A Connectionist Parser Aimed at Spoken Language

Ajay Jain **Alex Waibel**

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We describe a connectionist model which learns to parse single sentences from sequential word input. A parse in the connectionist network contains information about role assignment, prepositional attachment, relative clause structure, and subordinate clause structure. The trained network displays several interesting types of behavior. These include predictive ability, tolerance to certain corruptions of input word sequences, and some generalization capability. We report on experiments in which a small number of sentence types have been successfully learned by a network. Work is in progress on a larger database. Application of this type of connectionist model to the area of spoken language processing is discussed.

This research was funded by grants from ATR Interpreting Telephony Research Laboratories and the National Science Foundation under grant number EET-8716324. The views and conclusions contained in this document are the authors' and should not be interpreted as representing the official policies, either expressed or implied, of ATR Interpreting Telephony Research Laboratories, the National Science Foundation, or the U.S. Government.

Introduction

Traditional methods employed in parsing natural language have focused on developing powerful formalisms to represent syntactic and semantic structure along with rules for transforming language into these formalisms. The builders of such systems must accurately anticipate and model all of the language constructs that their systems will encounter. Spoken language, with its weak grammatical structure, complicates matters. We believe that connectionist networks which *learn* to transform input word sequences into meaningful target representations offer advantages in this area.

Much work has been done applying connectionist computational models to various aspects of language understanding. Some researchers have used connectionist networks to implement formal grammar systems for use in syntactic parsing [1, 5, 10, 6]. These networks do not learn their grammars. Other work has focused on semantics [8, 11, 3, 2] but either ignored parsing, or the networks did not *learn* to parse. The networks presented in this paper learn their own "grammar rules" for transforming an input sequence of words into a target representation, and learn to use semantic information to do role assignment.

The remainder of this paper is organized as follows. First, there is a description of our network formalism. Next, we describe in detail a modest experiment in which a network was taught to parse a small class of sentences. We show how the network behaves with some novel sentences and with sentences that have been corrupted as in spoken language. Then, we show how we have generalized our architecture to model a much larger class of sentences and discuss the work as it currently stands. Lastly, we offer some concluding remarks about this work and suggest future directions.

Network Formalism

The most common type of deterministic connectionist network is a back propagation network [9]. Processing units are connected to each other, and each connection has an associated weight. Connections are unidirectional. Units have an activity value and an output value which is usually a sigmoidal function of the activity. For a connection from unit A to unit B, we define the stimulation along the connection to be the output value of unit A multiplied by the weight associated with the connection. A unit's activity is simply the sum of the stimulation along each of its input connections. A network learns input / output mappings by iteratively updating its weight values using a gradient descent technique.

Spoken language is an inherently sequential domain, and standard back propagation is not well suited to such a task. Recently, some recurrent extensions to back propagation where sequences of connections can form cycles have been proposed that can handle sequential input [4, 7]. Our networks extend these notions by explicitly accounting for time in our processing units. Units have activities which decay during each discrete time step by a constant factor. Thus, the activation of a unit can be built up over time from repetitive weak stimulation. Activity values are also damped to prevent unstable behavior. By gently "integrating" activities, the network has time to adapt to new information smoothly.

The activity of a unit is passed through a sigmoid squashing function to produce an output value as in standard back propagation. In addition, a value called the *velocity* is calculated. It is the rate of change of the output of a unit. Each connection in the network has two weights associated with it -- one for the output value and one for the velocity value. The velocity values are important to represent dynamic behavior which depends on changes in activation more than on absolute activation.

