Hybrid Neural and Symbolic Language Processing

Stefan Wermter, University of Sunderland, UK

Abstract

In this paper we outline hybrid approaches to artificial neural network-based natural language processing. We suggest various types of symbolic/connectionist integration for language processing: connectionist structure architectures, hybrid transfer architectures, hybrid processing architectures. Furthermore, we focus particularly on loosely coupled, tightly coupled, and fully integrated hybrid processing architectures. We argue that the hybrid approach to artificial neural network-based language processing has a lot of potential to overcome the gap between a neural level and a symbolic conceptual level.

1 Motivation for hybrid processing

In recent years, the field of hybrid symbolic/connectionist processing has seen a remarkable development [21, 17, 27, 16, 30, 4]. Currently it is still an open issue whether connectionist or symbolic approaches alone will be sufficient to provide a general framework for processing natural language [5, 10, 26]. However, since human language capabilities are based on real neural networks in the brain, artificial neural networks (also called connectionist networks) provide one essential starting point for modeling language processing. On the other hand, human language capabilities include rule-based reasoning which is supported well by symbolic processing. Given this general situation, the motivation for examining hybrid connectionist models of natural language processing comes from different directions.

From the perspective of cognitive neuroscience, a symbolic interpretation of a connectionist architecture is desirable, since the brain has a neuronal structure and has the capability to perform symbolic reasoning. This leads to the question how different processing mechanisms can bridge the large gap between, for instance, acoustic or visual input signals and symbolic reasoning for language processing. The brain uses a complex specialization of different structures. Although a lot of the functionality of the brain is not yet known in detail, its architecture is highly specialized and organized at various levels of neurons, networks, nodes, cortex areas and their respective connections. Furthermore, different cognitive processes are not homogeneous and it is to be expected that they are based on different representations. Therefore, there is evidence from the brain that multiple architectural representations may also be involved in language processing.

From the perspective of knowledge-based natural language processing, hybrid symbolic/connectionist representations are advantageous, since different mutually complementary properties can be combined. Symbolic representations have advantages with respect to easy interpretation, explicit control, fast initial coding, dynamic variable binding and knowledge abstraction. On the other hand, connectionist representations show advantages for gradual analog plausibility, learning, robust fault-tolerant processing, and generalization to similar input. Since these advantages are mutually complementary, a hybrid symbolic connectionist architecture can be useful if different processing strategies have to be supported.

The development of hybrid symbolic/connectionist architectures is still a new research area and there is no general theory of hybrid architectures. In general, there are different global possibilities for a symbolic/connectionist integration. Based on the argument from cognitive neuroscience, symbolic connectionist integration relies on a symbolic interpretation of a connectionist architecture and connectionist processing. There are just connectionist representations for different tasks which can be interpreted symbolically. On the other hand, based on the argument from knowledge-based natural language processing, symbolic connectionist integration relies on a combination of different symbolic and connectionist representations. However, these interpretation approaches and representation approaches are closely related and in general there is a continuum of possible connectionist/symbolic architectures.

2 Types of symbolic/connectionist integration

Although there has been quite a lot of work in the field of hybrid and connectionist natural language processing, most work has concentrated on a single individual system rather than on types of hybrid systems, their interpretation, and communication principles within various architectures.

In figure 1 there is an overview of different possibilities for integration in natural language processing. Continuous connectionist representations are represented by a circle, discrete symbolic representations by a square. Symbolic interpretations of connectionist representations are shown as squares with dotted lines.

Connectionist structure architectures are the first type of symbolic/connectionist architectures. They can model higher cognitive functions and rely solely on connectionist representations. Symbolic knowledge arises by an interpretation process of the connectionist representations. Often specific knowledge of the task is built into the connectionist structure architecture.

Hybrid transfer architectures transfer symbolic representations into connectionist representations or vice versa. Using a transfer architecture it is possible to insert or extract symbolic knowledge into or from a connectionist architecture. The main processing is performed by connectionist representations but there are automatic procedures for transferring from connectionist representations to symbolic representations or vice versa. Hybrid transfer architectures differ from connectionist structure architectures by the automatic transfer into and from symbolic representations. While certain units in connectionist structure architectures may be interpreted symbolically by an observer, only hybrid transfer architectures allow the knowledge transfer into rules, automata, grammars, etc.

