Towards Learning Semantics of Spontaneous Dialog Utterances in a Hybrid Framework

Volker Weber Stefan Wermter
University of Hamburg, Computer Science Department
Vogt-Kölln-Straße 30, D-22527 Hamburg, Germany
weber@informatik.uni-hamburg.de
wermter@informatik.uni-hamburg.de

Abstract. In contrast to text processing, spontaneous language contains many discontinuities caused by unusual order, false starts, repairs, repetitions, pauses, etc. Data-driven connectionist learning methods and their inherent fault tolerance can consider this form of "sequential noise". We describe a new hybrid approach for a flat semantic interpretation of sentences in spontaneous dialogs using symbolic methods for communication and simple known mappings as well as connectionist methods for unknown mappings. We describe the semantic performance of our hybrid approach using real–world spontaneous dialog utterances including discontinuity errors. Furthermore, we demonstrate that this integration of connectionist and symbolic methods can deal with spontaneous discontinuities which other more traditional strictly rule-based parsers typically cannot.

1. Introduction

During the last decade connectionist or hybrid symbolic/connectionist work on natural language has primarily concentrated on text processing. Progress could be made demonstrating the ability of certain connectionist architectures to represent symbolic properties like compositionality [1], sequentiality [2], and role binding [3]. Concurrent to these general properties, specific tasks were tackled among them learning structural analysis [4], semantic case role analysis [5], and semantic context assignment [6]. These research efforts have demonstrated the ability of connectionist networks with respect to learning and generalization. However, we believe that learning sequential classifications, processing spontaneous language in a fault-tolerant manner, and representing flat syntactic and semantic analysis have not yet been addressed sufficiently under hard real-world conditions of spontaneous language.

Our work is motivated by three basic principles for processing spontaneous language. First, *fault-tolerant processing*: spontaneous language contains messy syntax, false starts, pauses, breaks, interjections, repairs, repetitions, and other phenomena which need fault-tolerant processing. Second, *learning*: Learning and generalization is very important since especially noisy spontaneous language is highly irregular and regularities can only be partially predicted. Third, the *screening approach*: An in-depth syntactic and semantic approach may not be necessary or useful for many tasks, especially for erroneous, messy spontaneous language. Our screening approach is based on and extends an earlier hybrid approach which has been referred to as flat scanning understanding [7]. However, while the scanning understanding focused on a flat analysis of written language, our screening approach extends the flat analysis to spoken language in a new system SCREEN[1].

We have chosen the task of interpreting spontaneous language syntactically and semantically as a testbed for examining fault tolerance. Using the RTC[2] corpus of spontaneous spoken and transcribed utterances at a railway counter we have designed a hybrid architecture SCREEN for learning fault-tolerant processing of spontaneous language [8]. This architecture consists of many symbolic or connectionist modules working in parallel for syntactic and semantic processing, speech analysis, and error correction. The communication and integration of these modules is performed by an incremental parallel interaction, similar to message passing. In this paper we present new results on learning flat semantic representations. We describe the overall parallel architecture with an emphasis on semantic processing and the performance on assigning semantic flat interpretations to noisy real-world spontaneous language.

2. Fault-tolerant flat Semantic Analysis

As data material we used the RTC corpus which contains transcriptions of spoken dialogs at a railway counter. These spontaneous real utterances incorporate various forms of "noise", for instance, interjections (eh, ...), hesitations (mm, ...), repetitions (the the track ...), repairs (in / at the sea ...), etc. For our initial experiments we used 95 of these utterances. Typical examples are:

Category name	Examples for basic semantic category
need	need, would like
move	go, ride
state	know, exist
aux	can, could
say	say, ask
question	which, when (question words)
physical	train, wagon (physical objects)
animate	I, you (animate objects)
abstract	connection, class (abstract objects)
here	on, in (time or location state words, prepositions)
source	from, (time or location source words prepositions)
destination	to (time or location destination words prepositions)
location	Frankfurt, Hamburg
time	tomorrow, 3 o' clock
how	with, without
negation	no
nill	the (words "without" specific semantics like determiner)

Table 1: Basic semantic categories

[1] SCREEN stands for Symbolic Connectionist Robust EnterprisE for Natural language.