In order to facilitate symbolic processing, we use special units, called gating units, which gate the connections between groups of units. Fig. 1 diagrams the behavior of gating units. Slot C represents a particular word. It can be

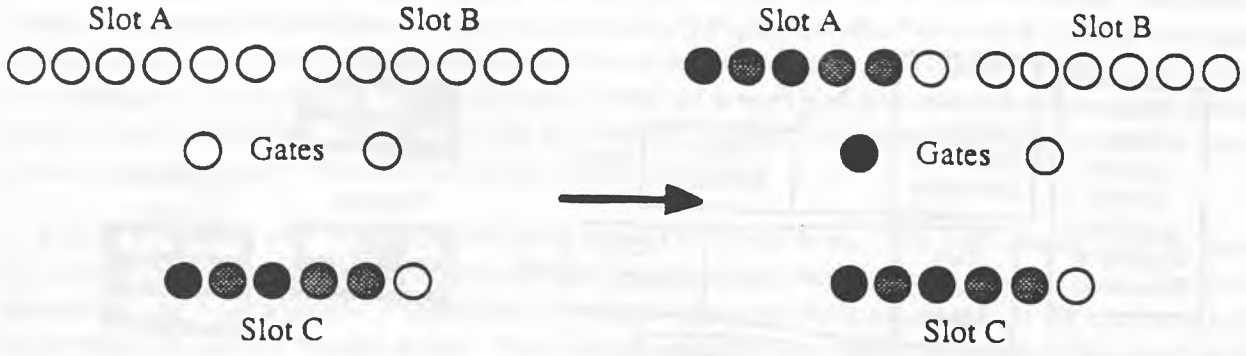


Figure 1: Gating Units

assigned to either slot A or slot B. The connections from the units of Slot C to both Slots A and B are gated by the two units below the slots (the connections are not shown here). In this case, the gating unit for slot A becomes active (see the right hand side of the diagram), and the pattern of activation across slot C becomes active across slot A. This type of assignment behavior can, in principle, be learned by a network without using gating units but is computationally wasteful.

Parsing Sentences

Our domain for this experiment consists of active and passive sentences consisting of up to 3 noun phrases and 2 verb phrases each. There are three roles for nouns to fill for each verb -- agent, patient, and recipient. The network also models subordinate and relative clause structure as well as prepositional attachment. The lexicon consists of 40 words which are divided into 7 classes -- nouns, verbs, adjectives, adverbs, auxiliaries, prepositions, and determiners. Each word is defined at most once within a class, but some words belong to two classes.

Words are represented as patterns of activation across a set of feature units. There are seven sets of feature units, one for each class of words. The pattern for a word consists of two parts: a feature part and an identification part. The feature part contains a small set of binary features encoding semantic information about a word. The identification part serves to disambiguate words which have identical feature parts (like a serial number). This allows one to add words to the lexicon which have the same features as existing words without any re-training of the network (the modifiable connections of the network do not connect to any identification units). Our 40 word lexicon is in a virtual sense much larger than 40 words. Each word is associated with one unit in the network which has hard-wired connections to excite the appropriate pattern across the feature units. A sentence is presented to the network by stimulating the word units corresponding to the words in the sentence each for a short time in sequence.

The target representation for sentences in the network has two levels: the Phrase level and the Structure level. Refer to Fig. 2 for a picture of the network structure. The Phrase level consists of groups of units called blocks, each of which contain a noun or a verb and its modifiers. A noun block has slots for a noun, two adjectives, a preposition, and a determiner. A verb block has slots for a verb, an auxiliary, and an adverb. There are 3 noun blocks and 2 verb blocks. Each block captures a phrase. The blocks are filled in order, with the first noun phrase occupying the first noun block, the second NP occupying the second noun block, and so on. The exact ordering relationship between the verb phrases and the noun phrases is lost in this representation, but due to the simplicity of the sentences this is not a problem.

The units in the Structure level describe the relationships between the phrases in the Phrase level the clauses they make up. There are six relationships possible:

- Agent: Noun block (NB) is agent of Verb block (VB). Group of 3 by 2 units.