Hybrid transfer architectures transfer symbolic and connectionist representations, but the symbolic and connectionist knowledge is not yet applied at the same time to combine the complementary advantages of each representation for a given task. Such a combination of symbolic and connectionist representations is possible in *hybrid processing architectures*, which contain both symbolic and connectionist modules appropriate to the task. Here, symbolic representations are not just initial or final representations. Rather, they are combined and integrated with connectionist representations in many different ways.

Connectionist and symbolic modules in hybrid processing architectures can be loosely coupled, tightly coupled or completely integrated (see figure 1). A *loosely coupled hybrid architecture* has separate symbolic and connectionist modules. The control flow is sequential in the sense that processing has to be finished in one module before the next module can

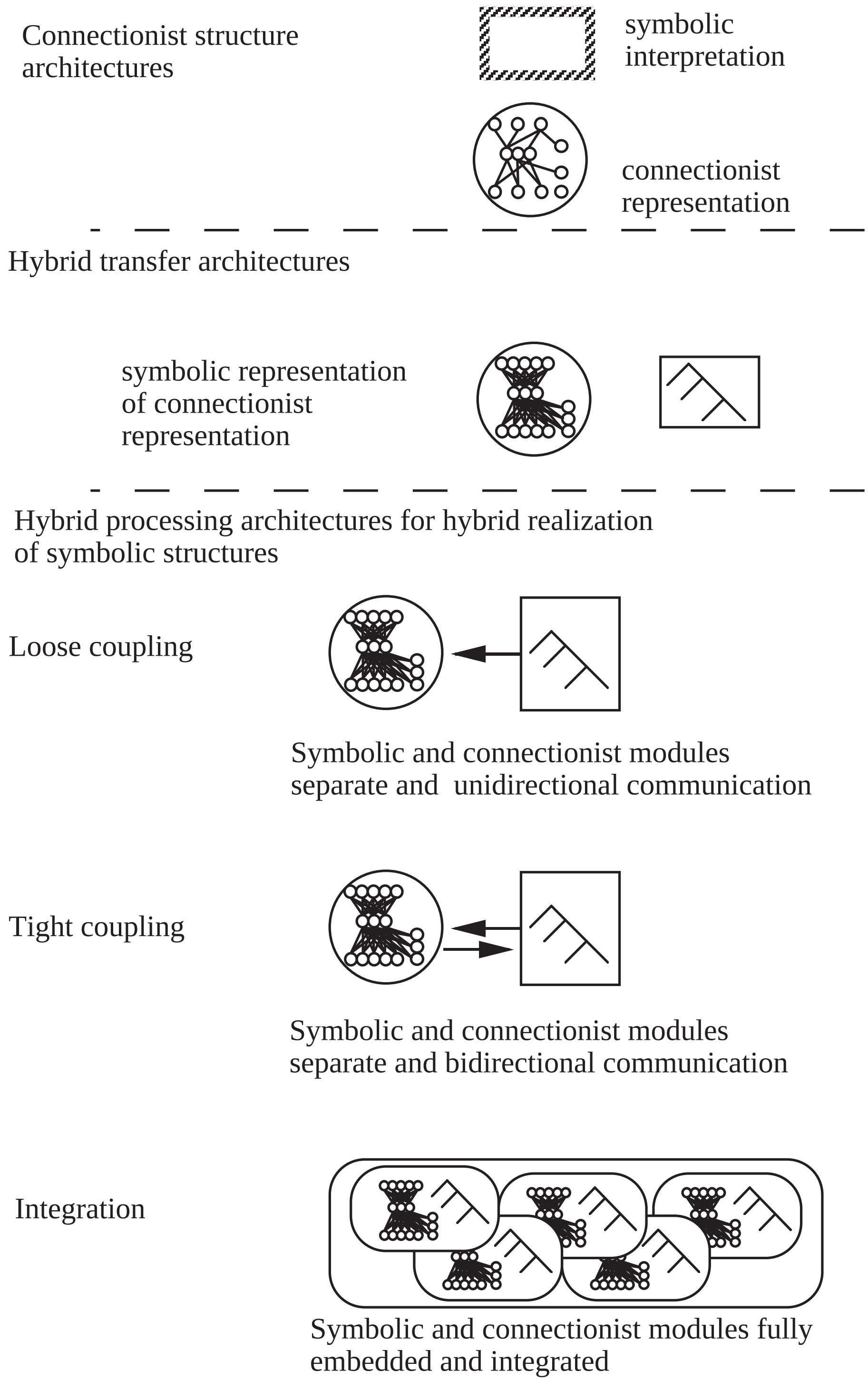

Figure 1: Overview about various types of symbolic/connectionist integration

begin. Only one module is active at any time, and the communication between modules is unidirectional.

