Communicated by Jeffrey Elman

Parsing Complex Sentences with Structured Connectionist Networks

Ajay N. Jain
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA

A modular, recurrent connectionist network is taught to incrementally parse complex sentences. From input presented one word at a time, the network learns to do semantic role assignment, noun phrase attachment, and clause structure recognition, for sentences with both active and passive constructions and center-embedded clauses. The network makes syntactic and semantic predictions at every step. Previous predictions are revised as expectations are confirmed or violated with the arrival of new information. The network induces its own "grammar rules" for dynamically transforming an input sequence of words into a syntactic/semantic interpretation. The network generalizes well and is tolerant of ill-formed inputs.

1 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 complicates matters even further in several ways. It is more strictly sequential than written language (one *cannot* look ahead). Spoken language also has a loose structure that is not easily captured in formal grammar systems. This is compounded by phenomena such as ungrammaticality, stuttering, and interjections. Errors in word recognition are also possible. Independent of these factors, systems that can produce predictive information for speech recognition are desirable. Parsing methodologies designed to cope with these requirements are needed.

Connectionist networks have three main computational strengths that may be useful in such domains. First, they learn and can generalize from examples. This offers a potential solution to the difficult problem of constructing grammars for spoken language. Second, by virtue of the learning algorithms they employ, connectionist networks can potentially exploit statistical regularities across different modalities (e.g., syntactic information and prosodic information). Lastly, connectionist

networks tend to be tolerant of noisy input as is present in real speech. The work presented here is a step toward a connectionist parsing system that demonstrates these benefits in the context of a speech processing system.

Many connectionist architectures have been devised for processing natural language. Several of these architectures have implemented formal syntactic grammar systems (e.g., Charniak and Santos 1987; Selman and Hirst 1985; Fanty 1986). Others have modeled semantic phenomena but have paid less attention to parsing (e.g., Waltz and Pollack 1985; McClelland and Kawamoto 1986). These systems, as with standard formal grammar systems, do not acquire grammar. In contrast, this article describes a connectionist network that *learns* to parse complex sentences presented one word at a time by acquiring a statistical grammar based on a combination of semantic and syntactic cues.[1]

The goals of this work were threefold: first, to show that connectionist networks can learn to incrementally parse nontrivial sentences; second, to show how modularity and structure can be exploited in building complex networks with relatively little training data; and third, to show generalization ability and noise tolerance suggestive of application to more substantial problems.

2 Incremental Parsing

Language processing is particularly difficult for connectionist systems in part because of its sequential nature. As input tokens are received, it is not generally possible to immediately determine how to process them. Complex temporal behavior is required to parse effectively. Earlier work produced a connectionist architecture that learned to parse a small set of sentences, including some with passive constructions (Jain 1989). This article describes an extension to the architecture that processes grammatically complex sentences and requires a substantial scale increase.

A set of sentences with up to three clauses, including sentences with center-embedding and passive constructions, formed the training corpus.[2] Here are some example sentences:

- Fido dug up a bone near the tree in the garden.
- I know the man who John says Mary gave the book.
- The dog who ate the snake was given a bone.

[1] A lengthier presentation of this work appears in Jain and Waibel (1990).

[2] The training set contained over 200 sentences. They were a subset of the sentences that form the example set of a parser based on a left associative grammar developed by Roland Hausser (Hausser 1989). These sentences are grammatically interesting, but they do not reflect the statistical structure of common speech.

```
[Clause 1:  [Phrase Block 1: The dog   (RECIPIENT)]
            [Phrase Block 2: was given (ACTION)]
            [Phrase Block 3: a bone    (PATIENT)]]
[Clause 2:  [Phrase Block 1: who       (AGENT)]
            [Phrase Block 2: ate       (ACTION)]
            [Phrase Block 3: the snake (PATIENT)]
   (RELATIVE: "who" refers to Clause 1, Phrase Block 1)]
```

Figure 1: Parser's representation of, "The dog who ate the snake was give a bone." The sentence is represented as two clauses made up of phrase blocks to which role labels are assigned. The embedded relative clause is also labeled.

Given the input one word at a time, the network's task is to incrementally build a representation of the sentence that includes the following information: phrase block structure,[3] clause structure, semantic role assignment, and interclause relationships. Figure 1 shows a representation of the desired parse of "The dog who ate the snake was given a bone."

