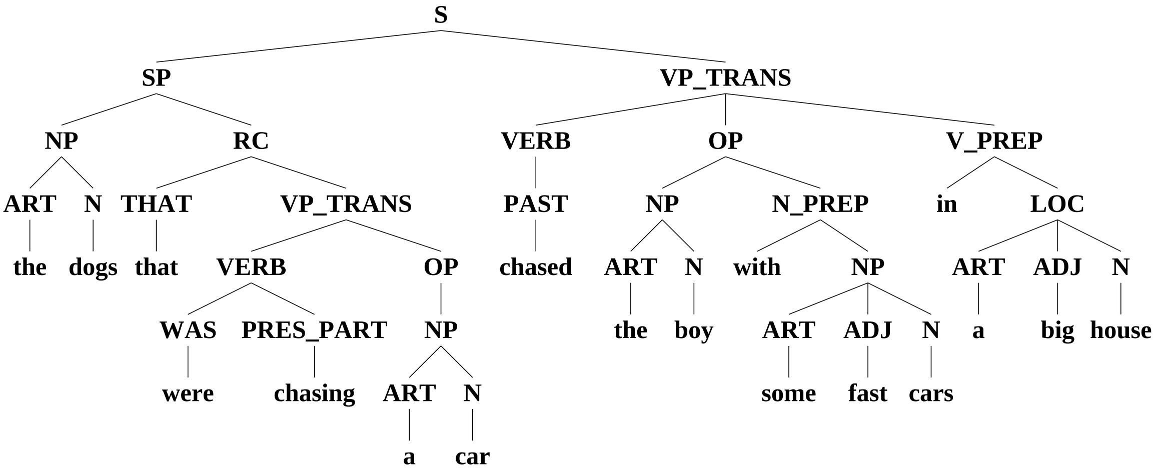

Figure 6: A *Syntactic* Parse Tree.

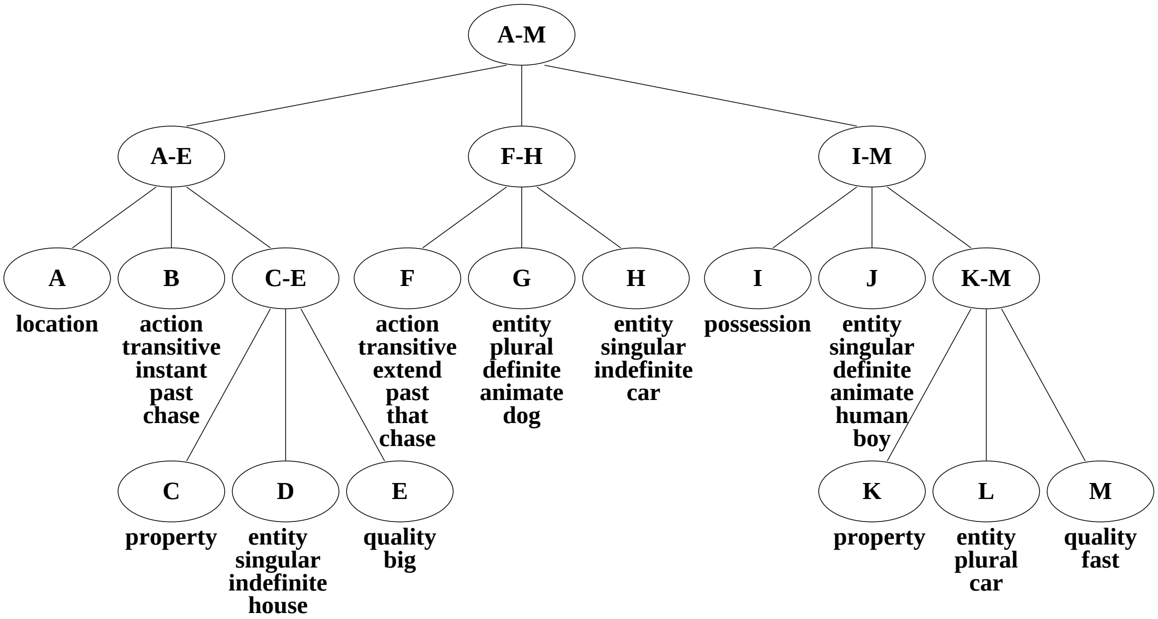

Figure 7: A *Semantic* Parse Tree.

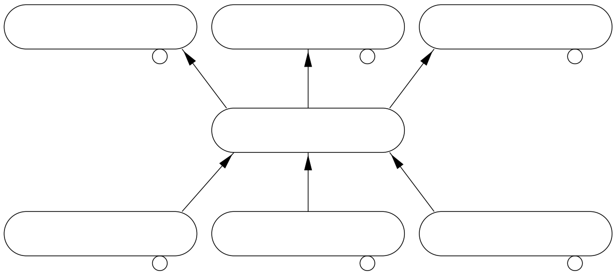

Figure 8: A very simple RAAM network for encoding ternary trees.

slot and marking the verb representation. It should also be relatively easy to extend the semantic tree to encode conjunction, disjunction, subordination, and temporal ordering of propositions.

The semantic trees of sentences in the training corpus average 5 terminals and 2 non-terminals. The largest trees have 17 terminals and 8 non-terminals. In the simple version of the language, which limits sentences to 10 words, trees average 4.1 terminals and 1.6 non-terminals, the largest with 11 terminals and 5 non-terminals.

6.1 Recursive auto-associative memories

The semantic parse tree represents a message using a variable-size structure but for the purposes of comprehension we would like a static representation that uses a vector of fixed dimension. To accomplish this, networks will be trained to compress the information in the semantic tree in such a way that the information can be later decoded.

One obvious candidate architecture for the encoder network is a recursive auto-associative memory or RAAM (Pollack, 1990). These networks are specifically designed for compressing and decompressing tree structures. A simple version of a RAAM is illustrated in Figure 8. The network has 7 basically equal-sized groups. The three input groups project to a single hidden layer which projects to three output groups. The network is trained as an auto-encoder: it learns to produce the same representations on the output groups as are presented on the input groups. However, because the activation must feed through the single hidden layer, the network must form a compressed representation of the inputs at the hidden layer.

The RAAM can be used to encode entire trees by recursively compressing each sub-tree from the bottom up. In encoding the tree from Figure 7, the network begins with the C-E sub-tree. The representations of terminals C, D, and E are clamped on the input groups and the network attempts to reproduce these representations on the output groups. In doing so, the hidden layer encodes the entire sub-tree. Then the network can encode the A-E sub-tree by clamping the first two input groups to A and B and clamping the third input group to the compressed representation of C-E. Once each sub-tree is processed in this way, the hidden layer will contain a single representation of the entire message. If the compression was successful, the network should then be able to expand this representation into encodings of the three sub-trees and each of those recursively until the entire tree is reconstructed. In order to know when to stop the recursive decoding, a bit is added to the input and output units indicating whether the representation is of a terminal or a non-terminal and only non-terminals are recursively expanded.

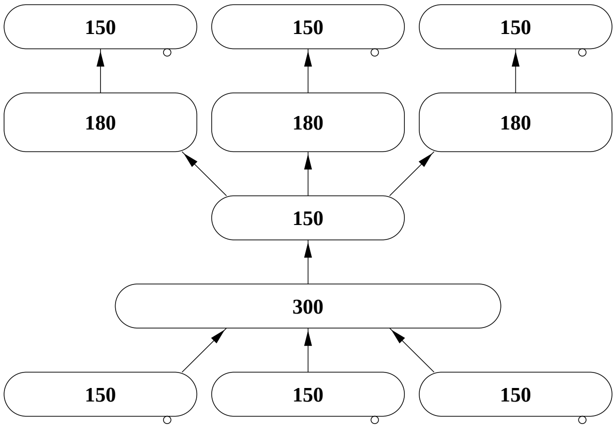

Figure 9: The RAAM network actually used in the current work.