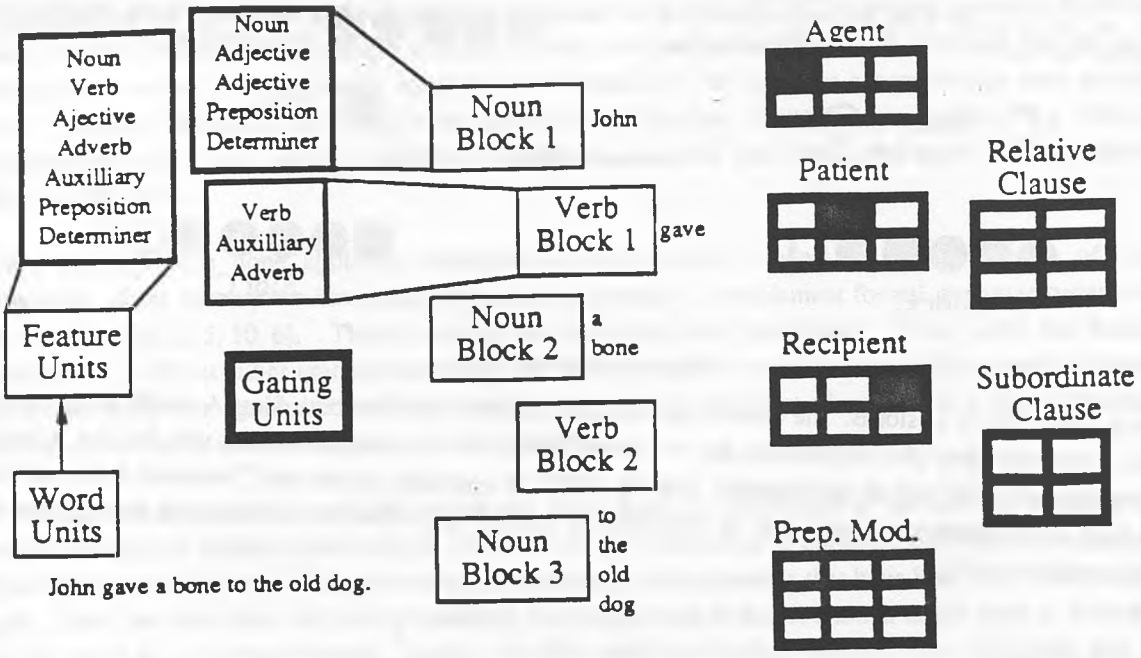


Figure 2: Network Structure

- Patient: NB is patient of VB. Group of 3x2.
- Recipient: NB is recipient of VB. Group of 3x2.
- Prepositional Modification: NB modifies other NB. Group of 3x3.
- Relative Clause: VB modifies NB. Group of 2x3.
- Subordinate Clause: VB subordinate to other VB. Group of 2x2.

The sentence, "John gave a bone to the old dog." is shown in Fig. 2.

In Fig. 2, the units shown in thick lined boxes have modifiable input connections -- they learn their behavior. The gating units at the Phrase level share a group of hidden units. These hidden units have connections from the feature units, the noun and verb blocks, and the gating units themselves. The Phrase level forms a recurrent subnetwork. The representation units of the Structure level also share a set of hidden units. These hidden units "see" all that the other set of hidden units see plus the structure representation units. The Structure level also forms a recurrent subnetwork. None of the hidden units have connections to the identification bit portions of the slots in the network.

The network whose performance we will characterize below was trained in two phases. First, the gating units in the Phrase level which are responsible for the behavior of the slots of the noun and verb blocks were trained. Their behavior is quite complex. They must learn to turn on when a word appears across the feature units for their slot (and their slot is supposed to be filled), stay on until the word disappears (even after the word has been assigned to the slot), turn off sharply, and stay off even when another word appears across their feature units. They must also learn to overwrite or empty out incorrectly assigned slots. Words get assigned incorrectly when they have representations in more than one class and there is insufficient information to disambiguate the usage. The word "was" has representations both as a verb and as an auxiliary verb. The network must assign it to both the auxiliary and the verb slots of the current verb block, and disambiguate the assignment when the next word comes in by either overwriting the verb slot with the real verb or emptying out the auxiliary slot.