[2] Corpus compiled at the University of Regensburg (FRG) containing travel inquiries.

- Yeah, I need a train from Regensburg to Dortmund via Köln · with at least two hours time in Köln

- when leaves please · [eh] a train · from Regensburg to Dortmund · at Monday [mm] [ts] [u] · at Monday · morning

- I need a ticket · to Hamburg · and wonna ask therefore [w] · · when and which track that train leaves then

While the first sentence is fairly regular the second one contains a self repair (at Monday ... at Monday morning), an interjection (eh) and a hesitation (mm). The third sentence shows irregular syntax. It also contains pauses (·) and a break within a word (w . . when). Such errors cause a lot of problems for traditional symbolic syntactic parsers and also for semantic analyzers since they interrupt the expected sequence of constituents and violate the symbolically encoded syntactic and semantic rules. However, in real spoken language unusual syntax and errors like false starts, interruptions, hesitations, repairs, repetitions occur fairly often and any realistic speech/language system for spoken language has to deal with them.

Rather than representing an utterance as hierarchical tree structure we represent an utterance as a set of different flat sequential representations, e.g. for basic syntactic categories (like nouns), abstract syntactic categories (like prepositional phrases), and basic and abstract semantic categories described below in more detail. In previous work we focused on learning flat syntactic representations for such utterances [8]. In this paper we will focus on learning flat semantic representations. Such flat representations particularly support fault-tolerant processing since complex graphs or trees do not have to be restructured. Bellow we show an example of flat representations for a part of the second utterance from above.

...	from	Regensburg	to	Dortmund	at	Monday	[...]	at	Monday	morning
...	source	location	destination	location	here	time	nill	here	time	time
...	loc-from		loc-to		time-at			time-at		

A basic semantic category is assigned to each word of an utterance (from ← source, Regensburg ← location, ...) and an abstract semantic category to each phrase (from Regensburg ← loc-from, to Dortmund ← loc-to, ...).

In table 1 we show the basic semantic categories as well as several examples for illustration. These basic semantic categories were developed based on the used travel domain from the RTC corpus. Each word can belong to one or more semantic categories. The abstract semantic categories (see table 2) are more general than the basic categories. They are fairly independent from the domain with a slight focus on time and location. Although it can be argued that any form of semantic analysis is difficult since semantics relies on a choice of semantic primitives for a particular domain, the used semantic categories are rather general and only a few domain-dependent changes should be necessary for a transfer to a new domain. Furthermore, they are also similar to general well-known work by Allen and Fillmore [9, 10].

3. Overview of the modules

Based on principles of fault-tolerant processing, learning, and screening understanding we designed a parallel incremental hybrid architecture. The input is a stream of word

Category name	Abstract semantic category
action	action for full verb events
aux-action	auxiliary action for auxiliary events
agent	agent of an action
object	object of an action
recipient	recipient of an action
instrument	instrument for an action
manner	how to achieve an action
time-at	at what time
time-from	start time
time-to	end time
loc-at	at which location
loc-from	start location
loc-to	end location
question	question phrases
misc	miscellaneous words (e.g. for politeness)

Table 2: Abstract semantic categories

hypotheses from an underlying speech recognizer. The output is a stream of semantic and syntactic hypotheses about the syntactic and semantic properties of the utterance. The overall architecture contains 5 parts, which are shown in figure 1: the speech interface part, the category part, the correction part, the subclause part, and the case frame part. Most modules are realized by connectionist feedforward and recurrent networks, but there are also symbolic modules for simple mappings, like the detection of interjections. All connectionist networks are embedded and encapsulated in a symbolic communication interface so that symbolic messages can be exchanged between different cooperating modules.