A *tightly coupled hybrid architecture* contains separate symbolic and connectionist modules, and control and communication are via common internal data structures in each module. The main difference between loosely and tightly coupled hybrid architectures is common data structures which allow bidirectional exchanges of knowledge between two or more modules. Processing is still in a single module at any given time, but the output of a connectionist module can have direct influence on a symbolic module (or vice versa) before it finishes its global processing. In this way feedback between two modules is possible and the communication can be bidirectional.

In a *fully integrated hybrid architecture* there is no discernible external difference between symbolic and connectionist modules, since the modules have the same interface and they are embedded in the same architecture. The control flow may be parallel and the communication between symbolic and connectionist modules is via messages. Communication may be bidirectional between many modules, although not all possible communication channels have to be used. This is the most advanced of the hybrid processing architectures. In the remainder of this article we will show detailed examples of each of these types of symbolic/connectionist architectures.

3 Connectionist structure architectures

In this section we describe principles of connectionist structure architectures. Knowledge of some task is built into the connectionist structure architecture, and a symbolic interpretation is assigned by an observer. Much early work on structured connectionism can be traced back to work by Feldman and Ballard, who provided a general framework of structured connectionism [7]. This framework was extended for natural language processing in many different directions, for instance for parsing [6] and explanation [3].

More recent work along these lines focuses on the so-called *NTL, neural theory of language* which attempts to bridge the large gap between neurons and cognitive behavior [8, 22]. The NTL framework is challenging since it tries to study neural processing mechanisms for high conceptual cognitive processing like embodied semantics, reasoning, metaphor interpretation, etc. In general, within the NTL framework, structured connectionist networks are seen as central for modeling systems of natural language processing. However, above this level of structured connectionist networks there is a computational level which provides interpretations at a symbolic level and which provides links to a conceptual linguistic level. On the other hand, below the level of structured connectionist networks there is a computational neurobiology level which links connectionist networks with the biological level.

We will now focus on a few examples of different structured connectionist architectures which have been developed in recent years. One example of a connectionist structure architecture is provided as part of the SCAN project[27]. One task within this framework is structural disambiguation of a given sentence or phrase. In general many different structural interpretations of phrases are possible and only the integration of different constraints allows the system to make a correct structural disambiguation decision. Such an integration of competing constraints of varying strengths can be represented well in a localist connectionist network, since the explicit nodes and connections in the network can be used to model these competing constraints. This localist network represents the possible static connections for the structural disambiguation task. The relevant semantic knowledge can be learned in distributed connectionist plausibility networks.

Another recent connectionist structure architecture is CONSYDERR [25]. The task of this system is to make inferences, as they are required for instance in natural language understanding systems. The system has an interesting architecture since it consists of a structured connectionist architecture with two different components. The localist level represents inference knowledge at a more abstract concept level, while the distributed level represents the details of the inferencing process at the feature level. Inferences are represented either directly within the localist component or indirectly via flow of activation within the distributed component.

There are many more connectionist architectures which allow a symbolic interpretation for instance at the output layers of their network. There are modular network architectures for case role assignment in Gestalt-networks [24], the mixture of expert networks for hierarchical control [11] and the two-tier architecture for corpus analysis [1]. All this recent work contains various forms of modularity and structuring within an overall connectionist architecture. Furthermore, symbolic interpretation of the results is possible, at least at certain nodes.

4 Hybrid transfer architectures

Hybrid transfer architectures transfer symbolic representations into connectionist representations or vice versa. Using a transfer architecture it is possible to insert or extract symbolic knowledge into or from a connectionist architecture. Hybrid transfer architectures differ from connectionist structure architectures by this automatic transfer into and from symbolic representations. While certain units in connectionist structure architectures may be interpreted symbolically only hybrid transfer architectures allow the knowledge transfer into or from rules, automata, grammars, etc.

There are many ways to insert or extract symbolic representations into or from connectionist representations. The range of possibilities depends on the form of the symbolic representations (e.g. context-free rules, automata, etc) and the connectionist representations (e.g. use of weights, activations, etc). In this section, we will show two representative approaches for extracting knowledge from connectionist networks. We have chosen these approaches since they have been widely explored during the last few years and since they are relatively straightforward.