The next phase involves adding the Structure level and training the structure representation units. The targets for

the structure units are set at the beginning of a sentence and remain the same for the whole sentence. This forces the units to try to make decisions about sentence structure as early as possible; otherwise, they accumulate error signals. On the surface, it may seem that these units should have more or less monotonic behavior. However, the sentences in our domain do not necessarily contain sufficient information at word presentation time to make accurate decisions about the word's function. This coupled with the network's attempt to make decisions early causes the structure units to have surprisingly complicated activation patterns over time.

A set of 9 sentences was used to train the gating units of the Phrase level. They were selected to be the smallest set of sentences which would cover a reasonably rich set of sentences for training the Structure units. The network generalized very well to include "compositions" of sentence types from the initial set of 9. It was tolerant of varying word speed and silences between words. This is an important property, useful for integration of speech systems with natural language processing.

From this network, the Structure units were added. Eighteen sentences which were correctly processed at the Phrase level were chosen to train the Structure level. A variety of sentences was included. There were more active constructions than passive, more single clause than two clause sentences. Many different role structures were present in the training set. The network learned the set successfully.

Network Performance

The trained network displays several interesting properties on both the sentences in the training set and other new input sentences. A novel sentence is one which is not isomorphic to a training sentence modulo the identification bits of the words in the sentences. Thus, "Peter gave Fido the bone" is not different from "John gave Fido the bone." However, "Peter gave Fido the snake" is different since "snake" is animate, but "bone" is not.

The sentence "A snake ate the girl." is an example of the simplest type from the training set. The behavior of the key structure units corresponding to the roles of verb block 1 are shown in Fig. 3. Each box contains the indicated

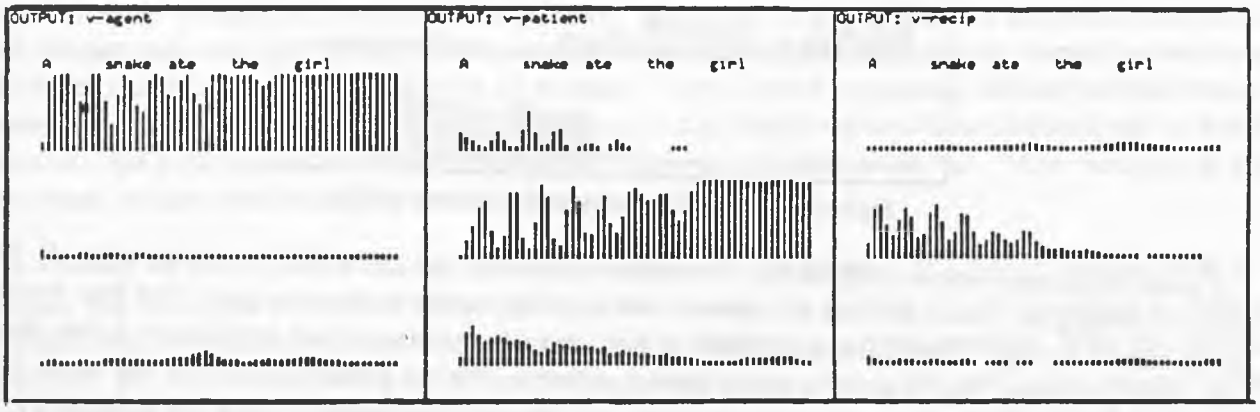


Figure 3: A snake ate the girl.

relationship units. The horizontal axis corresponds to time. Each word is presented for ten time steps. The first row of each box corresponds to the first noun phrase, the second to the second noun phrase and so on. The initial representation shows low activities for all of the relationship units. During presentation of the first word, the agent unit representing the first noun becomes quite active. It has not yet quite decided on its final value however, as can be seen by the oscillations. The other units are all either weakly active or oscillating. When the verb "ate" is presented, the agent unit corresponding to noun 1 fires strongly since it is now clear that the sentence is not a passive construction. Similarly, the patient unit for noun 2 becomes more active since "ate" is transitive. The last part of the sentence further verifies the correct representation. If "near the house" is appended to the sentence (forming a

sentence not in the training set), it gets attached to "the girl".

