Report Number: WUCS-89-33

1989-08-01

CDP: A Connectionist Deterministic Parser A Dissertation Proposal

Kanaan A. Faisal

We propose to address problems related to the problem of Natural Language Processing. Neural networks approach will be used in attaching these problems. In this proposal a number of problems are considered: parsing of ill-formed sentences in addition to well-formed sentences, resolutions of some lexical ambiguities using syntactic context, and parsing of sentences sequentially over an input stream which is unbounded in length. Two network training approaches will be examined. Deductive training, based on grammar rules, and inductive training, based on sentence processing will be fully investigated. Comparisons with each approach to its corresponding symbolic approach will be made.

... Read complete abstract on page 2.

CDP: A Connectionist Deterministic Parser A Dissertation Proposal

Kanaan A. Faisal

Complete Abstract:

We propose to address problems related to the problem of Natural Language Processing. Neural networks approach will be used in attaching these problems. In this proposal a number of problems are considered: parsing of ill-formed sentences in addition to well-formed sentences, resolutions of some lexical ambiguities using syntactic context, and parsing of sentences sequentially over an input stream which is unbounded in length. Two network training approaches will be examined. Deductive training, based on grammar rules, and inductive training, based on sentence processing will be fully investigated. Comparisons with each approach to its corresponding symbolic approach will be made.

CDP: A CONNECTIONIST DETERMINISTIC PARSER
A Dissertation Proposal

Kanaan A. Faisal

WUCS-89-33

August, 1989

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899

ABSTRACT

We propose to address problems related to the problem of Natural Language Processing. Neural networks approach will be used in attacking these problems. In this proposal a number of problems are considered: parsing of ill-formed sentences in addition to well-formed sentences, resolution of some lexical ambiguities using syntactic context, and parsing of sentences sequentially over an input stream which is unbounded in length. Two network training approaches will be examined. Deductive training, based on grammar rules, and inductive training, based on sentence processing will be fully investigated. Comparisons with each approach to its corresponding symbolic approach will be made.

CDP:

A CONNECTIONIST

DETERMINISTIC PARSER

A DISSERTATION PROPOSAL

Kanaan A. Faisal

August 2, 1989

WUCS-89-33

1. Introduction

The ultimate problem in Natural Language Processing (NLP) is to find a processor which has the same facility with language as humans have. This means that such a processor must not impose any artificial limits on sentences or words. Unfortunately, parsing of NL is not a solved problem. Although there have been many attempts to find such a processor, none of them have fully succeeded.

Of course, the proposed system does not claim to be capable of solving the ultimate problem of NLP, but it will offer better performance in the problem areas addressed than existing symbolic systems do. The problem areas to be addressed are: parsing of ill-formed sentences in addition to well-formed sentences, resolving some lexical ambiguities using syntactic context, and parsing sequentially over an input stream which is unbounded in length.

Our view is consistent with that of McClelland and Kawamoto. They observe,

'..that sentence processing is an on-line process, a process that unfolds in real time as each word is heard' (McClelland and Kawamoto, 1986, p.316).

We believe *the most promising approach is one based on human sentence processing. Determinism is central to that approach. NLP is thus a deterministic process, one that does not backtrack but uses its current information to choose correct interpretations as the structure unfolds.*

Any model of language processing should permit alternative linguistic structures to compete while inputs are processed. While computer models based on backtracking may capture the competitive nature of sentence processing they do not simulate it in the manner McClelland and Kawamoto observed above. Furthermore, there is no evidence from human experiments that any conscious re-processing of inputs is routinely performed, except perhaps for ''garden path'' sentences. Thus, competition among structures must be addressed through other means.

A good example of competition can be found in the TRACE model of speech perception (McClelland and Elman, 1986). In that work, competing interpretations of the mock-speech feature vectors are proposed and activation levels rise or fall as each potential interpretation is supported or contradicted. Parsers should permit syntax and other levels of processing to aid in resolving lexical ambiguities just as ambiguous phonemes were resolved in TRACE.