The *speech interface part* analyses the syntactic and semantic plausibility of input from a speech recognizer. The *category part* receives sentence part hypotheses and provides syntactic and semantic category preferences for words. The category part contains modules for the disambiguation of basic syntactic and semantic categories (BAS-SYN-DIS, BAS-SEM-DIS, see figure 2), the categorization of abstract syntactic and semantic categories (ABS-SYN-CAT, ABS-SEM-CAT, see figure 3), and the identification of phrase starts (PHRASE-START?). The *correction part* checks and corrects pause errors (pauses, interjections, word breaks), word errors, and phrase errors which might occur within sentences. Modules exist for hesitation detection (INTERJECTION?, PAUSE), for the detection of word errors due to the lexical, syntactic and semantic equality of two subsequent words (LEX-WORD-EQ?, BAS-SYN-EQ?, BAS-SEM-EQ?, see figure 4), and for the detection of phrase errors due to the same lexical start and the syntactic, and semantic equality of two subsequent phrases (LEX-START-EQ?, ABS-SYN-EQ?, ABS-SEM-EQ? see figure 4). The *subclause part* contains triggers for identifying individual subclauses within sentences and causes the system to generate new frames for subclauses (GEN-FRAME?). Finally the *case frame part* provides syntactic and semantic hypotheses about the incremental parts of the sentence hypotheses by filling slots with words (SLOT-FINDING) and checking for constraints attached to the slots (VERB-ERROR?, SLOT-ERROR?).

