Figure 5: Merging of sub-networks

Figure 4: Partial sub-networks extracted from grammar rules

Figure 3: Processing doubled input words

Figure 2: Processing of the sentence 'She goes to the ball' in a predetermined context

Figure 1: Processing of the sentence 'Susi wears a dress'

[JW90a] A. N. Jain and A. H. Waibel. Incremental parsing by modular recurrent connectionist networks. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*. Morgan Kaufmann, San Mateo, CA, 1990.

[JW90b] A. N. Jain and A. H. Waibel. Robust connectionist parsing of spoken language. In *ICASSP-90*, 1990.

[Kem87] C. Kemke. Modelling neural networks by means of networks of finite automata. In *Proc. IEEE - First Ann. Int. Conf. on Neural Networks*, San Diego, CA, 1987.

[Kem91] C. Kemke. Neural network modelling by means of networks of finite automata. In A. Prieto, editor, *Artificial Neural Networks*, pages 48–53. 1991.

[KK93] C. Kemke and H. Kóne. INCOPA - An incremental connectionist parser. In *Proceedings World Congress on Neural Networks*, volume 3, pages 41–44, 1993.

[Kon93] H. Koné. INKOPA - Ein inkrementeller konnektionistischer Parser für natürliche Sprache. Diplom-Arbeit, Fachbereich Informatik, Universität des Saarlandes, 1993.

[KS93] C. Kemke and C. Schommer. PAPADEUS - Parallel parsing of ambiguous sentences. In *Proceedings World Congress on Neural Networks*, volume 3, pages 79–82, 1993.

[Sch87] H. Schnelle. The challenge of concrete linguistic description: Connectionism, massively parallel distributed processing, net-linguistics. Technical Report GENET-24, Sprachwissenschaftliches Institut, Ruhr-Universität Bochum, Bochum, F.R.G., 1987.

[Sch93] C. Schommer. PAPADEUS - Ein inkrementeller konnektionistischer Parser mit einer parallelen Disambiguierungskomponente. Diplom-Arbeit, Fachbereich Informatik, Universität des Saarlandes, 1993.

[WP85] D. L. Waltz and J.B. Pollack. Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, 9(1):51–74, 1985.

8 Conclusions

In this paper we suggested a hybrid approach to parsing natural language sentences which combines advantages of traditional parsing techniques and neural networks.

We introduced a parsing method which is based on incremental construction of the parse tree and, in parallel, of the parsing network.

In short, we presented the system PAPADEUS for lexical and syntactical disambiguation, and the systems INKAS and INKOPA which are able to process a restricted kind of ungrammatical input. All three systems are able to construct a parse tree incrementally. They are based on networks of finite automata and are implemented in Allegro Common Lisp.

References

[Boo87] L. A. Bookman. A microfeature based scheme for modelling semantics. In *Proc. of the 10 th IJCAI*, pages 611–614, Milan, Italy, 1987.

[Boo88] L. A. Bookman. A connectionist scheme for modelling context. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Connectionist Models Summer School.* Morgan Kaufmann, Palo Alto, 1988.

[Cot89] G. W. Cottrell. *A Connectionist Approach to Word Sense Disambiguation.* Morgan Kaufmann, 1989.

[Fan85] M. Fanty. Context-free parsing in connectionist networks. Technical Report TR 174, Computer Science Department , The University of Rochester, Rochester, NY, 1985.

[Hen94] J. Henderson. Description based parsing in a connectionist network. Dissertation, University of Pennsylvania, 1994.

[Jai92] A. N. Jain. Generalization performance in PARSEC - a structured connectionist parsing architecture. In R. P. Lippman J. E. Moody, S. J. Hanson, editor, *Advances in Neural Information Processing Systems 4.* Morgan Kaufmann, 1992.

[JM88] M. F. St. John and J. L. McClelland. Applying contextual constraints in sentence comprehension. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Connectionist Models Summer School.* Morgan Kaufmann, Palo Alto, 1988.