In spoken language, determiners and other short function words tend to be poorly articulated. This is indeed a persistent problem for speech recognition systems, as it leads to word deletions. Despite such deletions, our network makes appropriate role assignments with such sentences as "Snake ate girl." The role assignment is agent / patient as in the uncorrupted sentence. Non-speech interjections are also possible as in, "A snake (ahh) ate the girl." A speech recognition system could easily interpret the non-speech "ahh" as "a". Our network puts the non-speech "a" in the determiner slot of the second noun block, and then overwrites it with "the". The result is a good parse of the ill-formed sentence. Similarly, simple stuttering does not adversely affect network performance in many cases. It is important to note that this behavior was not taught in any way to the network.

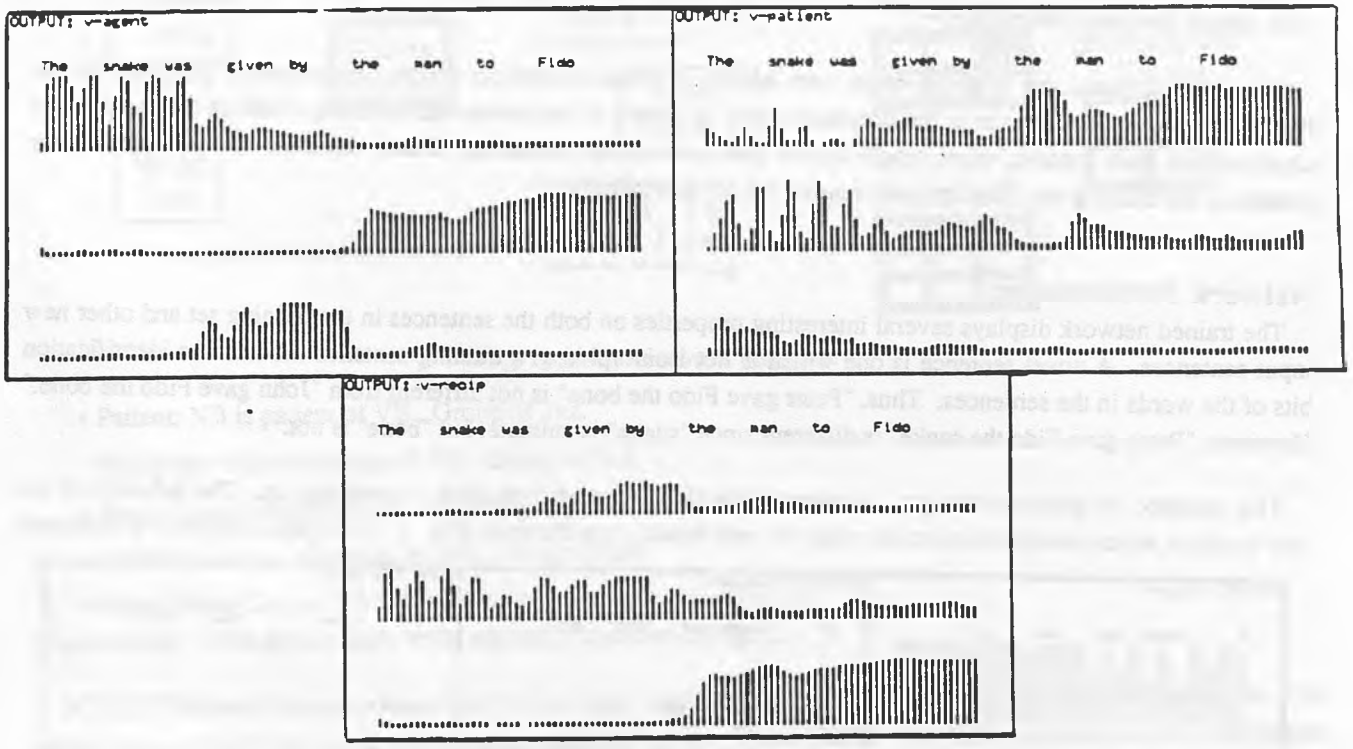


Figure 4: The snake was given by the man to Fido.