In practice, the performance of RAAMs can be improved by adding hidden layers to allow non-linear mappings between the input and bottle-neck and the bottle-neck and output layers. In order to learn the training language, the architecture shown in Figure 9 was used. Although the representations of terminal nodes only use 70 bits, the input, bottle-neck, and output layers each have 150 units, the extra space being used for encodings of non-terminals.

Units in the network used logistic output functions and cross-entropy error was applied on the outputs. A target radius of 0.1 was used on terminal representations to help the network focus on the bits that were truly wrong. This makes the effective targets 0.1 or 0.9 rather than 0.0 or 1.0 but does not penalize the network for exceeding the targets. Weights were randomly initialized in the range ± 0.3. Weight updates were performed every 10 sentences using bounded momentum with a momentum of 0.95. The learning rate was 0.1 initially and was scaled down by 0.9 every 10,000 updates. The network was trained for 40 passes through the training corpus or 2 million sentences. On the last 10,000 sentences the learning rate was dropped to 0.001.

The RAAM network was ultimately quite successful at encoding and decoding the sentences in the testing set. The structure of the decoded trees was nearly perfect. Of the 10,067 non-terminals in the corpus, the decoded trees incorrectly labeled only 3 of them terminals. Of the 25,134 terminals, 4 were incorrectly labeled non-terminals, 2 of which occurred on a single very difficult sentence. A terminal representation was considered correct if the activations of each of the 151 units was on the correct side of 0.5. The RAAM reproduced the correct terminal representations for 97.6% of the terminals, an error rate of 2.4%. In terms of individual output units, that is an error rate of just 0.025%. Given that nearly 70% of the testing sentences were novel, this indicates a remarkable ability to generalize.

When tested on the simple corpus, which only includes sentences with 10 words or fewer, the network made no structural errors in reproducing the trees and got just 0.28% of the terminals wrong. Therefore, as one might expect, nearly all errors are made on the long, complex sentences. Although the errors have not been formally analyzed, it appears that, when the network makes an error in a terminal representation, it tends to either flip the value of a single bit or replace the correct bit with a semantically similar one, such as turning a boy into a girl. Also, when tested on the large language, the network appears to have major breakdowns with the longest sentences. Rather than degrading gradually as sentence complexity increases, the network appears to either decode the tree perfectly or make a host of errors. This may be similar to the breakdown that humans observe in attempting to parse sentences that go beyond our capabilities.

6.2 Query networks

Despite the evident power of the RAAM, we suspected that the semantic encodings it produces may not be appropriate to serve as targets for comprehension. Because the RAAM encodes a sentence recursively, the relationship between a deep terminal and the final result may be rather complex, the information having been passed through multiple non-linear mappings. Thus, the RAAM tends to produce gordian encodings which do not have clear relationships to the surface elements of the semantic tree. Additionally, the RAAM process is arguably a symbolic one, requiring representations and structures to be stored in memory and manipulated rather like variables. Thus, a more appropriate encoding from a connectionist perspective might be one that does not explicitly attempt to store the structure of the semantic tree, but just the important relationships within the tree. As McClelland, St. John, and Taraban (1989) suggest, "representations of sentences are not required to exhibit a specifically propositional format *so long as they can be used to perform the tasks we require*" (p. 296).

To this end, we define, for each semantic tree, a set of *semantic triples* which capture the relationships between the main constituents. Each non-terminal symbol is associated with a single triple. The triple is composed of the head constituents of each of the three children of the non-terminal. For example, the triple associated with the root node, A-M, of the tree in Figure 7 would be (B, G, J) or (B:chased, G:dogs, J:boy). Whereas each non-terminal defines a relationship between three sub-trees, the triple associated with that non-terminal is the relationship between the heads of the three sub-trees. Because the example tree has six non-terminals, it also has six triples:

(B:chased,	G:dogs,	J:boy)
(A:location,	B:chased,	D:house)
(C:property,	D:house,	E:big)
(F:that were chasing,	G:dogs,	H:car)
(I:possession,	J:boy,	L:car)
(K:property,	L:car,	M:fast)

Our goal, then, is to train a network to remember these triples and the representation used to do so will be considered the message. One method to accomplish this would be to train a network to store the triples, one at a time, in such a way that they could later be produced in the same order in which they were stored. However, such tasks tend to be rather difficult for networks to learn. Because the network must remember not just the triples but the proper order of the triples, the network's ability to take full advantage of redundant information to develop a nice, compressed representation of the full sentence will be hindered.

A better solution, referred to here as a *query network*, is borrowed from St. John and McClelland (1988). The query network is trained to store the triples in such a way that it can, in effect, answer queries about those triples. The network is queried by probing it with a triple which has one of its three constituents removed. When the probe is presented to the network on a special set of inputs, the network responds by producing the complete triple with the missing information filled in. The architecture of the query network used in this work is shown in Figure 10.

The network is basically a simple-recurrent network and all units use logistic transfer functions. During training, triples are presented to the network one at a time at the main input layers. A representation of the semantic information so far builds up at the "message" layer. After each triple is presented, the activations of units at the message layer are frozen and the network is probed about its knowledge of each of the constituents in each of the triples presented so far. When a constituent is probed, the activations of the "query input" group corresponding to the constituent are set to 0.5 and the activations on the other two query input groups are clamped to the appropriate patterns for the other two constituents of the triple. The network is then trained to fill in the missing constituent, producing

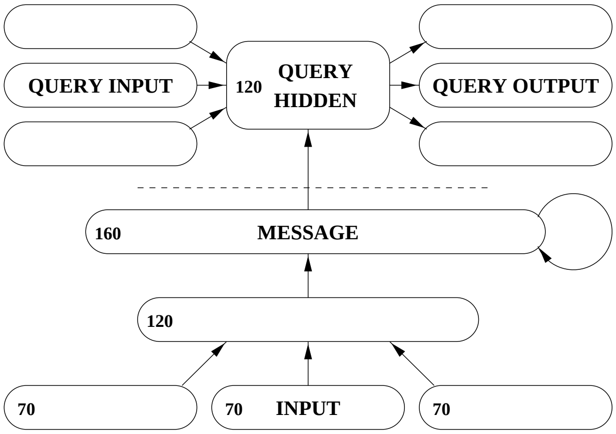

Figure 10: The query network used in the current work.

the complete set of patterns on the query output groups. There may be times when there are more than one possible answer to a query. For example, given the triples (property, dog, red) and (property, cat, red), how should the network respond to the query (property, ?, red)? Currently, we identify such ambiguous queries while testing the network and do not penalize its response on the output units that differ in the various answers.

To train the network on a semantic tree involving four triples, three constituents would be probed after the first triple is presented, six after the next triple, and then nine and twelve, for a total of 30 queries. Because the number of queries grows quadratically with the number of triples, complex sentences can take quite long to train. Thus, a straight-forward implementation of the query network is rather slow, taking four times longer than the RAAM. However, a basic implementation involves a considerable amount of redundant computation. When querying a particular message, the contribution of the message group to the input of the "query hidden" group does not change. Therefore, that projection could be traversed once and the results stored. Likewise, the query input groups frequently repeat the same patterns and their contribution to the input of the query hidden layer need not be recomputed. There can be significant time savings in the backward pass as well. If a particular query input group has the same representation on five time steps, the derivatives of the error with respect to the query hidden group inputs for those time steps can be summed and only backpropagated once. With all of these optimizations in place, the query network can be trained as quickly as the RAAM.