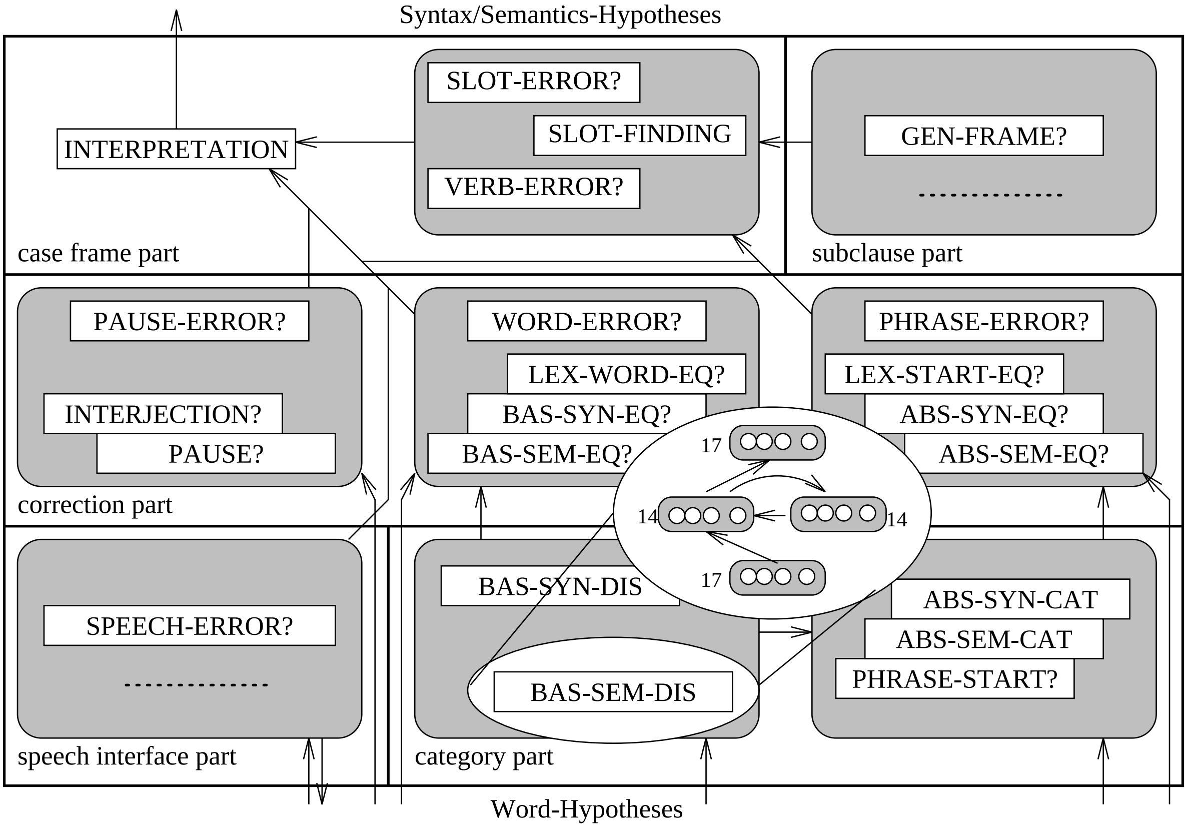

Figure 1: SCREEN: some modules of the five basic parts

4. Representative Training Results

In this section we illustrate the performance of learning a flat semantic analysis using connectionist modules. For the module BAS-SEM-DIS, input is a sequence of words and output is a sequence of disambiguated basic semantic categories. For the module ABS-SEM-CAT a sequence of words with their disambiguated basic semantic categories is mapped to a sequence of abstract semantic categories. Since both tasks are sequential learning tasks simple recurrent networks [2] have been used for training and generalization. For each training item the number of input and output units depends on the respective word representation. There are 17 input and output units for BAS-SEM-DIS for the 17 basic semantic categories. In figure 2 the 17 input and output units are labeled with their interpretation. The activation values in the input layer of this figure represent the semantic entry of our lexicon for the word 'switch'. Switch could be an abstract object, physical object, or 'move' event. In this context the move event has been chosen and therefore the output layer represents the disambiguated semantic 'move' for the word 'switch'. For the module ABS-SEM-CAT there are 17 input units (which is the output of the disambiguation of BAS-SEM-DIS) for the basic semantic categories, and 15 output units for the abstract semantic categories. In figure 3 the units of the input and output layers are labeled with the corresponding interpretations.

BAS-SEM-EQ? (ABS-SEM-EQ?) tests whether the basic (abstract) semantic categories of two subsequent words are equal. For BAS-SEM-EQ? and ABS-SEM-EQ? we use feedforward networks (figure 4), since their input (the output of BAS-SEM-DIS and

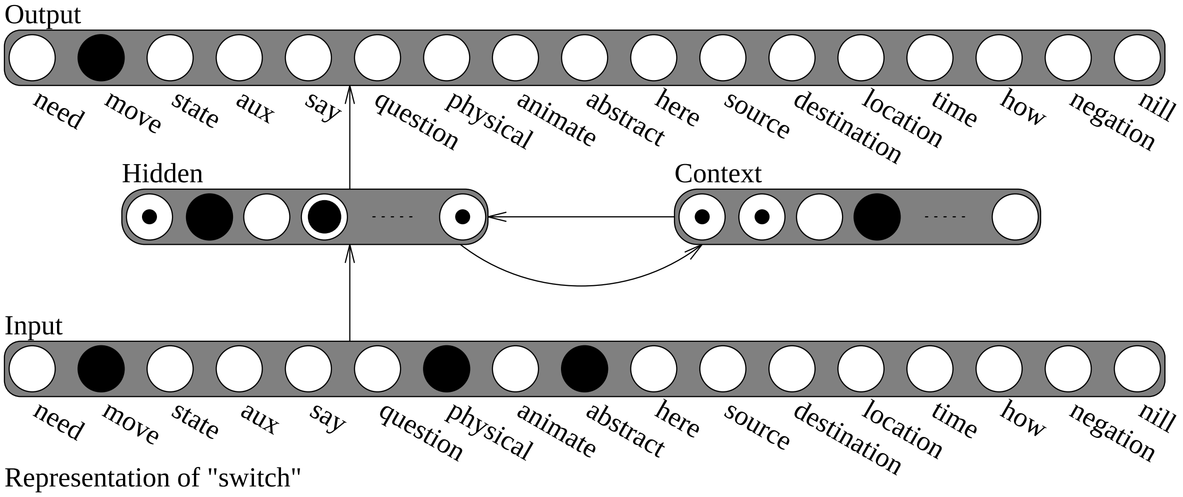

Figure 2: BAS-SEM-DIS: Elman-network for disambiguation

ABS-SEM-CAT) is a collection of analog values. The input to BAS-SEM-EQ? is the disambiguation result of BAS-SEM-DIS for a word and its predecessor. For ABS-SEM-EQ? the input is the categorization result of ABS-SEM-CAT for a word and the final categorization result for the previous phrase. Therefore we use 17 basic (15 abstract) semantic categories per word, that is 34 (30) input units for two words. The output consists of two units: equality and its negation (not equal) to provide the possibility for faster training. Based on many empirical tests the number of hidden units (resp. context units) was set to 14 for BAS-SEM-DIS and ABS-SEM-CAT and to 7 and 8 for BAS-SEM-EQ? and ABS-SEM-EQ? respectively.