A more complicated sentence is given by, "The snake was given by the man to Fido." as shown in Fig. 4. It was not in the training set. There was only one sentence with a similar structure in the training set: "The bone was given by the man to the dog." They differ significantly in that "snake" is animate and less significantly in their detailed noun phrase structure. Fig. 4 shows a similar display as before. For the duration of the first two words of this sentence, the units behave as they did in the previous one. However, the passive construction indicated by "was given" causes the agent unit for the first noun to decay and the agent unit for the third noun to grow. This is because several other passive sentences in the training set were structured where the third noun was the agent. The word "by" causes the agent units to move toward their final positions and indicate "by the man" is the agent block. The recipient and patient units make their final decisions with a little residual oscillation at this time as well. At the arrival of "to Fido" finally, the correct parse is locked up.

In the previous example, the network seized the preposition "by" to make its role assignments. The network is also able to use semantic cues from words in the absence of meaningful function words. Fig. 5 show the network's behavior on the sentence, "A snake was given an apple by John." Here, the network must rely on the semantic features of "snake" and "apple" to make the proper role assignment. Since "snake" is animate, and apple is not, their

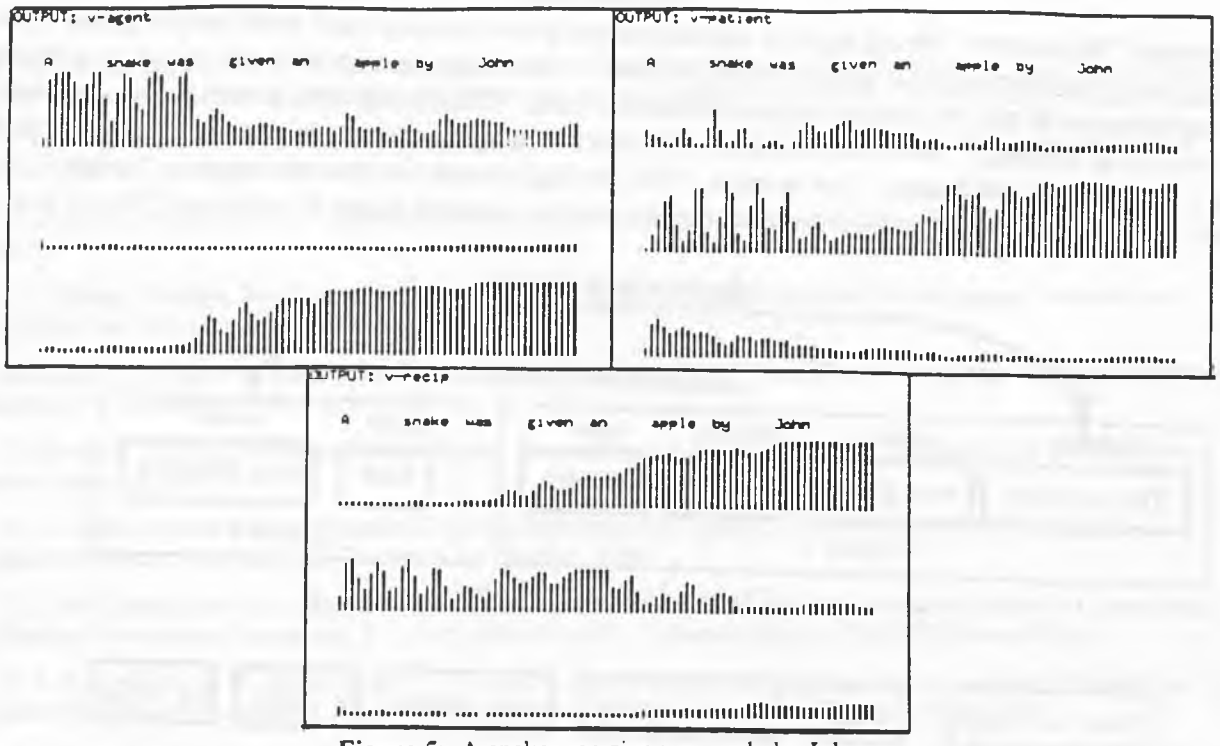


Figure 5: A snake was given an apple by John.