The encoder half of the network was trained with a modified version of backpropagation-through-time which propagated error back through the recurrent connections in the message group to the two previous input events. This helps the network form representations at the message level that will be extensible when new triples are presented. Like the RAAM, the query network was trained on 40 passes through the training corpus for a total of 2 million sentences. Weight updates were performed every 10 sentences using bounded momentum descent with a momentum of 0.95. Cross entropy error was used with a target radius of 0.1. The learning rate began at 0.1 and was reduced by 10% every 100,000 sentences and then cut to 0.001 for the last 100,000 sentences.

The query network was ultimately not as successful as the RAAM but still achieved reasonably good performance. The network was able to correctly respond to 91.9% of the queries on the testing set for an error rate of 8.1%. That is counting only the constituents which were not provided in the query input. Overall, only 2.8% of constituents had errors. The error rate on the missing constituents in terms of units was 0.23%. When the network does make a mistake on a constituent, it tends to get an average of about 2 bits incorrect. However, most of these errors do occur on the more difficult sentences. When tested on the set of simpler sentences with 10 words or fewer, the network had an error rate on the missing constituents of just 1.0%, rather than 8.1%.

Thus, although the query network could stand some improvement, which may come about with larger networks and better training methods, it is reasonably capable of encoding the sentences and, as we'll see, provides a better basis for learning comprehension than the RAAM. One drawback of the query method, however, is that the message representation can be ambiguous. For example, the sentences, "The red cat chased the cat," and "The cat chased the red cat," would both be represented by the triples (chased, cat, cat) and (property, cat, red). It is possible to augment the representations of constituents based on their position in the semantic tree to eliminate ambiguity, but that has not been done in the current simulations because ambiguity is relatively rare and should become even less of a problem as the vocabulary of the training language increases.

A more serious problem with the query method is that it is not possible to fully analyze the network's message representation. Although the network may be able to answer the queries correctly, it may mistakenly fabricate information about the sentence that is simply never queried. Thus, we might ask what the appropriate response should be if the network is queried with two constituents that are not drawn from triples in the semantic tree. For example, given the sentence, "The boy chased the dog," how should the network respond when asked the color of the cat? The RAAM does not suffer from this problem because, when the semantic tree is decoded, we discover all of the information in that representation. Short of ignoring it, one solution to this problem might be to train the network to respond with an "invalid query" signal when a bad query is presented. However, this would greatly expand training time and would still make analyzing the network's representations rather difficult.

Another solution might be to change the querying mechanism to one that actually forces the network to form a representation of the semantic tree but one that is less opaque than the RAAM encodings. For example, one might query the network with an encoding of a path in the tree and have the network identify the node at the bottom of that path as either a terminal or a non-terminal and provide its representation in the case of a terminal, but this could be rather difficult for the network to learn. A better solution may be a combination of the RAAM and a query network. This would take the form of a RAAM that is forced to answer queries about the triples stored in each of its sub-trees. This would have all the advantages and disadvantages of the RAAM but would hopefully produce encodings that are better suited for comprehension. Although it has been implemented, the query-RAAM has yet to be fully trained or tested.

7 Comprehension and Prediction

The primary goal of the comprehension network is to take as input the sequence of words in the sentence and produce as output the message encoded using either the RAAM or query network. Simultaneously, the network may be trained to predict the next word. In order to simplify the model, we make the assumption that the correct message is available to the learner in the form of a target while the sentence is being observed and that these representations are fully developed at the start of training. In reality, however, our ability to represent complex ideas is developing at the same time as—and is partially driven by—language processing.

It might be more reasonable to assume that task-relevant feedback, similar to the error signals produced by probes in the query network, directly influences sentence processing (as suggested in Allen, 1987; Elman, 1990; Miikkulainen & Dyer, 1991; St. John & McClelland, 1990; St. John, 1992a), as opposed to using a visible semantic representation of the sentence. However, I found in preliminary tests that only relying on direct probing of the message in a comprehension network was time consuming and not particularly effective. It is also harder to introduce context information during comprehension and, especially, production. Another advantage of using visible, albeit trained, semantic representations is that the same representations could be used for multiple languages, creating a bilingual network capable of translation. Therefore, in the experiments reported here, the pre-learned semantic encodings were used as the targets for comprehension.

The basic architecture of the comprehension/prediction network used in these experiments is shown in Figure 11.

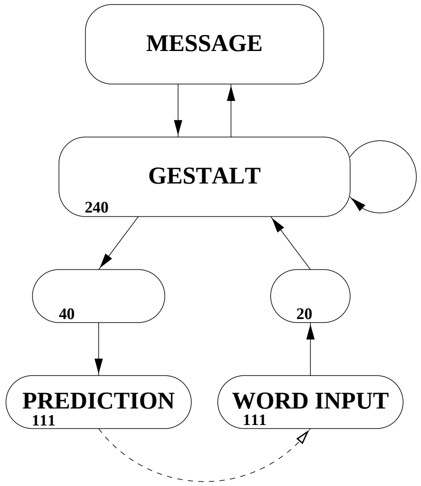

Figure 11: The basic comprehension/prediction network.

7.1 Comprehension with RAAM encodings

This network was first trained with the RAAM message encoding without using the prediction portion of the network. The word hidden layer contained 20 units and the gestalt layer 200 units. Initial weights were drawn from the range ± 0.3 except for those of links projecting to the first small hidden layer, which were in the range ± 2.0. The network was trained for 20 passes through a corpus of 50,000 sentences. Only sentences with fewer than 11 words were used in training and testing due to the expected difficulty of the task. The testing set consisted of 5,000 sentences, as reported in Section 5.2. Weight updates were performed with bounded momentum descent with a momentum of 0.9 and a learning rate of 0.2 annealed to 0.04 over the course of training and reduced to 0.001 for the last pass through the corpus.

The network was rather successful in inducing messages that capture the structure of the correct semantic trees of the testing sentences. During the decoding process, 38 of the 20,528 terminals (0.19%) were incorrectly labeled as non-terminals. 58 of the 7,764 non-terminals (0.74%) were incorrectly labeled terminals. However, of the 20,316 terminals produced during decoding, 3,905 (19.2%) had mistakes on at least one unit. Although the error rate per unit was only 0.29%, this was still too high to be considered adequate comprehension.

It is likely that the problem with the RAAM encodings is two-fold. The first is that the encodings are not particularly binary, relying heavily on the sensitive middle range of activation values. Because it is difficult for the comprehension network to reproduce these values exactly, there is slight error in the initial representation which is magnified during decoding. This is a well-known problem with RAAM representations. There are several possible solutions, one of which is to add a term to the error measure that encourages the RAAM hidden units to use values close to 0 or 1. Over time this will lead to binary representations and can lead to better performance given sufficient hidden units. However, this can substantially degrade performance if there are not enough hidden units, as appears to have been the case when this technique was applied to encoding the training language. Nevertheless, the second problem with RAAMs may be more fundamental. Because the encodings are produced by a recursive process, they tend to be complex. Thus, there is unlikely to be a simple mapping between constituents of the sentence and principle components of the message representation except for the simplest sentences. This may be alleviated by the combined RAAM and query network discussed above, but this remains to be seen.