3 Network Architecture

The approach to temporal context taken in this work is different from that of the simple recurrent network (Elman 1990) or the time-delay paradigm (Waibel *et al.* 1989). In the former approach, a network must learn to capture complex contextual information through somewhat indirect means. In the latter approach, time is represented spatially, and units have direct access to portions of past history and have no need to learn to capture temporal information. The approach taken here lies somewhere in the middle. Networks are given the computational hardware to use storage buffers that can atomically manipulate activation patterns. The process of capturing temporal context is integrated into the task to be learned.

The network formalism is described in Jain (1989). There are four major features of this formalism:

- Well-behaved symbol buffers are constructed using groups of units whose connections are *gated* by other units.

[3]The term *phrase block* denotes a contiguous sequence of tightly related words. It does not correspond to the classical grammatical notion of *phrase*.

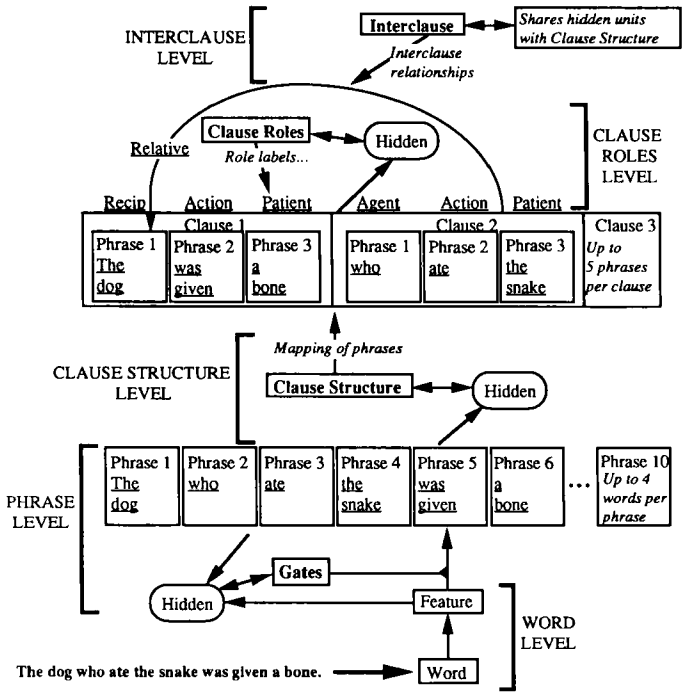

Figure 2: Parsing architecture with an example sentence.

- Units have temporal state; they integrate their inputs over time, and decay toward zero.

- Units produce a standard sigmoidal output value and a velocity output value. Units are responsive to both the static activation values of other units and their dynamic changes.

- The formalism supports recurrent networks.

Networks learn using gradient descent via error backpropagation.

Figure 2 shows the detailed network architecture. Information flows through the network as follows. A word is presented by stimulating its associated word unit for a short time. This produces a pattern of activation across the feature units that represents the meaning of the

word.[4] The Phrase level uses the sequence of word representations from the Word level to build contiguous phrase blocks. Connections from the Word level to the Phrase level are modulated by gating units that learn the required conditional assignment behavior to capture word feature activations patterns in the phrase blocks. The Clause Structure level assembles phrase blocks into clauses. For example, "[The dog] [who] [ate] [the snake] [was given] [a bone]," is mapped into "[The dog] [was given] [a bone]" and "[who] [ate] [the snake]." The Clause Roles level produces labels for the roles and relationships of the phrase blocks in each clause of the sentence (e.g., Agent, Action, and Patient). The final level, Interclause, represents the interrelationships among the clauses making up the sentence (e.g., clause 2 is relative to the first phrase block of clause 1).

The parser was constructed from three separately trained modules. The Phrase level formed one module, the Clause Roles level another, and the Clause Structure and Interclause levels together formed the third. Each module's hidden units received recurrent connections from the output units (those units with specified targets) to provide contextual information (similar to Jordan 1986). The recurrent connections also provided a means for competitive effects to develop among output units.

The Phrase and Clause Roles modules were constructed by replication. A subnetwork capable of assembling a single phrase block was trained to process all the phrase blocks in the corpus and was replicated to produce the 10 phrase blocks making up the Phrase level. Thus, even if a particular construction only appeared in one position in the training set, the full Phrase level module is able to properly process it at any position. Similarly, at the Clause Roles level, a single subnetwork was trained to process all of the clauses in the corpus. This subnetwork was also replicated. The replication process is similar to "weight slaving" in TDNNs (Waibel *et al.* 1989), where equality constraints are placed on weights serving analogous functions in different input positions.