roles are assigned as recipient and patient, respectively. This occurs when "an apple" is processed. The opposite role assignment is made in, "A bone was given the dog by John." The heuristic learned by the network is that inanimate objects are preferred as patients over animate objects.

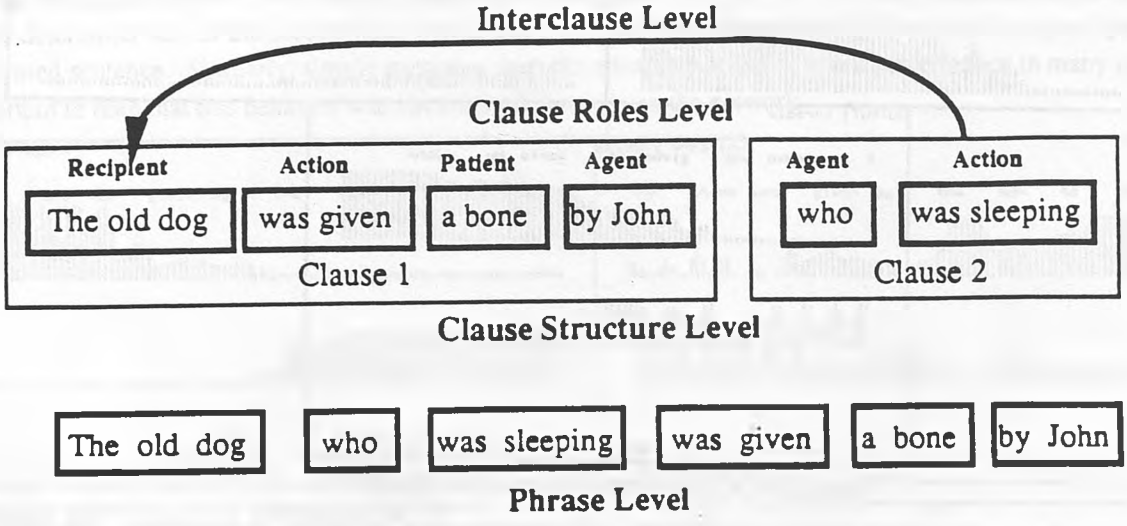
Single clause sentences dominated the training set, but a few two clause sentences were presented to explore the network's ability to learn the interactions among clauses. Since the network architecture allowed for only three noun phrases with two verb phrases, these sentences were quite simple. The network learned to recognize subordinate clauses as in, "John slept after he ate an apple." It also learned to recognize sentence terminal relative clauses as in, "John kissed the girl who slept." Generalization capability in the two clause sentences was not tested extensively due to the paucity of sentences constructible within the constraints of the task. Minor variations in the noun phrase structure from the training sentences were properly treated.

In summary, we have observed four key features in the network's performance. It is able to combine syntactic, semantic, and word order information effectively to perform its task. The network tries to be predictive, making decisions about the structure of the sentence as soon as sufficient information becomes available. When the network is uncertain, the units oscillate among sets of possible future states in a way that is detectable by the network via the velocity weights. The network responds reasonably to sentences which have been modified from those in its training set.