7.2 Comprehension and prediction with query encodings

In the next experiment, a comprehension network was trained using the message encodings produced by the query network. The architecture was the same as before, although the message representation comprised 160 units rather than the 150 for the RAAM encoding. The training methods were the same as before. The network was tested on its ability to respond to each probe based on the message representation produced during comprehension. It was much more successful than the RAAM comprehender, filling in the correct representation on all but 704 of the 23,292 probes on the testing corpus (3.0% error). This is approaching the lower limit of 1.0% error in the ability of the query network to encode and decode messages. When testing on 5,000 sentences drawn from the training set, the error rate only dropped from 3.0% to 2.6%. Thus, the network was just 15% worse on the mostly novel testing set, indicating a high degree of generalization. On average, each incorrect response had 1.9 bits wrong. The majority of the errors involved either failing to activate one of the units in the output, or swapping two mutually exclusive bits, such as reporting an action as extended in duration rather than instantaneous.

The networks described thus far were trained only on comprehension without prediction. The remaining networks were trained, as each word was presented, to produce both the complete message and a prediction of the next word in the sentence. Although the message output layer used logistic units with cross-entropy error, the prediction output group used a soft-max constraint (Luce, 1986) with the divergence error measure. Again, the same training methods were used as for the RAAM and query comprehender. Although prediction did not appear to help the network, it did not significantly impair performance. The error in responding to probes on the testing set increased from 3.0% to 3.2%.

7.3 Center-embedded versus right-branching sentences

It is interesting to examine the extent to which the network's difficulty comprehending certain sentence forms matches that of human subjects. The network's performance was evaluated on corpora of sentences involving a single relative clause modifying either the subject (center-embedded) or object (right-branching). The relative clauses were either all subject-relative (“that chased dogs”), all object-relative (“that dogs chased”), or all passive (“that was chased [by dogs]”). The error rates on these six classes of sentences is shown in Table 7.3. Like humans, the network has a significant preference for subject-relative sentences and a preference for passive embeddings over object-relatives. Although deeper factors may play a role, these results could reflect a pure frequency effect.

However, the network also shows a strong preference for center-embeddings, in which the subject of the sentence is modified, over right-branching sentences, in which the direct object is modified. This can partially be attributed to a frequency effect. Because the noun modified by a center embedding is the agent of the sentence, it tends to be a more typical agent and thus has a higher probability of having a subject-relative. Because the direct object is often inanimate, it is less likely to be modified by a subject-relative. Therefore, much of the advantage for center-embeddings could be due to a greater proportion of subject-relatives as well as a greater frequency overall. However, the network also shows a preference for center-embedded passives and object-relatives over right-branching passives despite lower frequency. This may be due to the fact that passives and object-relatives are actually encoded more like active sentences in the current semantic tree with the subject, rather than the agent filling the center position. This could cause a bias against right-branching sentences because the modified noun serves as the object of one clause and the subject of the other, which may be confusing.

Although we should not put much stock in these results since the network was not trained on a well-controlled language, they do raise some interesting questions. Should we believe the conventional wisdom that center-embeddings are really harder? The traditional story is that center embeddings are difficult because information must be remembered across the embedding (Weckerly & Elman, 1992). While that is true of a prediction network, which has the luxury of discarding unneeded information, that is not true in comprehension. In

Table 3: Frequency (per 100,000) and comprehension errors on single-embedding sentences.

Embedding	Center		Right		Overall	
Type	Freq.	Errors	Freq.	Errors	Freq.	Errors
Subject-relative	1130	**8.0%**	750	**13.3%**	1850	**10.6%**
Passive	360	**14.1%**	440	**20.6%**	760	**17.7%**
Object-relative	120	**20.3%**	150	**27.7%**	250	**22.7%**
Overall	1600	**10.0%**	1350	**15.9%**	2930	**13.1%**

comprehension, *all* useful information must be remembered until the end of the sentence. A right embedding is potentially more difficult because the system must remember the entire sentence, rather than just the subject, across the embedding. While this may not pose a problem for symbolic models, maintaining working memory is difficult for humans and neural networks.

As Christiansen (1994) argued, "The purported human ability to deal with an unlimited length of [right-branching] recursion has, to my knowledge, never been demonstrated in an experimental context." (p. 101) Historically, experiments dealing with the center versus right distinction seem to have confounded it with the subject-versus object-relative distinction (Blaubergs & Braine, 1974; Weckerly & Elman, 1992) and whether a relative pronoun is used. This is partially a necessity since multiply nested center embeddings must be object-relative in English. However, the two could be separated in the case of single embeddings as in my preliminary experiment. Tellingly, Blaubergs and Braine (1974) found no significant difference in the difficulty of single object-relative center embeddings and single subject-relative right embeddings, perhaps indicating that the object disadvantage is balanced by a center advantage.

In any case, this appears to be an area worth studying in greater depth. In particular, I plan to perform an analysis of the frequencies of various embedded structures in spoken English. These data will then be used to design a more realistic language on which to train the model. At that point, we could verify the model's predictions with a more carefully controlled study of human comprehension abilities. Questions of particular interest will be how much of human performance can be explained by frequency effects, center versus right embeddings, subject versus object relatives, marked versus reduced relatives, and semantic plausibility. This should serve both to extend our understanding of human language abilities and as an important validation of the model. As is often the case in modeling, when we set out to analyze a model we discover that our understanding of the human performance against which it will be judged is inadequate.

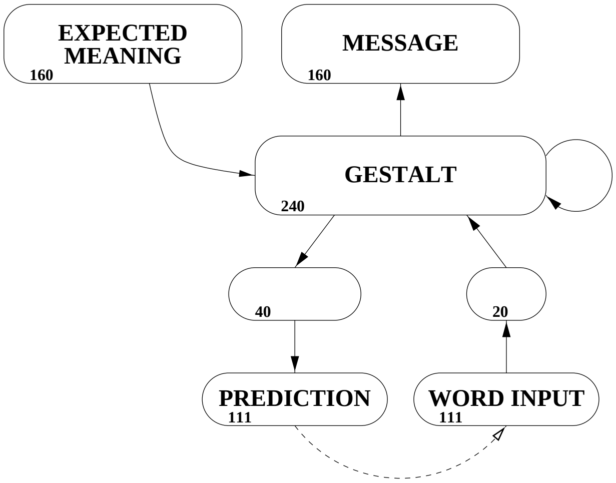

Figure 12: A comprehension/prediction network using an external "expected message" group.

7.4 Context

Given that we have made the assumption that targets for training comprehension are available throughout the presentation of the sentence, it is not a major step to assume that those targets are available as activations to influence comprehension and prediction. In essence, we hypothesize that the listener has a good idea of what will be said and the comprehension and prediction systems can use this information to refine their behavior. It would not necessarily be reasonable to assume that this context is always present and precise, but we would like to be able to provide the network with partially reliable, possibly noisy context.

There are several ways in which the semantic context could be presented to the network. One method is shown in Figure 12. In this case, semantic context, or the "expected message", resides in a separate input group that has a standard projection to the gestalt layer. In order to equate the number of links with the other networks, the projection from the message down to the gestalt layer was removed. Clearly we would not always want to give the network the complete message in advance or it would learn to ignore the words and simply copy the expected message to the derived message. However, if we only

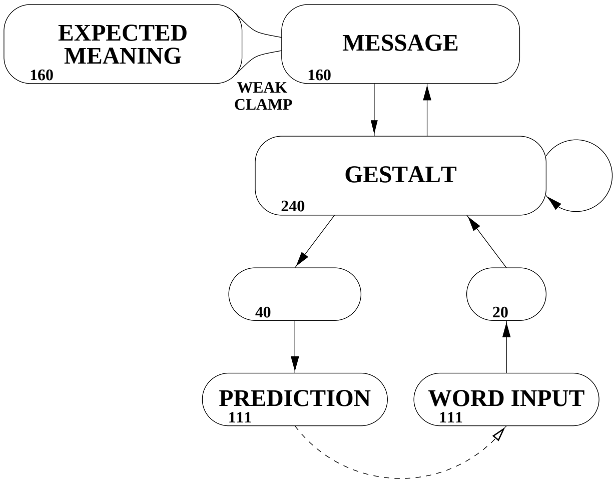

Figure 13: A comprehension/prediction network introducing expected message by weak-clamping the message layer.

provide the message on some trials the network should learn to rely primarily on the actual words. One advantage of this method of introducing expected messages is that the network can be given partial messages. For example, we might only load some of the semantic triples into the query network and use that partial representation as the expected message, as if the learner basically knew what was to be said, but not entirely.