Target values were set at the beginning of pattern presentation for all units with static target values. This encouraged predictive behavior since it was advantageous for units to achieve their target values as early as possible during the presentation of an input pattern to avoid accumulating error. Gating units have changing targets. They must become active during the time course of a single word and then become inactive. Their target values were computed dynamically during the presentation of each training sentence.

[4]The connections from the word units to the feature units, which encode semantic and syntactic information about words, are compiled into the network and are fixed. Connectionist networks have previously been used for acquiring semantic features of words (Miikkulainen and Dyer 1989), but in building large systems, it makes sense to precompile as much lexical information as possible — especially if one does not have a surfeit of training data. By making use of existing lexical knowledge, one can avoid the expense of acquiring such information through training and ensure that the lexicon is uniform and general.

It is important to note that while the parsing architecture is fixed for any particular parser, in principle there are no limits on the number of constituents, or number of labels and relationships that a parser can contain. If the training set contained sentences with conditional clauses, this would simply require an additional set of Interclause units to denote the conditional relationships between clauses. The Clause Structure level would not require additional units, but the existing units would have to learn the clausal structure of conditional sentences. The architecture supports manipulation of symbols, building of structures, and labeling (and attachment) of structures, and is thus quite general.

4 Parsing Performance

The network learned to parse a large, diverse training set. This section discusses three aspects of the network's performance: dynamic behavior of the integrated modules, generalization, and tolerance of noisy input.

4.1 Dynamic Behavior. The dynamic behavior of the network will be illustrated on the example sentence from Figures 1 and 2: "The dog who ate the snake was given a bone." This sentence was not in the training set.

Initially, all of the units in the network are at their resting values. The units of the phrase blocks all have low activation. The word unit corresponding to "the" is stimulated, causing its word feature representation to become active across the feature units of the Word level. The hidden layer causes the gating unit associated with slot 1 of phrase block 1 to become active, which in turn causes the feature representation of "the" to be assigned to the slot. The gate closes as the next word is presented. The remaining words of the sentence are processed similarly, resulting in the final Phrase level representation shown in Figure 1. While this is occurring, the higher levels of the network are processing the evolving Phrase level representation.

The behavior of some of the output units of the Clause Structure level is shown in Figure 3. Early in the presentation of the first word, the Clause Structure level hypothesizes that the first four phrase blocks will belong to the first clause — reflecting the dominance of single clause sentences in the training set. After "the" is processed, this hypothesis is revised. The network then believes that there is an embedded clause of three (possibly four) phrase blocks following the first phrase block. This predictive behavior emerged spontaneously from the training procedure (a large majority of sentences in the training set beginning with a determiner had embedded clauses after the first phrase block). The next two words ("dog who") confirm the network's expectation. The word "ate" allows the network to firmly decide on an embedded clause of three phrase blocks within the main clause. This is the correct clausal

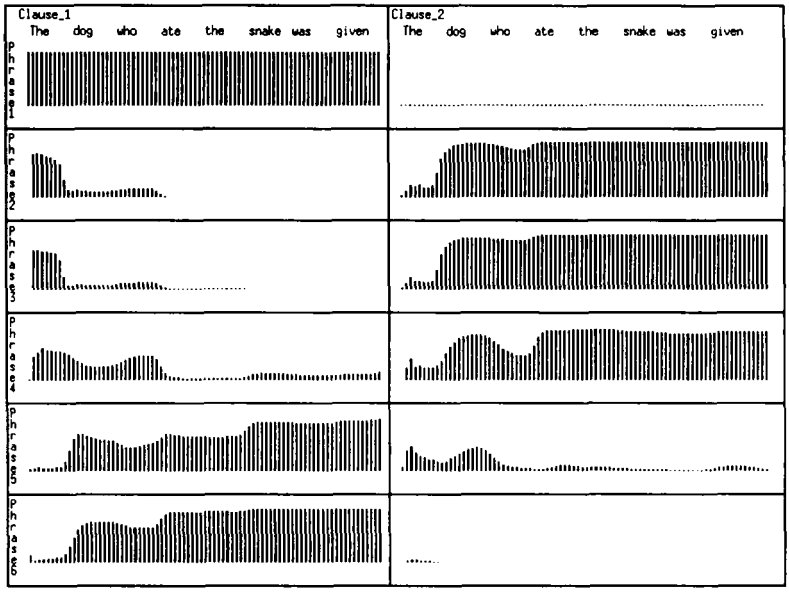

Figure 3: Example of Clause Structure dynamic behavior.

structure of the sentence and is confirmed by the remainder of the input. The Interclause level (not shown in the figure) indicates that the embedded clause is relative to the first phrase block of the main clause during the initial hypothesis of the embedded clause.