The proposed research is aimed at developing a computational model of language processing which is consistent with a subsymbolic view of cognition as well as the notion of determinism. The model is based upon the Determinism Hypothesis proposed by Mitchell Marcus:

> *'The syntax of any natural language can be parsed by a mechanism which operates "strictly deterministically" in that it does not simulate a nondeterministic machine'* (Marcus, 1980, p. 2).

This hypothesis makes explicit the idea that natural language processing need not depend in any fundamental way on the use of backtracking. The consequences of this hypothesis are being explored here through the development of parsing techniques which integrate work on symbolic, deterministic parsing with current work in neural networks.

The result will be a model we call the **Connectionist Deterministic Parser** (CDP). It features a parser which parses sequentially over an input stream which is unbounded in length, parses slightly ill-formed sentences in addition to well-formed sentences, and resolves some lexical ambiguities using syntactic context.

The primary results of this thesis will be to establish and explore the capabilities of a connectionist model (CDP) to process natural language deterministically with all the capabilities discussed above.

2. Background on Connectionist NLP

Recently several researchers have proposed NLP formalisms based on Neural Networks (NN). Waltz and Pollack (1985) and Cottrell (1985b) give models for word-sense and syntactic disambiguation. A central aspect of these models is that they process the different sources of knowledge used in NLP, such as lexical and world knowledge in a highly integrated way. That is, syntactic and semantic processing is combined. Fanty (1985) implements the Cocke-Younger-Kasami algorithm for parsing limited-length strings for context-free grammars in a bounded number of parallel computation steps which is a function of the maximum length of input strings. An arbitrary (15 word) limit is imposed on the lengths of input sentences. Selman and Hirst (1985), built a model that tries to simulate the rules of a context free grammar using an updating rule similar to the one used in the Boltzmann machine (Hinton and Sejnowski, 1986) and apply simulated annealing.

In most neural networks or connectionist parsers, grammar rules are processed into a network of units connected with excitatory and inhibitory links. The number of units required to realize a given grammar is a function of the maximum input sentence length and the complexity of the grammar. Hence, a limitation is introduced on the number of elements that can be present in the input streams. Sentences are processed within such a framework by presenting them, possibly in a simulated left-to-right fashion, at the input side of the network and activations are permitted to spread through the network (Cottrell, 1985a; Fanty, 1985; Waltz and Pollack, 1985). Alternatively, a stochastic method, such as simulated annealing, is used (Selman and Hirst, 1985).

All of these attempts share many of the same advantages and disadvantages, but none of the models allow "unbounded streams" to be processed iteratively.

Classically, parsers process inputs iteratively from an unbounded stream of input. Neural network parsers proposed to date do not work iteratively and often have limits imposed artificially on the length of a sentence. There is, however, work underway on neural network iteration mechanisms that could be used in neural network parsers of natural language. For examples of work on the mechanisms of iteration in neural networks see Williams and Zipser, (1988) or Servan-Schreiber, Cleeremans, and McClelland, (1988).

3. Background on Deterministic Parsing

There are several parsing frameworks which depend on search techniques to find possible interpretations of a sentence: Augmented Transition Networks (ATN, Woods, 1970), the chart-based parser (Kay, 1985) and the Definite Clause Grammar (DCG, Periera and Warren, 1980) all depend in a fundamental way on the use of backtracking. To many, such search processes do not correspond well with their intuitions about human processing of language. In particular, human parsing seems close to a deterministic process: a process that does not search alternatives but rather uses the information it has at the time to choose the correct interpretation. This section examines deterministic parsing and other computational models based on deterministic parsing.

Deterministic, or ''wait-and-see'' parsing (WASP)[1] requires that several (3 to 5) constituents of the input sentence be in view before deciding on the appropriate structure for the current constituent. Once this decision has been reached, it cannot be reversed and once structures have been constructed, they are never thrown away. Deterministic parsers are also rule-based in that their actions are controlled by a collection of rules. To aid in conflict resolution, the rules are partitioned into rule packets.