Extending the Architecture

The architecture described above is still limited in its present form. To extend and scale it to more complex sentences and to allow for a more flexible representation, we have designed a more general architecture. The new architecture is modular, hierarchical, and recurrent. It has four levels: Phrase, Clause Structure, Clause Roles, and Interclause. The Phrase level is analogous to that of the network described earlier, but differs in three important ways. The words in the lexicon all share the same feature units instead of being separated into classes. The phrases are not separated into verb and noun blocks; the input sentence is parsed into blocks of contiguous words which

form phrases. The sentence "The old dog who was sleeping was given a bone by John" would be split up into "(The old dog)'(who) (was sleeping) (was given) (a bone) (by John)". The Clause Structure level uses the evolving Phrase level representation to split the sentence into its constituent clauses: "(The old dog) (was given) (a bone) (by John)" and "(who) (was sleeping)". The Clause Roles level does the role assignment and noun phrase attachment for each of the clauses as they are mapped. For example, "(The old dog)" would be called the recipient, "(a bone)" the patient etc. The final level, Interclause, encodes the fact that the embedded clause is relative to "(The old dog)".



"The old dog who was sleeping was given a bone by John."

Figure 6: New Representation

Fig. 6 shows the representation of this sentence.

At the Phrase level and the Clause Roles level, the network consists of horizontally replicated modules which are trained on all of the phrases and clauses from a set of sentences. This artificially creates the effect of a very large training set on a very large network without the cost associated with building such networks. The Clause Structure and Interclause levels cannot be treated in this manner since they deal with whole sentence structure.

We are currently exploring such a network on a set of over 200 sentences. These include sentences with passive constructions, center embedded clauses, and some lexical ambiguity. Preliminary results on the individual modules comprising the network have been encouraging, and we hope to begin testing on the fully integrated network shortly.

Conclusion

We have presented a connectionist architecture which learns to incrementally parse sentences. Our networks exhibit behavior that could potentially be extremely useful for the integration of speech and language processing. Tolerance to corruptions of input including ungrammaticality, word deletions and insertions, and varying word speed are all desirable for speech applications. Connectionist networks appear to be less rigid than more formal systems thereby allowing them to handle a wider variety of sentences given only a limited initial set of examples. Their ability to learn complex dynamical behaviors from diverse knowledge sources makes them well suited for speech processing applications.

References

1. E. Charniak and E. Santos. A Connectionist Context-Free Parser Which is not Context-Free But Then It is not Really Connectionist Either. Proceedings of the Ninth Annual Conference of the Cognitive Science Society, 1987.

2. G. Cottrell. Connectionist Parsing. Proceedings of the Seventh Annual Conference of the Cognitive Science Society, 1985.

3. G. Cottrell. *A Connectionist Approach to Word Sense Disambiguation.* Ph.D. Th., University of Rochester, May 1985.

4. J. L. Elman. Finding Structure in Time. Tech. Rept. 8801, Center for Research in Language, University of California, San Diego, 1988.

5. M. Fanty. Context Free Parsing in Connectionist Networks. Tech. Rept. TR174, Computer Science Department, University of Rochester, November, 1985.

6. T. Howells. VITAL: A Connectionist Parser. Proceedings of the Tenth Annual Conference of the Cognitive Science Society, 1988.

7. M. I. Jordan. Serial Order: A Parallel Distributed Processing Approach. Tech. Rept. 8604, Institute for Cognitive Science, University of California, San Diego, 1986.

8. J. L. McClelland and A. H. Kawamoto. Mechanisms of Sentence Processing: Assigning Roles to Constituents. In *Parallel Distributed Processing*, J. L. McClelland and D. E. Rumelhart, Ed., The MIT Press, 1986.

9. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing*, J. L. McClelland and D. E. Rumelhart, Ed., The MIT Press, 1986.

10. B. Selman and G. Hirst. A Rule-Based Connectionist Parsing System. Proceedings of the Seventh Annual Conference of the Cognitive Science Society, 1985.

11. D. Waltz and J. Pollack. "Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation". *Cognitive Science 9* (1985).