[JW89] A. N. Jain and A. H. Waibel. A connectionist parser aimed at spoken language. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1989.

in the resulting networks (fig.5 f) and g)) is on, so the partial sub-networks are not connected further.

The only problem in transforming the grammar rules into networks of finite state machines is to determine the weights of the connections between the nodes. When we have a given rule, the weights between the father node and the child nodes are determined by the number of items on the right hand side of the rule. If this is n, the weight between a child node and the father node is given by $1/n$. This is quite simple but it works.

Note that several partial derivation trees might be existing at the same time. They hypothesize the ultimate parsing tree. The hypothesized partial derivation trees are maintained until the input of the sentence determines which network is representing the actual derivation tree. According to the grammar this is a deterministic process.

7 Future Work

The PAPADEUS, INKAS, and INKOPA systems were developed in order to provide natural language processing in combination with a speech understanding system. Therefore, we have to take care of some phenomena occuring in spoken natural language dialogues, for example extra words, doubled words, and words which are not known in the lexicon. The last item is very important to speech processing systems since we have to cope with distorted input, names, etc. Thus, we would like to enrich our systems with the ability to process unknown words. It should be possible for PAPADEUS and INKOPA to hypothesize the syntactic category of the given unknown word and to proceed with parsing.

Also the systems should be in the future connected to a real speech understanding system in order to do speech processing from the signal level to the level of semantic representation of the input sentence. Possible application are question answering systems of all kinds.

One problem for the PAPADEUS system is the construction of the microfeature descriptions of words in the semantic space. These descriptions are up to now implemented by the designer of the system which is an awful work to do and the results are not based on experiments or statistics etc. The designer has to set up all microfeatures and the weighted connections to the words on the word level. But the PAPADEUS system is able to learn semantic relations between words. We implemented a Hebbian learning of semantic relations, where the connections between a given word and its semantic represention by microfeatures, between words and word senses, between word senses and word senses, between word senses and contexts, and between contexts and contexts are increased, if they happen to be activated at the same time. Thus, we can incrementally build up the semantic space by providing the system with useful test sentences. This feature is already implemented but not yet tested.

given (man). The first word is connected with the succeeding word to form a new partial derivation tree while the second partial derivation tree is inhibited because of the double appearance of the NP, the second one of which could not be connected further. If one node cannot be connected further in building a parse tree, it will be inhibited. Thus, any appearance of a terminal or non-terminal is subject to inhibition, if it cannot be integrated into the parsing network in order to build up a correct parse tree. This inhibition between nodes happens whenever the appearance of the nodes are not according to the grammar. Thus, this process also works for doubled nouns etc. But note that, for example, lists of adjectives are processed correctly as they are allowed by the grammar rules.

6 Automatic Construction of Parsing Networks

INKOPA is able to transfer a given CFG into a parsing network. Therefore, the grammatical rules given in the CFG are transformed into partial neural networks. The neurons in this network are finite state machines which all have the same parameters (threshold, resting state etc.). The state set is the interval between the minimal value of the neural activity and the threshold; the resting state lies within this interval. The actual input is added to the current state, if the neuron is influenced positively or negatively. If the input is zero, the neuron attempts to reach the resting state.

The process of transforming the grammar rules into partial subnetworks is the same as described above. The left hand side of the rule becomes the father node of the partial sub-network, the items of the right hand side are represented by the children in the sub-network. Whenever a rule is addressed by a lexical item occuring in the input sentence, a partial sub-network is activated. In addition to activating a sub-network, several sub-networks might be merged into a greater sub-network representing the actual derivation tree.