A network was trained in similar fashion to those above, except that the error-free message was provided on 20% of the trials and no context was given on the other trials. When tested with the message present, the network not surprisingly performed very well, with a comprehension error rate of just 1.5%, compared to a rate of 3.2% when trained with no context. However, without the context provided, the network had an error rate of 7.4%. This may partially be due to the loss of the back-projection from the message layer and it would be informative to train a network with that projection intact. However, the poor performance is more likely due to a fundamental problem with this method of presenting context. The network must learn to essentially operate in two modes: with context and without and this seems to impair its performance when context is not available. Experiments with using partial context—a semantic representation without all of the triples loaded—resulted in very poor performance.

An alternative way to introduce context to the comprehender/predictor is shown in Figure 13. Rather than using a separate input with a standard projection, the meaning layer is "weak-clamped" by the expected message. Unlike a hard-clamp, which sets the outputs of a group, a weak-clamp merely pulls the outputs of the units a certain percentage of the way towards the targets. Thus, if the original output of the unit were 0.5 and the target were 1.0, a weak-clamp with a strength of 0.4 would cause the resulting output to be 0.7—it has moved 40% of the distance to the target. We also have the option of initializing the message layer to all 0.5's, to the actual expected message, or to a weak version of the message.

The network was trained as before. On half of the sentences there was no weak clamping and thus no context information. On the other half of the trials, the message layer was weak-clamped with a strength of 0.25. Thus, after each word the message was pulled 25% of the way towards the correct representation. On 40% of the weak-clamped trials, the message layer was initialized to the full expected message. On the other 60% of the weak-clamped trials, the message layer was initialized to the message weakened by either 25%, 50%, or 75%, with equal likelihood.

Results for this network were significantly better than for the other method of presenting context. With no context provided, the network has an error rate of 3.5%. This is just a bit worse than the network that was never trained with context. With the activation of the message layer initially set to the expected message weakened by 50% and with a clamp strength of 25% subsequently, the network has an error rate of 2.0%. The network actually performs a bit worse when the initial message is stronger. Nevertheless, this network has learned to comprehend quite well in the absence of context and, as we'll see in the next section, produce very accurate predictions when context is available.

8 Production

Having trained a comprehension and prediction network in the context of expected messages, one could immediately simulate production. Production begins by presenting the network with the message it is to convey and feeding the sentence "start" symbol into the word input. The network produces a distribution over the possible first words at the prediction layer and one word is selected from this distribution and fed back into the network. This process continues until the "end" symbol is produced. As the network generates the sentence, it continues to monitor itself and will possibly alter the intended message. This may have the beneficial effect of helping the network recover from unintended productions and, in a more detailed system, would allow the network to revise its thoughts in midstream, as people seem to do when attempting to convey a complex or vague idea. However, allowing the intended message to drift during production could result in the network losing track of its thoughts entirely, resulting in a sort of fluent aphasia.

Table 4: Sample incorrect productions. The upper sentence is the desired one and the lower is the network's production.

(1) the boys who were sitting will be sitting .
the boys who are sitting were sitting .
(2) food that was made is old .
food that is old food is tasty .
(3) a boy makes a picture which was made .
a boy made a picture which is beautiful .
(4) apples that were gotten were sitting .
apples which were gotten were sitting .
(5) boys were giving a mean cat some food .
boys were giving the mean cats food .
(6) a cop who was stealing is leaving .
a cop who is stealing is stealing .
(7) some cops that drove are stealing the dog .
some cops that stole are stealing the dog .
(8) the boy was chased by the beautiful cat .
the cat was chased by the beautiful cat .
(9) the big car is chasing the big cars .
the big car is chasing the big cars is chasing the big cars .
(10) some food which was eaten with pleasure sat .
some food which was eaten sat .
(11) the boy gave food to a cat .
the boy gave a cat food .
(12) a big boy with the car was sitting .
a big big big big big big big big big big big big big...

The comprehension/prediction network, described in the previous section, which was trained using weak-clamped context was tested on its production abilities. At the start of production, the full semantic representation of the sentence was loaded into the message layer. Subsequently, the message layer was weak-clamped to this intended message with a strength of 0.25. The most active word in the network's prediction layer was chosen as the next word in the sentence. Overall, the network produced 76% of the sentences correctly, which is comparable to the results published by Dell, Chang, and Griffin (in press), although that model only produced simple sentences and used an external mechanism to maintain context.

It is interesting to examine the errors made by the network. Some illustrative examples are listed in Table 8. Most of the incorrect productions were still grammatical sentences. In the first example, the verb tenses are confused. Sentence (2) illustrates two problems, the movement of the matrix clause to a subclause and fabrication of details. As we might expect, fabrications tend to be semantically plausible. In (3), a rather awkward sentence appears to have been improved. Sometimes errors could be as subtle as changing the relative pronoun, as in (4), or confusing the noun number or definiteness, as in (5).

One of the most common mistakes made by the network is duplicating or perseverating an element of the sentence. In (6) the embedded verb takes over the matrix clause and the opposite occurs in (7). A similar effect occurs with nouns, as in (8), and with adjectives. Perseveration is also quite common on the phrase level, as in (9). As one might expect, phrases are also dropped on occasion. In (11), the network performs a dative shift, which actually shouldn't be surprising since the two forms use the same semantic encoding in the trial language. Finally, in one instance, (12), the network either discovers hyperbole or enters an infinite loop.

Although the lack of a morphological system may prevent this model from addressing some of the patterns of production errors in children's speech, some of its problems, such as dropping phrases, and changing tenses, probably reflect common human errors. On the other hand, other characteristics of the network, such as its tendency to perseverate, may be uncommon. It will be interesting to see how actual production training, as opposed to just the prediction training that has been done so far, affects these results.

9 Discussion and Plan

While the connectionist approach to understanding human language has begun to offer novel answers to old questions and may be a diametric alternative to the symbolic tradition, connectionism remains in its infancy. There have been very few connectionist models applied to the problems of learning comprehension or production and these have, for the most part, been confined to simple sentences with very small vocabularies. This is largely due to limitations in our algorithms and hardware and the very reasonable desire to develop minimal, easily analyzable demonstrations of the properties of neural networks. It therefore remains open to reasonable doubt that neural networks, or any general learning method, will ever be able to model human language processing abilities.

The goal of this project is to develop a connectionist model of comprehension and production that is capable of handling languages of reasonable complexity and with the potential of being expanded to the full scale of natural language. This in itself would advance our understanding of the learning abilities of connectionist systems. However, a further goal is to compare the performance of the network to human behavioral data in order to validate it as a reasonable, if very much incomplete, model of the human sentence processing system. Rather than designing a new model for each task or experiment, there is a corroborative advantage to be gained by using a single system capable of performing several tasks in an integrated way to serve as a foundation for multiple experiments. While this is possibly an overly ambitious goal, there is much to be learned in its pursuit. Rather than allowing us to focus on the tasks for which neural networks appear well suited, this endeavor may force us to address their current failings.