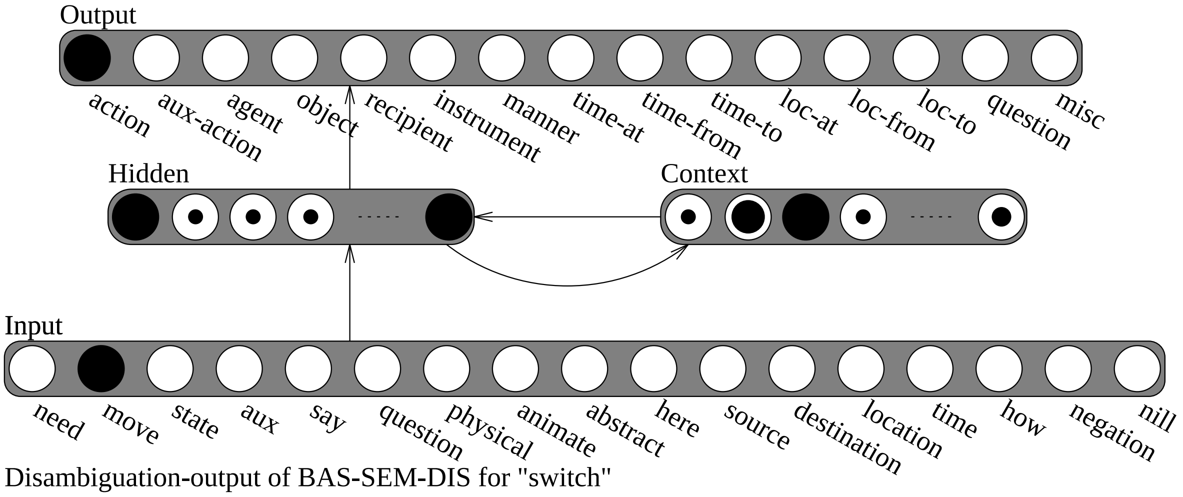

Figure 3: ABS-SEM-CAT: Elman-network for categorization

The overall performance for training and test sets is illustrated in table 3. The results for BAS-SEM-DIS and ABS-SEM-CAT are based on a training set of 393 training words from 37 training sentences and 823 test words from 58 unknown test sentences. We count a training or test instance as assigned correctly if the output unit with