The processes of activating sub-networks and merging of sub-networks can be seen in fig.4 and 5. The input in this case are a determiner and a noun, for example 'Der Mann' (The man). These input words cause activation of the sub-networks shown in fig.4. All rules or sub-networks matching on the right hand side with one of the categories of the input words are activated; we distinguish three different kinds of activation states: *on* means an activation value between 0.9 and 1.0, *active* is a state between 0.5 and 0.9, and *off* are all neurons with an activation under 0.5.

In the next phase the sub-networks are merged, if possible. The result of the merging process for the sub-networks of fig.4 is shown in fig.5. In fig.5 e) you can see that the sub-network with NP2 and NP is fully activated. This causes a further rule to be activated, namely the rule S $\rightarrow$ NP VP, which again is merged with the partial Derivation tree shown in fig.5 e). The result, an activation of the sentence neuron, is shown in fig.5 h). The other partial sub-networks containing an adjective (see fig.4 c)) and a realtive clause (see fig.4 d)) are merged but not every neuron measurement. The distance between the actual microfeature vector representing the semantics of the sentence and the stored microfeature vectors representing the different word senses is measured. The vector, which is closest to the actual microfeature vector, is chosen to represent the meaning of the ambiguous word in question. This symbolic computational process is used very seldomly.

The described disambiguation processes always come up with a solution, thus, the system never provides two interpretations for one word. However, sometimes it might be better to leave the final decision to some other components or systems, or to the user.

It also might be possible that the word meaning is restricted by the preceding sentence, the processing of which also determines the actual context in the context layer. When, for example, the sentence 'Susi goes to the ball' is given, the word 'ball' is ambiguous (dance event or sports item). If the preceeding sentence was 'Susi wears a dress', the context invoked by this sentence (HUMAN - FAMILY - SOCIETY) determines the word 'ball' to mean dance event. The syntactic analysis of the two sentences is shown in fig.1 and 2.[4]

The given context is especially useful for syntactical disambiguation. If you consider the sentence 'Susi sees Peter with the telescope', the parsing tree becomes ambiguous when the phrase 'with the telescope' is read. But the context determined by the word 'telescope' (SEE - PERCEIVE - ACT) determines the phrase 'with the telescope' to belong to the see-event and not to Peter, also because the word sense of telescope is 'device-for-seeing' and there is a strong relation between 'see' and 'telescope'. Another case is the sentence 'Susi sees Peter with the ball'. Then, the word 'ball' turns out to be a sports item, and the prepositional phrase 'with the ball' is attached to Peter because the phrase 'with the ball' cannot be attached to the see-event because there is no semantic relation between 'see' and 'ball' in the context of the preposition 'with'. But there is a semantic relation between 'person' and 'ball' where 'ball' is meant to be a sports kit. Thus, the phrase 'with the ball' is attached to Peter.

5 Processing of Ungrammatical Sentences

The systems INKAS and INKOPA are able to process a restricted kind of ungrammatical input, i.e. sentences with doubled words. This phenomenon often happens in spontaneous dialogues.

Processing doubled input words is very easy in these neural network parsers. When a word is given twice, for example in the sentence 'Der der Mann geht' (The the man goes) (see fig.3), the first appearance of the word 'The' is matched against a lexical rule to determine the category (determiner). Then, a further structure is hypothesized, namely noun phrase. Then the second appearance of 'the' happens and the same process takes place. Now, the succeeding word of the sentence is

[4]The output has been translated into english.

- the grammar rules (about 25).

The connections between the items in these distinguished knowledge sources are also stored in different files within the knowledge base. There are the weighted connections

- between words and microfeatures,
- between microfeatures and word senses,
- between word senses and word senses,
- between word senses and contexts, and
- between contexts and contexts.

Also stored are

- three-place connections between word senses, prepositions, and microfeatures.

They are used for structural disambiguation, for example in the sentence 'Susi sieht Peter mit dem Teleskop' (Susi sees Peter with the telescope) which has several derivation trees depending on the binding of the prepositional phrase.