One major problem for hybrid transfer architectures is that the architecture itself has to support the transfer. A good example is the work on activation-based automata extraction from recurrent networks [19]. First simple finite state automata are used for generating training examples of a given regular grammar. These training sequences are used to train a second order connectionist network as an acceptor for the regular grammar. This network has been designed particularly to meet the constraints of symbolic automata behavior. For a given input and a given state, the network determines the next state. For the extraction of symbolic representations from a trained connectionist network, the output unit activations are partitioned into q discrete intervals. Therefore, there are q^n partitions for n units. The presentation of all input patterns generates a search tree of transitions of an automaton where one node corresponds to one partition. In general, this architecture is a hybrid transfer architecture where symbolic automata are extracted from connectionist networks.

The activation-based extraction of automata is just one example of a hybrid transfer architecture and there are several further possibilities. For instance, a weight-based transfer between symbolic rules and feedforward networks has been extensively examined in knowledge based artificial neural networks (KBANN) [23]. This weight-based transfer uses the weights rather than the activations as the main knowledge source for induction and extraction. While an activation-based transfer is based on the activations of certain units a weight-based transfer focuses on a more detailed weight level. Therefore, the architectures are fairly simple, in this case three-layer feedforward networks.

In further work, activation-based transfer between context-free rules and feedforward networks is described in a symbolic manipulator for context-free languages [18], and another form of weight-based insertion of symbolic rules has been proposed for recurrent networks [15].

5 Hybrid processing architectures

Hybrid transfer architectures do not apply symbolic and connectionist knowledge simultaneously to combine the complementary advantages of each representation in solving a given task. Hybrid transfer architectures just transfer symbolic and connectionist representations into each other. However, a combination of mutual advantages of symbolic and connectionist representations may be advantageous for "hybrid tasks". This combination of symbolic and connectionist representations during task processing is performed in *hybrid processing architectures* which contain symbolic and connectionist modules.

5.1 Loosely coupled hybrid processing architectures

As we have outlined above, connectionist and symbolic modules in hybrid processing architectures can be loosely coupled, tightly coupled or completely integrated (see figure 1). In a *loosely coupled hybrid architecture* processing has to be completed in one module before the next module can begin. For instance, the WP model [20] for phenomenologically plausible parsing used a symbolic chart parser to construct syntactic localist networks which were also connected to semantic networks. Although the network itself may be viewed as a purely connectionist architecture, the overall architecture is a hybrid processing architecture. First a symbolic parser is used for network building, and after the chart parser has finished its processing the connectionist spreading activation in localist networks takes place.

Another architecture where the division of symbolic and connectionist work is even more loosely coupled has been described in a model for structural parsing within the SCAN framework [27]. First, a chart parser is used to provide a structural tree representation for a given input, for instance for phrases or sentences. In a second step, triples of "noun relationship noun" are used as input for several feedforward networks which produce a plausibility measure of the relationship. Based on this connectionist output, another symbolic restructuring component changes the original tree representation if the semantic feedforward networks indicate that this is necessary.

This system has a loosely coupled hybrid processing architecture since there is a clear division between symbolic parsing, connectionist semantic analysis, and symbolic restructuring. Only if the preceding module has finished completely, will the subsequent module start its processing. The architecture and interfaces are relatively simple to realize due to the sequential control between symbolic and connectionist processing. On the other hand, this simple sequential sequence of symbolic and connectionist processing does not support feedback mechanisms.

There are several other loosely coupled hybrid processing architectures. For instance, in the SICSA system, connectionist recurrent networks and symbolic case role parsing are combined for semantic analysis of database queries [2]. In BerP, connectionist feedforward networks for speech processing and symbolic dialog understanding are combined in a spoken language dialog system [13]. As in the WP model, in all these loosely coupled hybrid processing architectures the connectionist module has to finish before the symbolic module can start (and vice versa).

5.2 Tightly coupled hybrid processing architectures

A *tightly coupled hybrid architecture* allows multiple exchanges of knowledge between two or more modules. Processing is still a single module at a time, but the result of a connectionist module can have a direct influence on a symbolic module (or vice versa) before it finishes its global processing. For instance, CDP is a system for connectionist deterministic parsing [14]. While the choice of the next action is performed in a connectionist feedforward network, the action itself is performed in a symbolic module. During the process of parsing, control is switched back and forth between these two modules, but processing is confined to a single module at a time. Therefore, a tightly coupled hybrid architecture has the potential for feedback to and from modules.

Other tightly coupled hybrid processing architectures include e.g. [9] where the control changes between symbolic marker passing and connectionist similarity determination. These tightly coupled hybrid processing architectures interleave symbolic and connectionist processing at the module level and allow the control switch between these different modules. Therefore, this tight coupling has the potential for more powerful interactions. On the other hand, the development of such architectures needs more dynamic and more complex interfaces in order to support the dynamic control between symbolic and connectionist modules.