The Clause Roles level processes the individual clauses as they are mapped through the Clause Structure level. The output units for clause 1 initially hypothesize an Agent/Action/Patient role structure with some competition from a Recipient/Action/Patient role structure (the Agent and Recipient units' activation traces for clause 1, phrase block 1 are shown in Fig. 4). This prediction occurs because active constructs outnumbered passive ones during training. The final decision about role structure is postponed until just after the embedded clause is presented. The input tokens "was given" immediately cause the Recipient/Action/Patient role structure to dominate. The network also indicates that a fourth phrase block (e.g., "by Mary") is expected to be the Agent (not shown). For clause 2 ("[who] [ate] [the snake]"), an Agent/Action/Patient role structure is again predicted; this time the prediction is borne out.

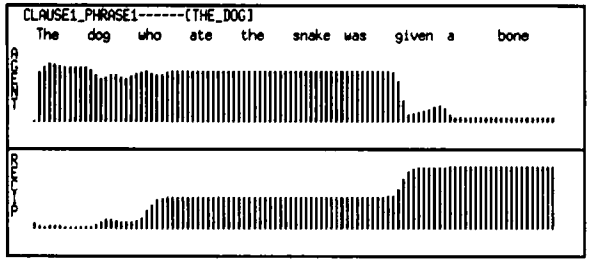

Figure 4: Example of Clause Roles dynamic behavior.

4.2 Generalization and Noise Tolerance. One type of generalization is implicit in the architecture. Word feature patterns have two parts: a syntactic/semantic part and an ID (identity) part. The representations of "John" and "Peter" differ only in their ID parts. Units in the network that learn do not have any input connections from the ID portions of the word units. Thus, when the network learns to parse "John gave the apple to the boy," it will know how to parse "Peter promised the cookie to the girl." This type of generalization is extremely useful, both for addition of new words to the network and for processing sentences for which the net was not explicitly trained.

The network also generalizes to correctly process truly novel sentences — sentences that are distinct (beyond ID features) from those in the training set. The weight sharing techniques at the Phrase and Clause Structure levels have an impact here. Although it is difficult to measure generalization quantitatively, some statements can be made about the types of novel sentences that are correctly processed relative to the training sentences. Substitution of single words resulting in a meaningful sentence is tolerated almost without exception. Substitution of entire phrase blocks by different phrase blocks causes some errors in structural parsing on sentences that have few similar training exemplars. However, the network does quite well on sentences that can be formed from major components of familiar sentences (e.g., interchanging clauses). More training data, especially for multiclause sentences, would improve the performance.

Noise tolerance is particularly important in processing spoken language. The effects of noise were simulated by testing the network on sentences that had been corrupted in several ways. Note that during training the parser was exposed only to well-formed sentences.

Sentences in which verbs were made ungrammatical were processed without difficulty (e.g., "We am happy."). Sentences in which verb phrases

were badly corrupted produced reasonable interpretations. For example, the sentence "Peter was gave a bone to Fido," received an Agent/Action/Patient/Recipient role structure as if "was gave" was supposed to be either "gave" or "has given." Interpretation of corrupted verb phrases was context dependent.

Single clause sentences in which determiners were randomly deleted to simulate speech recognition errors were processed correctly 85% of the time. Multiple clause sentences corrupted in a similar manner produced more parsing errors. There were fewer examples of multiclause sentence types, and this hurt performance. Deletion of function words such as prepositions beginning prepositional phrases produced few errors, but deletions of critical function words such as "to" in infinitival constructions introducing subordinate clauses caused serious problems.

The network was somewhat sensitive to variations in word presentation "speed,"[5] but tolerated interword silences. Interjections of "ahh," which were simulated by inserting "a" in the word sequence, and partial phrase repetitions were also tested. The network did not perform as well on these sentences as other networks trained for less complex parsing tasks. One possibility is that the modular replication technique is preventing the formation of strong attractors for the training sentences. There appears to be a tradeoff between generalization and noise tolerance.

5 Conclusion

This project shows that a connectionist network can acquire a statistical grammar for an interesting fraction of English. It predictively applies its knowledge as input tokens are processed. This differs from attempts to add stochastic components to rule-based grammars (e.g., Seneff 1989). The stochastic component is beneficial for disambiguation and prediction, but in such systems, probabilities are applied at a single level (e.g., along arcs in a transition network). The connectionist approach can model stochastic effects at varying degrees of coarseness: anything from a single word to a complex partially complete syntactic structure can be the (statistically trained) trigger of some action. The training procedure forces the network to "search" efficiently, to apply likely "rules" before less likely ones. To minimize error, the trained network must make decisions about sentence structure as early as possible.