All items and connections between them have been set up by the designers of the PAPADEUS system. They are not relying on any statistical, experimental, or theoretical basis, except some work on extracting knowledge bases from lexica.

Words, word senses, microfeatures, and contexts are regarded in the PAPADEUS system as neurons. The process involved in disambiguation is a kind of spreading activation where the activation function is a sigmoidal function $((1 - exp(x)/(1 + exp(x)))$, and a winner-take-all network is used for disambiguation on the word level.

In order to do word sense disambiguation, PAPADEUS first finds all possible meanings of a word which are contained in the lexical layer of the system. Then it uses the microfeature representation of the different word meanings for disambiguation. Microfeatures are, for example, adjectives like strong - weak, silent - loud, large - small, and colors (red - green - blue - white - black), and nouns denoting primitive concepts like time concepts (second - minute - hour - day - week - month - year) , species like bird, fish, mammal, flower, plant, or materials. Each word sense is described by several microfeatures, which on the other hand influence the word sense positively or negatively (excitatory or inhibitory connections). A word sense is not only described by its own set of microfeatures but also by the microfeatures representing the words in its context. Therefore, the different word meanings are competing in a winner-take-all network; the word meaning which is most positively influenced by the microfeatures and contexts wins the competition. Thus, the interpretation of a word is determined by its own semantics, the meaning of the words in its surroundings, and the context nodes.

If this semantic matching doesn't come up with a single solution, i.e. the different word senses couldn't be disambiguated, another process takes place: distance already parsed a noun phrase, we extend the network with the sentence or start node which is the father node of the NP and VP node. The resulting subnetwork is a directed graph where the nodes correspond to formal neurons and the arcs represent (weighted) connections which determine the flow of activity within the network. After parsing this, the network hypothesizes a sentence structure and expects a verb phrase. If a verb phrase is parsed next, activation is passed along the connection between the child and the father node, and the sentence node is activated. Also, the sub-networks representing the partial parsing trees are merged, i.e. the sub-network of the VP is connected to the sub-network of the NP by the rule S→ NP VP.

Thus, the parsing is truly incremental, constructs partial parsing trees as far as the input is processed, and hypothesizes expected structures.

If we want to have a free word order, there is no problem because the network doesn't distinguish the sequence inherent in the grammatical rule. But if we need the sequential information too, we have to do something extra. When a rule is matched which has more then one item on the right hand side, we need an additional link which prevents the father node to become active if the child nodes are not coming in the correct order. We called this link 'precondition link' and it is known from processing ordered sequences in neural networks.

4 Disambiguation in Neural Network Parsing

The PAPADEUS system was developed in order to provide lexical and syntactical disambiguation of ambiguous words and sentences. It determines the most plausible word meaning and parse tree based on semantic and context knowledge. This knowledge is provided by several distinguished knowledge sources, in particular the description of word meanings by microfeatures in the semantic space and the representation of contexts as primitive nodes in the context space.

PAPADEUS provides five knowledge sources containing[3]

- about 50 words, some of which are ambiguous like 'ball', but also determiners, adjectives, verbs, and other nouns, which are not ambiguous,
- 31 word senses representing in particular the different word senses for the ambiguous words in the lexicon, like 'dance evening' and 'sports kit' for the word 'ball', or the senses 'dismissal' and 'movement' for the word 'go',
- 11 contexts, for example the contexts LEISURE - FUN or HUMAN - FAMILY - SOCIETY,
- 102 microfeatures describing the words and word senses, for example 'human', 'animal', 'plant', 'big', 'small', 'animate', 'inanimate', and

[3]PAPADEUS disambiguates German sentences, thus the words etc. within the knowledge base are German words, although we describe the items in this paper in English.

parser in neural networks without gaining any advantages out of the parallel reimplementation.