Much of the work so far on the project has dealt with the problem of encoding messages in a fixed length vector. While the RAAM is able to learn the task quite well, its encodings may be too complex to serve as useful targets for comprehension. Furthermore, the RAAM involves recursive manipulation of combinatorial representations. A commitment to this ability may lead us to the wrong intuitions about language. The query network seems to offer a reasonable alternative that leads to better comprehension. However, it makes analysis of the network's representations more difficult and may need to be revised to reduce ambiguity. One further area of investigation is the combined use of RAAMs and querying. While this would not alleviate any qualms about using RAAMs, it may make them more effective targets for comprehension.

While the initial results in comprehension and prediction are encouraging, there is still much work to be done to improve these results and to further address the issue of how context should be made available to the network. But the area most in need of further investigation is production. It is unclear what a reasonable schedule for interleaving training on comprehension and production should be. It is also unclear that the model will be able to replicate the patterns of errors children show during development or the relative difficulties adults have with various syntactic structures. Finally, in addressing particular tasks, the language on which the model is trained will be of critical importance. While it would be desirable to use a single language for all experiments that accurately models English, this is simply not possible given the current computational limitations. Therefore, it is likely that several small languages may be required to address phenomena such as patterns of early development and ambiguity resolution.

There are numerous problems with the current model– some relatively easily addressed and others that may be fundamental limitations of the approach. One major problem is that we are still limited to relatively simple languages with small vocabularies. As we create more complex languages, the size of the network relative to this complexity will decrease even further, which may have both qualitative and quantitative effects on performance. Another limitation is that the model does not address the problem of choosing a particular message during production. This will limit its ability to address data from production patterns in children, as message choice and sentence formulation undoubtedly go hand-in-hand when a child attempts to communicate. Is a child who speaks in simple sentences unable to formulate ideas involving multiple propositions or just unable to express them? This is an important question that has not, to my knowledge, received much attention. However, the model simplifies or eliminates many other critical aspects of language, including phonology and morphology, the mechanism of word choice in production, much of the role of associative or world knowledge, a broader situational context, and the mechanisms by which we induce possible messages while learning comprehension.

Nevertheless, this project will be an important step in extending our knowledge of the learning abilities of neural networks as well as contributing to our understanding of the human language system.

9.1 Plan

The following is a rough outline for completion of the thesis:

1. Investigate production training using the current language.

2. Revise the language to include polysemous or vague words, lexical ambiguity (words able to serve as either nouns or verbs), questions and possibly commands, and carefully controlled properties relevant to the center-embedding versus right-branching issue. This will involve an analysis of corpora of spoken English.

3. Revise the semantic tree representations to make active and passive sentences more similar and to subtly distinguish datives from recipients in prepositional phrases.

4. Possibly revise the word input and output representations to encode morphological structure. Inputs to the comprehension network could even use phonological encodings.

5. Improve the current methods for training the message encoders and comprehension and production networks, including investigating the query-RAAM.

6. Test the network's comprehension and production of sentences involving embeddings and possibly attempt to verify those predictions with a study using similar sentences on normal adults.

7. Investigate the ability of the comprehension system to deal with syntactically invalid, but understandable, sentences and with ambiguous and garden-path sentences.

8. Train the network to be bilingual, introducing the second language at varying stages of development, and testing for the "critical period" effect as well

as similarities in performance of late-learning networks and adult second-language learners (Johnson & Newport, 1989).

9. Test the effects of lesions to various parts of the network and determine if the resulting "syndromes" reflect common forms of aphasia. If so, this may lead to a novel characterization of the neurobiology of language.

However, I expect that it would not be inconceivable for me to reduce this list of goals and focus on a few areas that turn out to be particularly interesting or fruitful. I hope to complete the thesis within a year[7] to a year and a half.

References

Allen, R. B. (1987). Several studies on natural language and back-propagation. In *Proceedings of the international conference on neural networks* (pp. II/335–341). San Diego.

Allen, R. B. (1988). Sequential connectionist networks for answering simple questions about a microworld. In *Proceedings of the 10th annual conference of the cognitive science society* (pp. 489–495). Hillsdale, NJ: Erlbaum.

Berg, G. (1992). A connectionist parser with recursive sentence structure and lexical disambiguation. In *Proceedings of the 10th national conference on artificial intelligence* (pp. 32–37). San Jose, CA: AAAI.

Blaubergs, M. S., & Braine, M. D. S. (1974). Short-term memory limitations on decoding self-embedded sentences. *Journal of Experimental Psychology*, *102*(4), 745–748.

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

Charniak, E., & Santos, E. (1987). A connectionist context-free parser which is not context-free, but then it is not really connectionist either. In *Proceedings of the 9th annual conference of the cognitive science society* (pp. 70–77). Hillsdale, NJ: Erlbaum.

Chater, N., & Conkey, P. (1992). Finding linguistic structure with recurrent neural networks. In *Proceedings of the 14th annual conference of the cognitive science society* (pp. 402–407). Hillsdale, NJ: Erlbaum.

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

Chomsky, N. (1975). *Reflections on language.* New York: Pantheon Books.

Chomsky, N. (1976). Conditions on rules of grammar. *Linguistic Analysis*, *2*.

Christiansen, M. H. (1992). The (non)necessity of recursion in natural language processing. In *Proceedings of the 14th annual conference of the cognitive science society* (pp. 665–670). Hillsdale, NJ: Erlbaum.

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

Christiansen, M. H., & Chater, N. (in press-a). Connectionist natural language processing: the state of the art. In M. H. Christiansen, N. Chater, & M. S. Seidenberg (Eds.), *Special issue of cognitive science: Connectionist models of human language processing: Progress and prospects.* Cognitive Science.

Christiansen, M. H., & Chater, N. (in press-b). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*.

Cottrell, G. W. (1985a). Connectionist parsing. In *Proceedings of the 7th annual conference of the cognitive science society* (pp. 201–211). Hillsdale, NJ: Erlbaum.

Cottrell, G. W. (1985b). *A connectionist approach to word sense disambiguation.* Unpublished doctoral dissertation, Department of Computer Science, University of Rochester, Rochester, NY.

Crain, S. (1991). Language acquisition in the absence of experience. *Behavioral and Brain Sciences*, *14*, 597–650.

Dell, G. S. (1986). A spreading activation theory of retrieval in language production. *Psychological Review*, *93*, 283–321.

Dell, G. S., Chang, F., & Griffin, Z. M. (in press). Connectionist models of language production: Lexical access and grammatical encoding. In M. H. Christiansen, N. Chater, & M. S. Seidenberg (Eds.), *Special issue of cognitive science: Connectionist models of human language processing: Progress and prospects.* Cognitive Science.

Dell, G. S., Juliano, C., & Govindjee, A. (1993). Structure and context in language production: A theory of frame constraints in phonological speech errors. *Cognitive Science*, *17*, 149–195.

Diederich, J. (1989). *Spreading activation and connectionist models for natural language processing* (Tech. Rep. No. TR-89-008). Berkeley, CA: International Computer Science Institute.

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

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

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

[7] Ok, so I honestly don't think it will be done in a year but it's good to keep that in the back of my mind lest it become two or three years

Fahlman, S. E., Hinton, G. E., & Sejnowski, T. J. (1983). Massively parallel architectures for ai: Netl, thistle, and boltzmann machines. In *Proceedings of the national conference on artificial intelligence* (pp. 109–113). Washington.