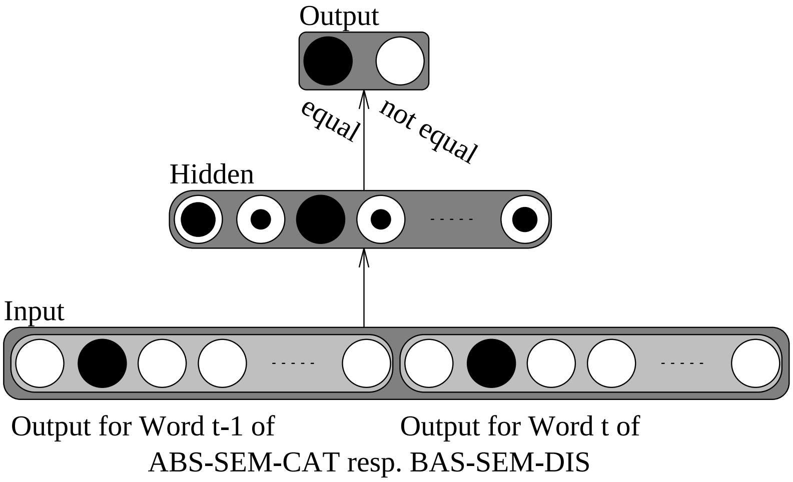

Figure 4: BAS-SEM-EQ and ABS-SEM-EQ: Feedforward-networks for testing equality

the maximal activation is equal to the desired category; otherwise we count it as an error. Our first example (BAS-SEM-DIS) illustrates that 96% of the training and 84% of the test set have been learned. This high performance is representative also for syntactic modules. On the other hand, ABS-SEM-CAT is a much more difficult learning task because the order of abstract semantic categories is much less constrained. Since training for equality was rather easy the performance for the training and test set came close to perfection for BAS-SEM-EQ and ABS-SEM-EQ.

Module	No. of units			correct assignments	
	I	H	O	train	test
BAS-SEM-DIS	17	14	17	96%	84%
BAS-SEM-EQ?	34	8	2	99%	98%
ABS-SEM-CAT	17	14	15	81%	77%
ABS-SEM-EQ?	30	7	2	98%	95%
PHRASE-START?	13	7	1	93%	89%

Table 3: Training and generalization performance of some semantic modules

5. An Example

The sentence shown in figure 5 has an ill-formed syntax and contains some ungrammatical phenomena like pauses and a word break. The first column illustrates the words of the utterance, the second the basic semantic category, the third the abstract semantic category and the fourth the phrase-starts within the utterance. In SCREEN parsing is incremental and parallel. So syntax and semantics (BAS-SYN-DIS and BAS-SEM-DIS resp. ABS-SYN-CAT and ABS-SEM-CAT) work at the same time. Here we illustrate the semantic performance but learned syntactic performance for faulty sentences has been described in [8].

The final parse contains only the underlined words where errors have been eliminated. As we can see, pauses and word breaks have been eliminated correctly. Further-

UTTERANCE	BAS-SEM-DIS		ABS-SEM-CAT		PHRASE-START?
I	■	animate	■	agent	■
need	■	need	■	action	■
a	■	nill	▪	misc	■
ticket	·	nill	▪	object	▫
.	■	nill	■	misc	▫
to	▪	destination	▪	loc-to	■
Hamburg	■	location	■	loc-to	□
.	■	nill	■	misc	▫
and	■	nill	■	misc	■
wonna	■	aux	■	aux-action	■
ask	·	need	▪	action	■
therefore	■	nill	■	misc	■
[w]	■	nill	■	misc	■
.	■	nill	■	misc	□
.	■	nill	■	misc	□
when	·	question	▪	time-at	■
.	■	nill	■	misc	□
and	■	nill	■	misc	■
which	·	question	▪	time-at	■
track	■	location	■	loc-at	▫
that	■	nill	■	misc	■
the	■	nill	■	misc	■
train	·	question	▪	object	▫
leaves	■	move	■	action	■
then	■	nill	■	misc	■

■	positive and
□	negative activation
size	strength of activation

Figure 5: Sentence with corrections

more highly unusual syntactic constructions have been dealt with. For instance, the sequence "which track that the train leaves then" would be difficult to analyze based on known syntactic and semantic rule representations.

In figure 5 'I' is disambiguated to be an *animate* being and afterwards categorized to be the *agent*. This word starts a new phrase. The same is done for the next word 'need' assigned to the categories *need* and *agent*. The phrase 'a ticket' is assigned correctly by PHRASE-START? since 'a' introduces the phrase and 'ticket' does not. There is a small miss for the disambiguation since 'ticket' is very weakly assigned as *nill* by BAS-SEM-DIS but the more important ABS-SEM-CAT does well. It is also essential to interpret the final abstract semantic category at the right end of a phrase as the system output. So 'a ticket' is finally assigned to *object* and not to *misc* as hypothesized at the beginning of this phrase. The same holds for 'which track' which is *loc-at* rather then *time-at*.