A single processing step in a deterministic parser consists of selecting a rule to be fired from an active rule packet and firing the rule to alter the structure and positions of constituents in the model. As with most rule-based systems, rules whose left-hand sides are found to match the state of the system are eligible to be fired. Rule packets are activated as a consequence of which portion of the structure is being built and, within the packet, conflicts are resolved through pre-assigned rule priorities and from the static ordering of rules within each priority value. Once selected, the rule is fired and its actions are performed. The action effects changes on the stack and buffer. After a series of processing steps, a termination rule fires, and the final parse structure is left on the top of the stack.

Deterministic parsing has provided convincing evidence that most sentences can be deterministically parsed with a stack, a buffer of sentence constituents, and partitioned packets of rules.

Several researchers have found rule-based, deterministic systems valuable in exploring various aspects of language processing. PARAGRAM (Charniak, 1983) considers all rules in parallel and "scores" each test performed on the left-hand side of a rule according to predefined weights. The rule with the best score is fired. Thus, some rule always fires and parsing never fails due to a collapse in rules. This provides a context in which to explore ungrammaticality, among other phenomena. On the other hand,

---

[1] Waltz and Pollack (1985) characterize this option as one based on ''delay'' as opposed to one based on backtracking.

ROBIE (Milne, 1986) is a deterministic parser that is able to resolve lexical ambiguities. It resolves only local ambiguities (ambiguities that can resolved within the sentence). ROBIE resolves word sense ambiguity without violating the strictness of the determinism assumption.

Other work on Deterministic parsing that relates to this research includes and LPARSIFAL (Berwick,1985) which attempts to learn PARSIFAL rules from examples of *positive evidence* (i.e. grammatical sentences). This parser starts with a small set of rules and grows up gradually by building new rules from experience of positive examples. So, in effect, the system is inductively learning the grammar rules from sentence examples.

4. Advantages of Neural Networks

Neural networks provide an alternative to the rule-based approach. Connectionist models of parsing exhibit more robustness, are easier to modify, and generalize better than rule-based parsers. With these advantages, connectionist parsing must be considered an important alternative to classical, symbolic approaches.

Networks implement soft-constraint satisfaction. Networks degrade gracefully when presented with input cases that are similar to, but not precisely like the training cases. The knowledge required in such tasks is not attached to a single unit in the network, but distributed across many units. Graceful degradation occurs as a natural by-product of a network model.

Our Neural Network (NN) approach leads to improvements over conventional, purely symbolic approaches. Generalization, competition, and learning capabilities of NN and the absence of backtracking in deterministic parsing are the most important reasons to consider merging deterministic parsing and NN approaches for attacking such NLP problems. This is true particularly since training is based on sample inputs and their individual processing steps with no facility for retracting previously built structure. The collection of rule packets is replaced with a single connectionist network. Training sets are derived either from existing rule-based grammars or from traces of sentence processing.

We will investigate whether a neural network degrades in ways advantageous to parsing novel structures (e.g., ungrammatical, and lexical ambiguous structures). Connectionist networks support soft-constraint satisfaction which make them well-suited for this type of task. Traditionally, this goal has been pursued through the addition of mechanisms such as relaxation techniques (Kwasny and Sondheimer, 1981) or meta-rules (Weischedel and Sondheimer, 1983) which are specifically designed to deal with those problems.

Encouraging results were found in a prototype which is based on a medium size grammar. Further description of the prototype system is available (Kwasny and Faisal, 1989a).

5. Backpropagation Network

A backpropagation network is used in this model (Rumelhart, et al., 1986). It is a feedforward model and uses supervised learning methods. Backpropagation networks are always hierarchical. They always consist of at least three layers of processing units: an input layer, output layer and one or more hidden layers. Ordinarily, not more than two hidden layers are needed. The network used here on all experiments consists of only one hidden layer. It is constructed so that each layer is fully connected to the next layer in feedforward manner. In other words, every unit in the input layer will send its output to every unit in the hidden layer, and every unit in the hidden layer will send its output to every unit in the output layer. There are no connections among the units in a given layer. Thus the units are connected to all units in the layer after them but not to any units within their own layer. This connectivity is only one way to build a backpropagation network. Backpropagation refers to a general learning rule, not to a specific architecture.