5.3 Fully integrated hybrid processing architectures

The modules in a *fully integrated hybrid architecture* have the same interface. They are embedded in the same architecture and externally there is no way to distinguish a symbolic from a connectionist module. The control flow may be parallel and the communication between symbolic and connectionist modules is via messages. This is the most integrated and interleaved version of the hybrid processing architectures.

One example of an integrated hybrid architecture is SCREEN which was developed for exploring integrated hybrid processing for spontaneous language analysis. Here we focus primarily on the architectural principles of this approach while other task-related details can be found in [28, 31, 29]. One main architectural motivation is the use of a common interface between symbolic and connectionist modules which are externally indistinguishable. This allows incremental and parallel processing involving many different modules. Besides this architectural motivation there are also other task-oriented motivations of exploring learning and fault-tolerance in a hybrid connectionist architecture for robust processing of faulty spoken language.

SCREEN is an integrated architecture since it does not crucially rely on a single connectionist or symbolic module. Rather, there are many connectionist and symbolic modules, but they have a common interface, and they can communicate with each other in many directions. From a module-external point of view it does not matter whether the internal processing within a module is connectionist or symbolic. This architecture therefore exploits a full integration of symbolic and connectionist processing at the module level. Furthermore, the modules can run in parallel and produce an analysis in an incremental manner.

Other integrated architectures can be found in PARSEC, a system for language analysis in the conference registration domain [12], which uses Jordan networks to trigger symbolic transformations within the modules, and CONNCERT [32], which has been developed for technical control problems using a mixture of expert networks controlled by symbolic supervisors.

6 Summary and Conclusions

We have provided an overview of the foundations of hybrid connectionist architectures and a classification of architectures. Furthermore, we have described some representative and current examples of such hybrid connectionist architectures. From this perspective it is clear that hybrid symbolic/connectionist techniques have become an important class of artificial intelligence techniques. Hybrid techniques can be used successfully for those problems where the tasks require different discrete and analog modes of computation and representation. Natural language processing is one important area for hybrid architectures, since natural language processing can be supported well by symbolic structure manipulations as well as connectionist gradual plausibility, learning, and robustness.

Hybrid architectures constitute a continuum from connectionist structure architectures with symbolic interpretation to hybrid transfer architectures and hybrid processing architectures with symbolic and connectionist representations. Today, hybrid connectionist architectures provide technology for realizing larger real-world systems for natural language processing. For instance, the SCREEN system uses a hybrid processing architecture including many different modules at speech, syntax, semantics, and dialog processing levels. While early work on connectionist and hybrid architectures often had to focus on small tasks and small architectures in order to gain initial knowledge about the possibilities of combination and integration, we now begin to see the potential of hybrid connectionist systems being used for larger real world tasks in natural language processing.

Acknowledgments

I would like to thank Jerry Feldman, Lokendra Shastri, Srini Narayanan and the members of the NTL research group at ICSI, Berkeley for their support and discussions. An earlier longer version of this paper is available as ICSI-TR-30-97.

References

[1] L. Bookman. *Trajectories through knowledge space.* Kluwer, Boston, 1994.

[2] Y. Cheng, P. Fortier, and Y. Normandin. A system integrating connectionist and symbolic approaches for spoken language understanding. In *Proceedings of the International Conference on Spoken Language Processing*, pages 1511–1514, Yokohama, 1994.

[3] J. Diederich and D. L. Long. Efficient question answering in a hybrid system. In *Proceedings of the International Joint Conference on Neural Networks*, Singapore, 1992.

[4] G. Dorffner. *Neural Networks and a New AI.* Chapman and Hall, London, UK, 1996.

[5] M. G. Dyer. Symbolic neuroengineering for natural language processing: a multilevel research approach. In J. A. Barnden and J. B. Pollack, editors, *Advances in Connectionist and Neural Computation Theory, Vol.1: High Level Connectionist Models*, pages 32–86. Ablex Publishing Corporation, Norwood, NJ, 1991.

[6] M. A. Fanty. Learning in structured connectionist networks. Technical Report 252, University of Rochester, Rochester, NY, 1988.

[7] J. A. Feldman and D. H. Ballard. Connectionist models and their properties. *Cognitive Science*, 6:205–254, 1982.