6. Discussion and Conclusion

We have described the flat semantic interpretation of sentences in spontaneous dialogs using various connectionist feedforward and recurrent networks. The hybrid architecture as well as the networks are incremental and can run in parallel. The choice of a hybrid connectionist architecture was primarily motivated by the fault-tolerant learning behavior of connectionist networks and the advantages of an explicit symbolic message passing control. The fault tolerance and data-driven learning has a lot of potential to model faulty and messy real-world spoken utterances.

So far connectionist networks for syntactic and semantic language processing have been tested on relatively well-formed texts, in early work sometimes with artificially generated text [11, 5]. While such work examined learning rule-like behavior, it also reduced the ability of connectionist networks to demonstrate their strong ability of robust fault-tolerant behavior. One notable exception is work by Jain and Waibel [12, 13]. Especially the connectionist parser PARSEC demonstrated the possibility of fault-tolerant learning of flat semantic representations although the focus was not yet on dealing with *real-world faulty* dialog utterances. We use the ability of connectionist networks for representing real-world, spontaneous, and potentially ill-formed language. We argue that connectionist networks provide good performance - even under real-world conditions - if they are used for tasks that particularly need robust fault-tolerant learning of flat representations.

Acknowledgements

This research was supported in part by the German Research Community (DFG) under contract DFG Ha 1026/6-1 and by the Federal Secretary for Research and Technology (BMFT) under contract #01IV101A0. We would like to thank Stefanie Haack, Matthias Löchel, Manuela Meurer, Ulf Peters, and Ulf Sauerland for their assistance with labeling the corpus and training various networks.

References

[1] Jordan B. Pollack. Recursive distributed representations. *Artificial Intelligence*, 46(1-2):77–105, 1990.

[2] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[3] Charles Patrick Dolan and Michael G. Dyer. Symbolic schemata, role binding and the evolution of structure in connectionist memories. In *Proceedings of the 1st International Conference on Neural Networks*, volume II, pages 287–298. (San Diego, CA), 1987.

[4] Mark A. Fanty. Context-free parsing in connectionist networks. Technical Report 174, University of Rochester, Rochester, NY, November 1985.

[5] Mark F. St. John. *The Story Gestalt – Text Comprehension by Cue-Based Constraint Satisfaction.* PhD thesis, Department of Psychology, Carnegie Mellon University, 1990.

[6] Stefan Wermter. A hybrid and connectionist architecture for a SCANning understanding. In *Proceedings of the 10th European Conference on Artificial Intelligence*, pages 188–192, Vienna, Austria, 1992.

[7] Stefan Wermter. *A Hybrid Connectionist Approach for a Scanning Understanding of Natural Language Phrases.* PhD thesis, Universität Hamburg, Hamburg, FRG, May 1993.

[8] Stefan Wermter and Volker Weber. Learning fault-tolerant speech parsing with SCREEN. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, volume 1, pages 670–675, Menlo Park, July/August 1994. Seattle, Washington, AAAI Press/The MIT Press.

[9] James Allen. *Natural Language Understanding.* Benjamin/Cummings Publishing Company, Menlo Park, CA, 1987.

[10] Charles J. Fillmore. The case for case. In E. Bach and R. Harms, editors, *Universals of Linguistic Theory*, pages 1–90. 1979.

[11] James L. McClelland and Alan H. Kawamoto. Mechanisms of sentence processing: Assigning roles to constituents of sentences. In James L. McClelland, David E. Rumelhart, and The PDP research group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 2., Psychological and Biological Models, chapter 19, pages 272–331. MIT Press, Cambridge, MA, 1986.

[12] Ajay N. Jain. Generalization performance in PARSEC - a structured connectionist parsing architecture. In John E. Moody, Steve J. Hanson, and Richard R. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 209–216. Morgan Kaufmann, San Mateo, CA, 1992.

[13] Alexander Waibel, Ajay N. Jain, A. McNair, J. Tebelskis, L. Osterholtz, H. Saito, O. Schmidbauer, T. Sloboda, and M. Woszczyna. JANUS: Speech-to-speech translation using connectionist and non-connectionist techniques. In John E. Moody, Steve J. Hanson, and Richard R. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 183–190. Morgan Kaufmann, San Mateo, CA, 1992.