This learning procedure involves presenting the network with a set of pairs of input and output patterns. The system first uses the input pattern to produce its own output pattern and then compares this with the correct output pattern. If there is no difference, no learning takes place. Otherwise the weights are changed to reduce the difference using *the generalized delta learning rule*. This rule for changing weights following presentation of an input/output pattern is given by

$$\Delta_p \; w_{ji} = \eta \; (\; t_{pj} \; - \; o_{pj} \;) \; i_{pi} = \eta \; \delta_{pj} \; i_{pi}$$

Where t_{pj} is the teaching input for the *j*th component of the output pattern for pattern *p*, o_{pj} is the *j*th of the actual output pattern produced by the presentation of the input pattern *p*, i_{pi} is the value of the *i*th element of the input pattern, $\delta_{pj} = t_{pj} - o_{pj}$, and $\Delta_p \; w_{ji}$ is the change to be made to the weight from the *i*th to *j*th unit following presentation of pattern p.

The process of training the network is fairly straightforward. Before beginning, the weights in the network are randomized, except those weights from the outside world to the input layer. These simply are fixed at +1.

Then, the network is presented with repeated sets of input patterns. For each input pattern, the weights on the interconnections are adjusted using the generalized delta rule. After doing this many times, watching the network's output from the output layer, its performance improves so it will eventually be able to correctly generate the desired output pattern for each input pattern presented to it.

To explain briefly the operation of the network during training, and how the weights are changed, a pattern to the network will be presented and the resulting activation will be allowed to flow from the input layer through to the output layer. Then, the resulting output pattern is compared to the desired pattern and, beginning with output layer, adjust the weights layer by layer, propagating each layer's error back to the previous layer and computing weight changes according to the generalized delta rule. The weights actually are not changed until after the error is propagated back to the previous layer. Once the activation has flowed forward through the network and the error has flowed backward through the network, this iteration of the network is complete and the network is ready to be presented with the next pattern. This process is repeated many times until the system is trained.

6. Proposed Research

6.1. Objectives

Deterministic parsing promises to never backtrack. Neural network technology promises generalization, competition, and learning capabilities among other aspects. In a small prototype system (Kwasny and Faisal, 1989a) these ideas are continuing to be investigated and encouraging results are being found. I propose to continue conducting experiments along the same line as the prototype system and improve these capabilities by merging these two ideas and investigating them by building a natural language parsing system (CDP) that combines the best features of both deterministic parsing and neural networks. The result of this work will be a deterministic parser that learns, generalizes, supports competition among potential sentence structures, and performs sequentially over an unbounded input stream. In addition to parsing well-formed sentences, the parser will be capable of parsing some types of ill-formed sentences and resolving some lexical ambiguities using syntactic context. The network will be able to parse sentences sequentially.

The main goals of this research are:

(1) To provide a mechanism that can parse input sentences and produce a correct parse structure regardless of other capabilities the parser may have. *Although other aspects of language processing such as a semantic component and the lexicon are of great importance, they are not part of this model.*

(2) To study various grammar learning and training strategies. Two strategies for grammar learning will be used in the model. One is based on rules in a rule-based grammar and the other is based on sentences and their parse structures. The former is inductive and the latter is

deductive learning. Also, two types of training will be pursued: "deductive" training in which the training sequence is derived from the rules of a deterministic rule-based grammar, and "inductive" training in which the training sequence is derived from the states of sentence processing.

(3) To evaluate the generalization capability of the model. The model should be robust and parse ungrammatical sentences in addition to grammatical sentences, and also be able to resolve some lexical ambiguities using local syntactic context.

4) To define the notion of "competition".

(5) To achieve all these goals without introducing any artificial limits on sentences or words. The model should be able to parse input sentences of any length and perform sequentially over an input stream.

6.2. Impact

Aside from the enormous advantages to be attained through the achievement of language processing capabilities approaching those of humans, language is of particular importance in the ongoing debate between connectionists and those advocating purely symbolic approaches to cognition. As Pinker and Prince note,

> *'language has been the domain most demanding of articulated symbol structures governed by rules and principles and it is also the domain where such structures have been explored in greatest depth and sophistication, within a range of theoretical frameworks and architectures, attaining a wide variety of significant empirical results'* (Pinker and Prince, 1988, p. 78).