On the other hand, we found approaches that are more closely related to neural networks, for example work of Jain and Waibel [Jai92] [JW90b] [JW90a] [JW89], Waltz and Pollack [WP85], and also Cottrell [Cot89], Bookman [Boo87], [Boo88], and St. John and McClelland [JM88]. The work of Jain and Waibel is based on recurrent neural networks which are divided into several levels of representation, i.e. a word level, where words are described by features, the syntax level which represents syntactical units, and the structure level representing semantical relations between the phrases. Also the networks developed by Waltz and Pollack are divided into several layers of representation, i.e. the syntax layer, the lexical layer, and the context layer.

The development of PAPADEUS, which does lexical and syntactical disambiguation, is mostly influenced by the work of Waltz and Pollack; INKAS and INKOPA are closely related to the approach of Jain and Waibel.[2] Although the results achieved by the connectionist parsers we developed are similar to the work already done, the processing and the implementation are a bit different from known approaches.

The basis for the three parsing systems are networks of finite automata ([Kem87], [Kem91]), which are suitable and well-defined structures for representing neural networks.

The parsing in these systems proceeds incrementally; the network is built up according to the construction of the parse tree. Thus, we don't have problems with variable binding nor with representing recursive structures. The parsing just follows the grammar rules like a traditional parser, but in our systems (except for INKOPA, see below) the grammar rules are represented by partial sub-networks. For each grammar rule there is a corresponding sub-network where the father node represents the left hand side of the rule and the child nodes represent the right hand side of the rule. In the following we will also speak of rules, when these subnetworks are meant.

The construction of the network proceeds as follows: The input sentence is read word by word. When a new word is read, the lexical rules are tested in order to see to which lexical item the word belongs. In this way we cope with morphologically different appearances of one word. Next, the input word is matched against the grammatical rule base in order to find out to which grammatical category the word belongs.

If the category is determined, the network proceeds with the parsing process. Iteratively, the grammatical rules are matched to the yet determined non-terminals. If a rule matches, the network is extended with the corresponding sub-network representing the grammatical rule where the father node of the rule is represented by the successor node and the child nodes are represented by the preceding nodes of the resulting network. If we have, for example, the rule S $\rightarrow$ NP VP and we have

[2] For more details on PAPADEUS, INKAS, and INKOPA see [Sch93], [KS93], [Kon93], and [KK93].

usual in spontaneous natural language dialogues. There are, for example, additional words which don't appear in the grammar, such as 'hm' or 'oops'. Also there might be some words doubled as in 'The - hm - the man went to the market'.

Another problem is the ambiguity on the word and the sentence levels, referred to as lexical and syntactical ambiguity. Examples are the word 'bank' and the sentence 'Mary saw Peter with the telescope' where the phrase 'with the telescope' can belong to the seeing of Mary, or to Peter, who may possess the telescope. These problems can only be solved with large amounts of world knowledge, which are not easy to handle and to incorporate into the parsing process. Thus, traditional parsers first need to build up all possible meanings and structures, and then, in a second pass, have to decide on the most plausible one, regarding the knowledge base and other semantic restrictions.

3 Parsing Neural Networks

We attempted to bring together the advantages of traditional parsers and neural networks, but to avoid the above mentioned problems.

Thus, we first chose context-free grammars (CFGs) to be the central structure of our parsing networks. CFGs are well investigated in CL and although some people prefer context-sensitive grammars in order to build their parsers because some theories and phenomena are more easy to handle using context-sensitive structures, CFGs are well suited for usual parsing tasks which don't incorporate too complicated inter- and intrasentential relations.

Next, we had to decide on a parsing theory well-suited to the kind of processing performed by neural networks. We chose a kind of chart-parser which has also been used by Waltz and Pollack in their famous work on word sense disambiguation. A chart-parser builds up a chart which is a kind of directed graph or tree where each leaf is a lexical item and the nodes within the tree represent non-terminals of the grammar. The parse-tree is built up incrementally such that the sentence is processed from the left to the right. The interim structures are shown in the chart to be the upper nodes. The parsing process is finished when the sentence symbol is derived or when the input sentence is finished. Then the chart displays the complete sentence structure if it was a grammatically correct input sentence. One of the advantages of this parsing method is that the input sentence is processed incrementally and we thus get information about the structures that have been processed so far. Also the processing of an input proceeds until an error (ungrammatical structure etc.) occurs, and the result achieved up to that point is kept in the chart.