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

Fanty, M. (1994). Context-free parsing in connectionist networks. In G. Adriaens & U. Hahn (Eds.), *Parallel natural language processing* (pp. 211–237). Norwood, NJ: Ablex Publishing.

Fodor, J. A. (1983). *Modularity of mind.* Cambridge, MA: MIT Press.

Francis, W. N., & Kucera, H. (1979). *Manual of information to accompany a standard corpus of present-day edited american english.* Providence, RI: Brown University, Department of Linguistics.

Frazier, L. (1979). *On comprehending sentences: Syntactic parsing strategies.* Bloomington, IN: Indiana University Linguistics Club.

Gasser, M., & Dyer, M. G. (1988). Sequencing in a connectionist model of language processing. In *COLING Budapest: Proceedings of the 12th international conference on computational linguistics* (pp. 185–190). Budapest.

Gasser, M. E. (1988). *A connectionist model of sentence generation in a first and second language.* Unpublished doctoral dissertation, Computer Science Department, University of California, Los Angeles, CA. (UCLS 880050)

Gold, E. M. (1967). Language identification in the limit. *Information and Control, 10*, 447-474.

Goldowsky, B. N., & Newport, E. L. (1993). Modeling the effects of processing limitations on the acquisition of morphology: the less is more hypothesis. In E. Clark (Ed.), *The proceedings of the 24th annual child language research forum* (pp. 124–138). Stanford, CA: Center for the Study of Language and Information.

Hahn, U., & Adriaens, G. (1994). Parallel natural language processing: Background and overview. In G. Adriaens & U. Hahn (Eds.), *Parallel natural language processing* (pp. 1–134). Norwood, NJ: Ablex Publishing.

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 9th annual conference of the cognitive science society* (pp. 106–119). Hillsdale, NJ: Erlbaum.

Harley, T. A. (1993). Phonological activation of semantic competitors during lexical access in speech production. *Language and Cognitive Processes, 8*, 291–309.

Henderson, J. B. (1990). *Structure unification grammar: A unifying framework for investigating natural language* (Tech. Rep. No. MS-CIS-90-94). Philadelphia, PA: University of Pennsylvania.

Henderson, J. B. (1994a). Connectionist syntactic parsing using temporal variable binding. *Journal of Psycholinguistic Research, 23*(5), 353–379.

Henderson, J. B. (1994b). *Description based parsing in a connectionist network.* Unpublished doctoral dissertation, University of Pennsylvania, Philadelphia, PA.

Henderson, J. B. (1996). A connectionist architecture with inherent systematicity. In *Proceedings of the 18th annual conference of the cognitive science society* (pp. 574–579). Hillsdale, NJ: Erlbaum.

Henderson, J. B., & Lane, P. C. R. (1998). A connectionist architecture for learning to parse. In *Proceedings of the 17th international conference on computational linguistics and the 36th annual meeting of the association for computational linguistics (COLING-ACL'98).* University of Montreal, Canada.

Hinton, G. E. (1981). Implementing semantic networks in parallel hardware. In G. E. Hinton & J. A. Anderson (Eds.), *Parallel models of associative memory* (pp. 161–187). Hillsdale, NJ: Erlbaum.

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (1986). Distributed representations. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations* (pp. 77–109). Cambridge, MA: MIT Press.

Howells, T. (1988). Vital, a connectionist parser. In *Proceedings of the 10th annual conference of the cognitive science society* (pp. 18–25). Hillsdale, NJ: Erlbaum.

Huang, X. D., Ariki, Y., & Jack, M. A. (1990). *Hidden markov models for speech recognition.* Edinburgh: Edinburgh University Press.

Jain, A. N., & Waibel, A. H. (1990). Incremental parsing by modular recurrent connectionist networks. In D. Touretzky (Ed.), *Advances in neural information processing systems 2* (pp. 364–371). San Mateo, CA: Morgan Kaufmann.

Johnson, J. S., & Newport, E. L. (1989). Critical period effects in second language learning: The influence of maturational state on the acquisition of english as a second language. *Cognitive Psychology, 21*, 60–99.

Jordan, M. I., & Rumelhart, R. A. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science, 16*, 307–354.

Kalita, J., & Shastri, L. (1987). Generation of simple sentences in english using the connectionist model of computation. In *Proceedings of the 9th annual conference of the cognitive science society* (pp. 555–565). Hillsdale, NJ: Erlbaum.

Kalita, J., & Shastri, L. (1994). A connectionist approach to generation of simple sentences and word choice. In G. Adriaens & U. Hahn (Eds.), *Parallel natural language processing* (pp. 395–420). Norwood, NJ: Ablex Publishing.

Kukich, K. (1987). Where do phrases come from: Some preliminary experiments in connectionist phrase generation.

In G. Kempen (Ed.), *Natural language generation: New results in artificial intelligence, psychology and linguistics* (pp. 405–421). Boston, Dordrecht: Martinus Nijhoff Publishers.

Kwasny, S. C., & Faisal, K. A. (1990). Connectionism and determinism in a syntactic parser. *Connection Science*, *2*, 63–82.

Lane, P. C. R., & Henderson, J. B. (1998). Simple synchrony networks: Learning to parse natural language with temporal synchrony variable binding. In *Icann.* Skövde, Sweden++.

Luce, D. R. (1986). *Response times.* New York: Oxford.

MacWhinney, B. (1999). *The CHILDES database*. Mahwah, NJ: Erlbaum.

Marcus, G. F. (1993). Negative evidence in language acquisition. *Cognition*, *46*, 53–85.

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

Marslen-Wilson, W., & Tyler, L. K. (1980). The temporal structure of spoken language understanding. *Cognition*, *8*, 1–71.

McClelland, J. L. (1989). Parallel distributed processing and role assignment constraints. In Y. Wilks (Ed.), *Theoretical issues in natural language processing* (pp. 64–72). Hillsdale, NJ: Erlbaum.

McClelland, J. L., & Kawamoto, A. H. (1986). Mechanisms of sentence processing: Assigning roles to constituents of sentences. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 2: Psychological and biological models* (pp. 272–325). Cambridge, MA: MIT Press.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, *88*(5), 375-407.

McClelland, J. L., & St. John, M. (1987). *Three short papers on language and connectionism* (Tech. Rep. No. AIP-1). Pittsburgh, PA: Department of Psychology, Carnegie Mellon University.

McClelland, J. L., St. John, M., & Taraban, R. (1989). Sentence comprehension: A parallel distributed processing approach. *Language and Cognitive Processes*, *4*, 287–335.

Medler, D. A. (1998). A brief history of connectionism. *Neural Computing Surveys*, *1*, 61–101.

Miikkulainen, R. (1990a). Script recognition and hierarchical feature maps. *Connection Science*, *2*, 83–101.

Miikkulainen, R. (1990b). A PDP architecture for processing sentences with relative clauses. In *COLING-90: Papers presented to the 13th international conference on computational linguistics* (pp. 3/201–206). Helsinki.

Miikkulainen, R., & Dyer, M. (1990). *Natural language processing with modular neural networks and distributed lexicon* (Tech. Rep. No. CSD-900001). Los Angeles, CA: Computer Science Department, University of California.

Miikkulainen, R., & Dyer, M. G. (1989a). Encoding input/output representations in connectionist cognitive systems. In D. Touretzky, G. Hinton, & T. Sejnowski (Eds.), *Proceedings of the 1988 connectionist models summer school* (pp. 347–356). Los Altos, CA: Morgan Kaufman.