Thus it is of great interest to determine whether a connectionist language processing system can be developed which rivals symbolic systems in scope. As envisioned, the CDP model once completed will constitute a connectionist language processor, accommodating many aspects of deterministic parsers to date.

At the same time the proposed model will have the following improvements:

(1) It will unify several deterministic parsing systems in one system that is capable of handling many different aspects of language processing. These capabilities range from parsing grammatical sentences (PARSIFAL, Marcus, 1980), to parsing ungrammatical sentences (PARAGRAM, Charniak, 1983), to resolving some lexical ambiguity (ROBIE, Milne, 1986), to learning new rules from examples (LPARSIFAL, Berwick, 1985).

(2) Unlike other models which impose a limit on the length of the input sentences, the CDP will impose no artificial limit on the length of input sentences. Parsers which are limited to sentences of a prespecified length (e.g., Fanty, 1985) suffer some drawbacks. Most importantly, to expand such a parser, one needs not only to add more units, one also needs to program these units with connections that allow them to do the job they need to do. In addition, the size grows rather quickly with the allowable length.

(3) Classically, parsers process inputs iteratively from an unbounded stream of input. Neural network parsers typically do not work iteratively. CDP will work iteratively on the input stream. It will provide one method of processing an input stream the traditional way (iteratively) and still retain its connectionist advantages.

7. The Proposed Method of Solution

The method that will be used here is to train an adaptive neural network using rule templates derived from the rules of a deterministic parser. The performance of the network is measured in the course of parsing various types of sentences. Grammaticality is defined in a relative sense:

grammatical sentences are ones for which the rules of a particular deterministic grammar apply to find a parse. Novel (ungrammatical, ill-formed, or lexically ambiguous) sentences violate the rules in some way and thus no rule-based parse can be found. Processing in the system is symbolic in building structures and sub-symbolic in rule-following behavior. The result is a deterministic parser that learns, generalizes, and supports competition among structures and lexical interpretations.

7.1. Learning a Rule-Based Grammar

A deterministic parser applies rules to a stack and buffer of constituents to generate and perform actions on those structures. Its primary feature is that it does not backtrack, but proceeds forward in its processing never building structures which are later tossed away.

Training of the neural network proceeds by presenting patterns to the network and teaching it to respond with an appropriate action using backward error propagation. The input patterns represent encodings of the buffer positions and the top of the stack from the deterministic parser. The output of the network contains a series of units representing actions to be performed during processing and judged in a winner-take-all fashion. The training data are derived as "rule templates" from rules in a deterministic grammar. These rule templates are instantiated once in each epoch of training. Once the network is trained, the weights can be stored in a file so that various experiments can be performed with the network.

A variety of sentence forms were processed by the prototype system. For test purposes, several sentences were coded that would parse correctly by the rules of the deterministic parser. Also, several mildly ungrammatical sentences were coded to determine if the network was generalizing in any useful way. Finally, sentences containing lexically ambiguous items were coded as ambiguous sets of features. The objective is to test if the syntactic context could aid in resolving such ambiguities. The grammar used is capable of processing a variety of simple sentence forms which always end with a final punctuation mark. Simple declarative sentences, yes-no questions, imperative sentences, and simple passives are permitted by the grammar. What the model actually sees as input is not the raw sentence but a canonical representations of each word in the sentence in a form that could be produced by a simple lexicon. Such a lexicon is not part of the model in its present form.

A medium size grammar has been implemented and used for all experiments in the small prototype system. Each grammar rule is coded as a training template which is a list of feature values. Each template represents many training patterns. On each training epoch every template is instantiated once yielding a specific training case. Thus, each training epoch is slightly different. Further details are available in Kwasny and Faisal (1989a, 1989b, 1989c).

7.2. Performance

Several experiments were conducted using the small prototype. Grammatical and ungrammatical sentences were examined as well as cases of lexical ambiguity. Each example receives a score representing the overall average strength of responses during processing.