In addition to deciding on a traditional parser we looked at the work on neural network parsers.

On the one hand, we found approaches which were heavily based on traditional parsing techniques such as [Sch87] and [Fan85]. They used the same structures as the equivalent traditional parser and thus merely reimplemented a symbol-oriented

Next, we give a survey of the techniques used in our approach. Important is that the parse tree and thus the parsing neural network is built up dynamically according to the rules of the given grammar.

In the following sections we describe special features of the PAPADEUS system, INKAS and INKOPA. The PAPADEUS system is able to parse a given input sentence and to disambiguate lexical and syntactical ambiguities. INKAS parses a given sentence incrementally and constructs, in addition to the parse tree, a case-based semantic representation of the input. INKOPA is a further development of INKAS which is able to construct the parsing network out of a given context-free grammar. All three systems are based on neural networks implemented as networks of finite automata.

2 The Main Problems

2.1 Problems with Connectionist Networks

The main problems in building connectionist parsers are the representation of recursive structures, the correct processing of sequences, and the representation of variables.

The known techniques of processing recursive structures in connectionist networks are usually very time- and space-consuming. Unlimited recursion requires large matrices in order to represent the resulting structures. Alternatively, one might choose to represent only a certain degree of recursion.

Also the representation of sequences, investigated particularly in the work on signal processing, for example in speech recognition, is a problem for connectionist networks. In particular, the inputs to one single unit are not processed in a serial manner; they are processed in parallel which leads to difficulties when units represent lexical items and grammar rules. Recurrent networks and other techniques have been developed to handle sequential input, but unfortunately recurrent networks are difficult to train and to analyze.

Another problem with parsing using neural networks is the representation and handling of variables. There are solutions to this problem but most of them depend on space-consuming representations of variables and their possible bindings in large matrices.[1]

2.2 Problems with Traditional Parsers

On the other hand are several severe problems with traditional symbolic parsers. They usually lack the ability to process ungrammatical sentences, which are quite

[1]In the last few years, researchers have started using synchronized oscillations of units to represent variable bindings. This technique is used for example in the Structure Unification Parser developed by James Henderson [Hen94].

Parsing Neural Networks Combining Symbolic and Connectionist Approaches

Christel Kemke
ICSI, Berkeley, CA 94704-1198, U.S.A.
DFKI, D-66123 Saarbrücken, F.R.G.
e-mail: kemke@icsi.berkeley.edu
TR-94-021

May 1994

Abstract

In this paper we suggest combining symbolic and subsymbolic approaches in order to build fast parsers based on context-free grammars. Symbol-based parsers well known in Artificial Intelligence (AI) and Computational Linguistics (CL) provide highly developed tools and techniques, but they suffer from certain inabilities, for example to process ambiguous sentences or ungrammatical structures.

Connectionist parsers, on the other hand, have problems with representing recursive structures, processing sequences, and the handling of variables. But they have the advantage of being fault-tolerant and representing syntactic and semantic knowledge in a distributed manner.

We analyzed the existing work on connectionist parsers and developed three different systems (PAPADEUS, INKAS, and INKOPA) in order to tackle the above described problems of symbolic and connectionist approaches. The main common characteristic of all three systems is the dynamic generation of the parse tree and thus of the parsing network. This technique was developed using the known parsing techniques in AI and CL, especially chart-parsing. Also the use of context-free grammars had its source in these fields.

1 Introduction

In the following, we first describe the problems of connectionist parsers and of traditional symbolic parsers and cite the relevant literature.