[8] J. A. Feldman, G. Lakoff, D. R. Bailey, S. Narayanan, T. Regier, and A. Stolcke. L_0 - the first five years of an automated language acquisition project. *AI Review*, 8, 1996.

[9] J. Hendler. Developing hybrid symbolic/connectionist models. In J. A. Barnden and J. B. Pollack, editors, *Advances in Connectionist and Neural Computation Theory, Vol.1: High Level Connectionist Models*, pages 165–179. Ablex Publishing Corporation, Norwood, NJ, 1991.

[10] V. Honavar and L. Uhr. *Artificial Intelligence and Neural Networks: Steps Toward Principled Integration.* Academic Press, Cambridge, MA, 1994.

[11] R. A. Jacobs, M. I. Jordan, and A. G. Barto. Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. *Cognitive Science*, 15:219–250, 1991.

[12] A. N. Jain. Parsec: A connectionist learning architecture for parsing spoken language. Technical Report CMU-CS-91-208, Carnegie Mellon University, Pittsburgh, PA, 1991.

[13] D. Jurafsky, C. Wooters, G. Tajchman, J. Segal, A. Stolcke, E. Fosler, and N. Morgan. The Berkeley Restaurant Project. In *Proceedings of the International Conference on Speech and Language Processing*, pages 2139–2142, Yokohama, 1994.

[14] S. C. Kwasny and K. A. Faisal. Connectionism and determinism in a syntactic parser. In N. Sharkey, editor, *Connectionist natural language processing*, pages 119–162. Lawrence Erlbaum, 1992.

[15] R. Maclin. *Learning from Instruction and Experience: Methods for Incorporating Procedural Domain Theories into Knowledge-Based Neural Networks.* PhD thesis, Department of Computer Sciences, University of Wisconsin-Madison, 1995. (Also appears as UW CS-TR-95-1285).

[16] L. R. Medsker. *Hybrid Intelligent Systems.* Kluwer Academic Publishers, Boston, 1995.

[17] R. Miikkulainen. *Subsymbolic Natural Language Processing.* MIT Press, Cambridge, MA, 1993.

[18] M. C. Mozer and S. Das. A connectionist symbol manipulator that discovers the structure of context-free languages. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 863–870. Morgan Kaufmann, San Mateo, CA, 1993.

[19] C. W. Omlin and C. L. Giles. Extraction of rules from discrete-time recurrent neural networks. *Neural Networks*, 9(1):41–52, 1996.

[20] J. B. Pollack. On connectionist models of natural language processing. Technical Report PhD thesis, Technical Report MCCS-87-100, New Mexico State University, Las Cruces, NM, 1987.

[21] R. G. Reilly and N. E. Sharkey. *Connectionist Approaches to Natural Language Processing*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1992.

[22] L. Shastri. A model of rapid memory formation in the hippocampal system. In *Proceedings of the Meeting of the Cognitive Science Society*, Stanford, 1997.

[23] J. Shavlik. A framework for combining symbolic and neural learning. In V. Honavar and L. Uhr, editors, *Artificial Intelligence and Neural Networks: Steps towards principled Integration*, pages 561–580. Academic Press, San Diego, 1994.

[24] M. F. St. John and J. L. McClelland. Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, 46:217–257, 1990.

[25] R. Sun. *Integrating Rules and Connectionism for Robust Commonsense Reasoning*. Wiley, New York, 1994.

[26] R. Sun and L.A. Bookman. *Computational Architectures Integrating Neural and Symbolic Processes*. Kluwer Academic Publishers, Boston, MA, 1995.

[27] S. Wermter. *Hybrid Connectionist Natural Language Processing*. Chapman and Hall, Thomson International, London, UK, 1995.

[28] S. Wermter and M. Löchel. Learning dialog act processing. In *Proceedings of the International Conference on Computational Linguistics*, pages 740–745, Kopenhagen, Denmark, 1996.

[29] S. Wermter and M. Meurer. Building lexical representations dynamically using artificial neural networks. In *Proceedings of the International Conference of the Cognitive Science Society*, pages 802–807, Stanford, 1997.

[30] S. Wermter, E. Riloff, and G. Scheler. *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*. Springer, Berlin, 1996.

[31] S. Wermter and V. Weber. SCREEN: Learning a flat syntactic and semantic spoken language analysis using artificial neural networks. *Journal of Artificial Intelligence Research*, 6(1):35–85, 1997.

[32] A. Wilson and J. Hendler. Linking symbolic and subsymbolic computing. *Connection Science*, 5:395–414, 1993.