The main experiment consisted of testing a group of sentences derived from the types of sentences permitted in the grammar. For example, the following grammatical sentences were processed by the system just as the rule-based grammar would process them.

(1) John should have scheduled the meeting.
(2) They can(v) fish(np).

Each example shows a high average strength value, indicating that the rules used in training have been learned. During parsing, the input sentence is presented in the input buffer from left to right. On each iteration, the network is presented with three constituents in the input buffer. The action specified by the network is performed and the buffer and stack are updated as required. New input items replace empty buffer positions as needed. The process then repeats until a stop action is performed, usually when the buffer becomes empty.

An important test of the system's generalization capabilities is its response to novel sentences. This is strictly dependent upon its experience since no relaxation rules were added to the original grammar to handle such cases. This experiment consists of testing a few ungrammatical sentences that are close to the training data and within the scope of the encoding. For example the following ungrammatical sentence results in a reasonable structure within the system:

(3) *John have should scheduled the meeting.

Overall average strength is lower for ungrammatical sentences compared to similar grammatical ones.

In a final set of experiments, the parser was tested for its ability to aid in the resolution of lexical ambiguity. Normal sentences were presented, except that selected words were coded ambiguously to represent an ambiguously stored word from the lexicon. consider the following sentences :

(4) They <can> fish.
(5) They can <fish>.

In the case shown, the lexically ambiguous words (in brackets) were correctly interpreted and reasonable structures resulted. It was noted that the overall average strengths were lower than comparable grammatical sentences discussed, as expected.

8. Plan of the Research

The previous sections have outlined in detail ideas about CDP. Research will continue in the direction of the prototype system (Kwasny and Faisal, 1989a, 1989b) and Kwasny (1988a, 1988b). CDP will be able to process natural language deterministically, overcome many of the limits of symbolic and rule-based systems, and subsume the advantages of many deterministic rule based-parsers. It will be capable of parsing sequentially over an input stream which is unbounded in length, supporting competition among potential sentence structures, parsing ill-formed sentences in addition to well-formed sentences, and resolving some lexical ambiguities using syntactic context.

The planned research work involves conducting many experiments using a connectionist network simulator to study and investigate the possibility of CDP model that offers better performance in the above problems and can illustrate some of these good capabilities listed above. Simulations thus far have been done using VICE, a simple backpropagation model.

Of course, all aspects of language processing will not be dealt with in the framework of this dissertation. A complete language system is far too complex to be handled in this work. *The work will start with the assumption that the lexicon is an input to the system but not part of the model. Only the syntax of language (parsing) will be considered, although certainly some aspects of other components will have to be considered.*

This work will progress according to the following plan and directions:

(1) Investigating the possibility of improving CDP's ability to offer better performance in the problem areas mentioned above.

(2) Characterizing the limits of generalization. We want define boundary of the generalization capability gained in CDP. We want to know what it can do and what it can not do.

(3) Comparing the Deductive-based CDP with the Inductive-based CDP.

(4) Comparing the two CDP models to other related previous work.

(5) Additional plans for Deductive approach. As my understanding of the capabilities of this approach increases, I will scale up to a much larger

grammar of English. The target grammar is a "mature" grammar that can handle a wide variety of constructions of English (the grammar created by Marcus, given in appendix D (Marcus 1980)). This is a mature grammar in my opinion, though it is by no means a complete set of rules to syntactically analyze all of English. Such a grammar must parse different varieties of sentences from simple sentences, imperative sentences, yes-no-question, WH-questions, passives, and WH-clauses to parsing embedded sentences. Also, I am expanding the coding of the grammar to include a more complete set of features, for example, person and number as well as other labels, and attachments that appear in final structures.

(6) Additional plans for Inductive approach. Since learning is such an integral part of this approach, a comparison to LPARSIFAL will be done. LPARSIFAL (Berwick, 1985) has the Marcus grammar as the target for learning. By the time Berwick's system has processed several hundred sentences, it has acquired approximately 70% of the parsing rules originally hand-written for the Marcus parser. In this system, it is expected to acquire more than 70% of the behavior and capabilities of the Marcus grammar after a few hundred sentence traces.