The connectionist approach also offers advantages over conventional parsers in terms of noise tolerance. Ungrammatical near-misses can be processed sensibly in many cases in the connectionist approach whereas grammar-based approaches often include no error correction (Hausser 1989). Other grammar-based approaches rely on complex, handcrafted

[5] *Speed* refers to the number of network update cycles during the presentation of each word. The network was trained on a constant speed.

rules to cope with foreseeable input variations (Young *et al.* 1989). A connectionist parser can potentially be trained to cope with expected input variations, but it will also be tolerant of other variations that were not explicitly modeled.

The modular technique permitted the three component modules of the network to be constructed and trained independently — an important advantage when designing large networks. In addition, replicative procedures that remove positional sensitivities were an efficient way to maximize generalization from an unbalanced training set. However, replication prevented the network from modeling position-specific regularities that may have enhanced noise tolerance. More systematic work needs to be done to understand the effects of the various training procedures on generalization and noise tolerance.

Work is in progress applying this type of network to a spoken language system for a conversational domain with a limited vocabulary. The connectionist approach should prove useful because tight coupling is desired between the parsing system and the underlying speech system. The predictive nature of this type of parser (its outputs can help drive the word hypothesizer of a speech system), its robustness, and the potential to integrate multiple input modalities (e.g., pitch and stress cues) should benefit the system. The suggestive results presented here will be more fully explored in this ongoing work.

Acknowledgments

This research was funded by grants from ATR Interpreting Telephony Research Laboratories, the National Science Foundation under Grant EET-8716324, and the Office of Naval Research under contract number N00014-86-K-0678. I thank Dave Touretzky and Alex Waibel for helpful comments and discussions.

References

Charniak, E., and Santos, E. 1987. A connectionist context-free parser which is not context-free but then it is not really connectionist either. In *Proc. Ninth Ann. Conf. Cog. Sci. Soc.*, 70–77.

Elman, J. L. 1990. Finding structure in time. *Cog. Sci.* **14**(2), 179–212.

Fanty, M. 1986. Context-free parsing with connectionist networks. In *AIP Conference Proceedings number 151*, J. S. Denker, ed. American Institute of Physics, New York.

Hausser, R. 1989. *Computation of Language: An Essay on Syntax, Semantics, and Pragmatics in Natural Man-Machine Communication.* Springer-Verlag, Berlin.

Jain, A. N. 1989. *A Connectionist Architecture for Sequential Symbolic Domains.* Tech. Rep. CMU-CS-89-187, School of Computer Science, Carnegie Mellon University.

Jain, A. N., and Waibel, A. H. 1990. Incremental parsing by modular recurrent connectionist networks. In *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, ed., pp. 364–371. Morgan Kaufmann, San Mateo, CA.

Jordan, M. I. 1986. Attractor dynamics and parallelism in a connectionist sequential machine. In *Proc. Eighth Ann. Conf. Cog. Sci. Soc.*, pp. 531–546.

McClelland, J. L., and Kawamoto, A. H. 1986. Mechanisms of sentence processing: Assigning roles to constituents. In *Parallel Distributed Processing*, Vol. 2, J. L. McClelland and D. E. Rumelhart, eds., pp. 273–331. The MIT Press, Cambridge, MA.

Miikkulainen, R., and Dyer, M. G. 1989. Encoding input/output representations in connectionist cognitive systems. In *Proceedings of the 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski, eds., pp. 347–356. Morgan Kaufmann, San Mateo, CA.

Selman, B., and Hirst, G. 1985. A rule-based connectionist parsing system. In *Proc. Seventh Annu. Conf. Cog. Sci. Soc.*, 212–221.

Seneff, S. 1989. TINA: A probabilistic syntactic parser for speech understanding systems. In *Proc. 1989 IEEE Conf. Acoustics, Speech Signal Process.*, pp. 711–714.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. 1989. Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoustics, Speech, Signal Process.* **37**(3), 328–339.

Waltz, D., and Pollack, J. 1985. Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cog. Sci.* **9**, 51–74.

Young, S. R., Hauptmann, A. G., Ward, W. H., Smith, E. T., and Werner, P. 1989. High level knowledge sources in usable speech recognition systems. *Commun. ACM* **32**(2), 183–193.

Received 14 May 1990; accepted 23 October 90.