Miikkulainen, R., & Dyer, M. G. (1989b). A modular neural network architecture for sequential paraphrasing of script-based stories. In *Proceedings of the international joint conference on artificial intelligence, ieee* (pp. II/49–56).

Miikkulainen, R., & Dyer, M. G. (1991). Natural language processing with modular pdp networks and distributed lexicon. *Cognitive Science*, *15*, 343–399.

Nakagawa, H., & Mori, T. (1988). A parser based on connectionist model. In *COLING Budapest: Proceedings of the 12th international conference on computational linguistics* (pp. 454–458). Budapest.

Newport, E. L. (1990). Maturational constraints on language learning. *Cognitive Science*, *34*, 11–28.

Newport, E. L., Gleitman, H., & Gleitman, L. R. (1977). Mother, i'd rather do it myself: Some effects and non-effects of maternal speech style. In C. E. Snow & C. A. Ferguson (Eds.), *Talking to children: Language input and acquisition* (pp. 109–149). Cambridge, England: Cambridge University Press.

Noelle, D. C., & Cottrell, G. W. (1995). A connectionist model of instruction following. In *Proceedings of the 17th annual conference of the cognitive science society* (pp. 369–374). Hillsdale, NJ: Erlbaum.

O'Seaghdha, P. G., Dell, G. S., Peterson, R. R., & Juliano, C. (1992). Models of form-related priming in comprehension and production. In R. G. Reilly & N. E. Sharkey (Eds.), *Connectionist approaches to natural language processing* (pp. 373–408). Hillsdale, NJ: Erlbaum.

Pinker, S. (1989). *Learnability and cognition: The acquisition of argument structure.* Cambridge, MA: MIT Press.

Pinker, S. (1990). Language acquisition. In D. N. Osherson & H. Lasnik (Eds.), *Language: An invitation to cognitive science, volume 1* (pp. 199–241). Cambridge, MA: MIT Press.

Plaut, D. C., Seidenberg, M. S., McClelland, J. L., & Patterson, K. (1996). Understanding normal and impaired word reading: computational principles in quasi-regular domains. *Psychological Review*, *103*, 56–115.

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

Rager, J. E. (1992). Self-correcting connectionist parsing. In R. G. Reilly & N. E. Sharkey (Eds.), *Connectionist approaches to natural language processing* (pp. 143–167). Hillsdale, NJ: Erlbaum.

Rohde, D. L. T. (1999a). *LENS: The light, efficient network simulator* (Tech. Rep. No. ??). Pittsburgh, PA: Carnegie Mellon University, Department of Computer Science.

Rohde, D. L. T. (1999b). *The Simple Language Generator: Encoding complex languages with simple grammars* (Tech. Rep. No. CMU-CS-99-123). Pittsburgh, PA: Carnegie Mellon University, Department of Computer Science.

Rohde, D. L. T., & Plaut, D. C. (1997). Simple recurrent networks and natural language: How important is starting small? In *Proceedings of the 19th annual conference of the cognitive science society* (pp. 656–661). Hillsdale, NJ: Erlbaum.

Rohde, D. L. T., & Plaut, D. C. (in press). Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations* (pp. 318–362). Cambridge, MA: MIT Press.

Seidenberg, M. S., & MacDonald, M. C. (in press). A probabilistic constraints approach to language acquisition and processing. In M. H. Christiansen, N. Chater, & M. S. Seidenberg (Eds.), *Special issue of cognitive science: Connectionist models of human language processing: Progress and prospects.* Cognitive Science.

Selman, B., & Hirst, G. (1985). Connectionist parsing. In *Proceedings of the 7th annual conference of the cognitive science society* (pp. 212–221). Hillsdale, NJ: Erlbaum.

Selman, B., & Hirst, G. (1994). Parsing as an energy minimization problem. In G. Adriaens & U. Hahn (Eds.), *Parallel natural language processing* (pp. 238–254). Norwood, NJ: Ablex Publishing.

Sharkey, N. E., & Reilly, R. G. (1992). Connectionist natural language processing. In R. G. Reilly & N. E. Sharkey (Eds.), *Connectionist approaches to natural language processing* (pp. 1–12). Hillsdale, NJ: Erlbaum.

Small, S., Cottrell, G., & Shastri, L. (1982). Toward connectionist parsing. In *Proceedings of the national conference on artificial intelligence* (pp. 247–250). Pittsburgh, PA: AAAI.

Steedman, M. (in press). Connectionist sentence processing in perspective. In M. H. Christiansen, N. Chater, & M. S. Seidenberg (Eds.), *Special issue of cognitive science: Connectionist models of human language processing: Progress and prospects.* Cognitive Science.

St. John, M. F. (1992a). Learning language in the service of a task. In *Proceedings of the 14th annual conference of the cognitive science society* (pp. 271–276). Hillsdale, NJ: Erlbaum.

St. John, M. F. (1992b). The story gestalt: A model of knowledge-intensive processes in text comprehension. *Cognitive Science*, *16*, 271–306.

St. John, M. F., & McClelland, J. L. (1988). Applying contextual constraints in sentence comprehension. In *Proceedings of the 10th annual conference of the cognitive science society* (pp. 26–32). Hillsdale, NJ: Erlbaum.

St. John, M. F., & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, *46*, 217–457.

St. John, M. F., & McClelland, J. L. (1992). Parallel constraint satisfaction as a comprehension mechanism. In R. G. Reilly & N. E. Sharkey (Eds.), *Connectionist approaches to natural language processing* (pp. 97–136). Hillsdale, NJ: Erlbaum.

Tabor, W., Juliano, C., & Tanenhaus, M. K. (1997). Parsing in a dynamical system: An attractor-based account of the interaction of lexical and structural constraints in sentence processing. *Language and Cognitive Processes*, *12*(2/3), 211–271.

Thornton, R. (1999). *Adventures in long-distance moving: The acquisition of complex wh-questions.* Unpublished doctoral dissertation, University of Connecticut, Hartford, CT.

Waltz, D. L., & Pollack, J. B. (1985). Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, *9*, 51–74.

Ward, N. (1991). *A flexible, parallel model of natural language generation.* Unpublished doctoral dissertation, Computer Science Division, Univeristy of California, Berkeley, CA. (UCB/CSD 91/629)

Weber, V., & Wermter, S. (1996). Using hybrid connectionist learning for speech/language analysis. In S. Wermter, E. Riloff, & G. Scheler (Eds.), *Lecture notes in artificial intelligence 1040: Connectionist, statistical, and symbolic approaches to learning for natural language processing* (pp. 87–101). Berlin: Springer-Verlag.

Weckerly, J., & Elman, J. L. (1992). A PDP approach to processing center-embedded sentences. In *Proceedings of the 14th annual conference of the cognitive science society* (pp. 414–419). Hillsdale, NJ: Erlbaum.

Wermter, S., Riloff, E., & Scheler, G. (1996). Learning approaches for natural language processing. In S. Wermter, E. Riloff, & G. Scheler (Eds.), *Lecture notes in artificial intelligence 1040: Connectionist, statistical, and symbolic approaches to learning for natural language processing* (pp. 1–16). Berlin: Springer-Verlag.

Wermter, S., & Weber, V. (1994). Learning fault-tolerant speech parsing with screen. In *Proceedings of the 12th national conference on artificial intelligence* (pp. 670–675). Seattle, WA: AAAI.

Wermter, S., & Weber, V. (1997). SCREEN: Learning a flat syntactic and semantic spoken language analysis using artificial neural networks. *Journal of Artificial Intelligence Research*, *6*, 35–85.

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n^3. *Information and Control*, *10*(2), 189–208.