(7) Studying the feasibility of a better stack representation. My choice of encoding is based on its simplicity and directness. The current system involves duplication of local representations into a fixed-length set of "pools". The parse stack is represented in a localist manner using a fixed-length vector that represents the current state of the parse tree. A fixed-length representation for the stack and tree (recursive structures) has some limitations. We are examining some ideas from Pollack (1988) with the hope of improving the representation of the stack being built and used, and will report on those findings. Recursive Auto-Associative Memory (RAAM) is proposed for such representation. I want to investigate its applicability on CDP and try to use its representation.

(8) Studying the feasibility of a fully connectionistic parser. We believe the work here will progress this goal. The output layer should produce an updated encoding of the input and no external symbolic action will be required. Also, we are examining some of the recent work on recurrent networks with the hope of improving the iteration properties of the system.

9. Evaluating the Model

Any natural language processing model must pass a set of evaluation criteria. Evaluation is a difficult issue, but the following criteria are proposed.

First, in the deductive learning based approach, the system will work on all grammatical sentences tested, and actually learn to behave as the target grammar rules (i.e Appendix D of Marcus 1980 Grammar). This will be demonstrated by comparing it to PARSIFAL's parsing capabilities and the results will be reported.

Second, in the inductive learning based approach, the system will be compared with LPARSIFAL and the results will be reported. The system is expected to learn most of the target grammar rules from sentence traces and perform comparably well to LPARSIFAL.

Third is the system's generalization capabilities and the issue of ill-formedness plausibility. The system must acquire enough knowledge to generalize and parse some ungrammatical sentences. How well does it work on such sentences? One reference standard is PARSIFAL and the extension to PARAGRAM. CDP will be tested on some reasonable examples from both systems. PARSIFAL just gives up and stops without trying to parse such sentences. PARAGRAM parses a few such sentences but still produces nonsensical structures for some sentences. CDP is expected to parse more of these ungrammatical sentences and produce some reasonable structures which PARAGRAM fails to parse.

A fourth evaluation criteria centers on the ability of the system to resolve some lexical ambiguities using local syntactic context. One reference standard is ROBIE which is also based on PARSIFAL. CDP will be tested on some reasonable examples from ROBIE and the results will be reported. CDP is expected to perform comparably well as ROBIE.

A fifth criterion is the ability of the system to iteratively process an unlimited stream of input of variable length. CDP processes all sentences iteratively and therefore evaluation of success is based on the success of all processing above.

10. Summary

Once all the experiments are completed the CDP will include the following features:

(1) It will extend Deterministic Parsing into a more robust model.

(2) It will unify several Deterministic Systems into one systems (PARSIFAL, PARAGRAM, LPARSIFAL, ROBIE).

(3) It will combine Symbolic/Subsymbolic approaches.

(4) It will move closer to a fully connectionist parser.

11. Acknowledgments

I express gratitude to my advisor Dr. Stan Kwasny for his guidance, advise on defining my research topic and his constant help, and to Dr. William Ball for his constant assistance. Also, my gratitude to Steve Cousins, and David Harker, for thoughtful discussions and comments concerning this work. I, of course, am responsible for errors.

References

Berwick, Robert C. (1985). The Acquisition of Syntactic Knowledge. MIT press, Cambridge, MA.

Charniak, E. (1983). A Parser with Something for Everyone. In Margaret King (Ed.), *Parsing Natural Language*, 117-149. (Academic Press: New York).

Cottrell, G.W. (1985a). Connectionist Parsing. *Proceedings of the 7th Annual Conference of the Cognitive Science Society*, Irvine, CA, 201-211.

Cottrell, G.W. (1985b). A Connectionist Approach to Word Sense Disambiguation. Phd Dissertation, available as: *Technical Report 154*, Computer Science Department, University of Rochester.

Fanty, M. (1985). Context-Free Parsing in Connectionist Networks. *Technical Report 174*, Computer Science Department, University of Rochester.

Hinton, G .E, and Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machine. In D.E. Rumelhart, J., L. McClelland and the PDP Research Group (Eds.), *Parallel Distributed Processing I: Foundations*, MIT Press, Cambridge, MA.

Kay, M. (1985). Parsing in functional unification grammar. In D.R. Dowty, L. Karttunen and A. Zwicky (eds), 251-278. *Natural Language Parsing*. New York: Cambridge U. Press.

Kwasny, S.C. (1988a). A PDP Approach to Deterministic Natural Language Parsing. *Neural Networks*, Vol.1, Supplement 1, 305.

Kwasny, S.C. (1988b). A Parallel Distributed Approach to Parsing Natural Language Deterministically. *Technical Report WUCS-88-21*, Department of Computer Science, Washington University, St. Louis, Missouri.

Kwasny, S.C., and N.K. Sondheimer (1981). Relaxation Techniques for Parsing Ill-Formed Input. *American Journal of Computational Linguistics*, 7(2): 99-108.

Kwasny, S.C, and Faisal, K.A. (1989a). Competition and Learning in a Connectionist Deterministic Parser. *Proceeding of the Eleventh Annual Conference of the Cognitive Science Society, Ann Arbor*.

Kwasny, S.C, and Faisal, K.A. (1989b). Determinism and Connectionism in a Natural Language Parsing System. *Technical Report WUCS-89-31*, Department of Computer Science, Washington University, St. Louis, Missouri.

Kwasny, S.C, and Faisal, K.A. (1989c). Deterministic Parsing of Novel and Lexically Ambiguous Sentences. (Forthcoming talk). *IJCAI-89 Workshop: Symbolic Problem Solving in Noisy, Novel, and Uncertain Task Environments,* Eleventh International Joint Conference on Artificial Intelligence, Detroit.

Marcus, M. P. (1980). *A Theory of Syntactic Recognition for Natural Language.* (MIT Press; Cambridge, MA).

McClelland, J.L., and Kawamoto, A H. (1986). Mechanism of sentence processing: Assigning roles to constituents of sentence. In D.E. Rumelhart, J., L. McClelland and the PDP Research Group (Eds.), *Parallel Distributed Processing II: Psychological and biological models*, MIT Press, Cambridge, MA.

McClelland, J.L., and J.L. Elman (1986). The TRACE Model of Speech Perception. *Cognitive Psychology*, 18, 1-86.

McClelland, J. L., and Rumelhart, D. E. (1988). *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises*, MIT Press, Cambridge, MA.

Milne, Robert (1986). Resolving Lexical Ambiguity in a Deterministic Parser. *Computational Linguistics* 12, 1, January-March.

Pereira, F.C.N and D.H.D. Warren. (1980) Definite clause grammars for language analysis--A survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence 13, 231-278*

Pinker, S., & Prince, A (1988). On language and connectionism: Analysis of a parallel distributed processing of language acquisition. *Cognition, 28*, 73-194.

Pollack, Jordan (1988). Recursive Auto-Associative Memory: Devising Compositional Distributed Representations. Report MCCS-88-124, New Mexico State University.

Rumelhart, D. E., G. Hinton, and R.J. Williams (1986). Learning Internal Representations by Error Propagation. In Rumelhart, David E., & James L. McClelland (Eds). *Parallel Distributed Processing I: Foundations*, MIT Press, Cambridge, MA.

Selman, B. & G. Hirst (1985). A Rule-Based Connectionist Parsing System. *Proceedings of the 7th Annual Conference of the Cognitive Science Society*, Irvine, CA, 212-221.

Servan-Schreiber, David, Axel Cleeremans, & James L. McClelland (1988). Encoding Sequential Structure in Simple Recurrent Networks. Report CMU-CS-88-183, Carnegie Mellon University.

Waltz, D.L. & J.B. Pollack (1985). Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation. *Cognitive Science 9*, 51-74.

Weischedel, R.M. and N.K. Sondheimer (1983). Meta-Rules as a Basis for Processing Ill-Formed Input. *American Journal of Computational Linguistics*, 9(3-4): 161-177.

Williams, R. J., and D. Zipser (1988). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. ICS Report 8805, University of California, San Diego.

Woods, W.A. (1970). Transition network grammars for natural language analysis. *Communication of the ACM 13, 591-606.*