

- 13 Chomsky, N. (1986) *Knowledge of Language*, Praeger
- 14 Chomsky, N. (1973) Conditions on transformations. In *A Festschrift for Morris Halle* (Anderson, S. and Kiparsky, P., eds), pp. 232–286, Holt, Rinehart & Winston
- 15 Chomsky, N. (1995) Categories and transformations. In *The Minimalist Program*, pp. 219–394, MIT Press
- 16 Collins, C. (1997) *Local Economy*, MIT Press
- 17 Chomsky, N. (2000) Minimalist inquiries: the framework. In *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik* (Martin, R. et al., eds), pp. 89–155, MIT Press
- 18 Collins, C. (2001) Economy conditions in syntax. In *The Handbook of Contemporary Syntactic Theory* (Baltin, M. and Collins, C., eds), pp. 45–61, Oxford University Press
- 19 Lasnik, H. (1995) Case and expletives revisited: on Greed and other human failings. *Linguist. Inq.* 26, 615–633
- 20 Huang, C.-T.J. ((1981) 82) Move *wh* in a language without *wh*-movement. *Linguist. Rev.* 1, 369–416
- 21 Ross, J.R. (1967) *Constraints on Variables in Syntax*, Doctoral dissertation MIT, Cambridge, MA, USA; published 1986 as *Infinite Syntax!*, Ablex
- 22 Chomsky, N. (1993) A minimalist program for linguistic theory. In *The View from Building 20: Essays in Linguistics in Honor of Sylvain Bromberger* (Hale, K. and Keyser, S.J., eds), pp. 1–52, MIT Press
- 23 Lasnik, H. (1997) On feature strength: three minimalist approaches to overt movement. *Linguist. Inq.* 30, 197–217
- 24 Chomsky, N. (2001) Derivation by phase. In *Ken Hale: A Life in Language* (Kenstowicz, M., ed.), pp. 1–52, MIT Press
- 25 Bresnan, J.W. (1971) Sentence stress and syntactic transformations. *Language* 47, 257–281
- 26 Jackendoff, R. (1972) *Semantic Interpretation in Generative Grammar*, MIT Press
- 27 Epstein, S.D. (1999) Un-principled syntax: the derivation of syntactic relations. In *Working Minimalism* (Epstein, S.D. and Hornstein, N., eds), pp. 317–345, MIT Press
- 28 Uriagereka, J. (1999) Multiple spell-out. In *Working Minimalism* (Epstein, S.D. and Hornstein, N., eds), pp. 251–282, MIT Press
- 29 Boskovic, Z. (2001) *On the Nature of the Syntax–Phonology Interface: Cliticization and Related Phenomena*, Elsevier Science
- 30 Fox, D. (2000) *Economy and Semantic Interpretation*, MIT Press
- 31 Kayne, R.S. (1994) *The Antisymmetry of Syntax*, MIT Press

Connectionist natural language parsing

Dominic Palmer-Brown, Jonathan A. Tepper and Heather M. Powell

The key developments of two decades of connectionist parsing are reviewed. Connectionist parsers are assessed according to their ability to learn to represent syntactic structures from examples automatically, without being presented with symbolic grammar rules. This review also considers the extent to which connectionist parsers offer computational models of human sentence processing and provide plausible accounts of psycholinguistic data. In considering these issues, special attention is paid to the level of realism, the nature of the modularity, and the type of processing that is to be found in a wide range of parsers.

Dominic Palmer-Brown*
Leeds Metropolitan
University,
Computational
Intelligence Research
Group, School of
Computing, Beckett Park,
Leeds, UK, LS6 3QS.
*e-mail: d.palmer-brown@
lmu.ac.uk

Jonathan A. Tepper
Heather M. Powell
The Nottingham Trent
University, Department of
Computing &
Mathematics, Burton
Street, Nottingham,
UK NG1 4BU.
e-mail: jonathan.tepper@
ntu.ac.uk;
heather.powell@ntu.ac.uk

Connectionist parsers are neural-network-based systems (see Boxes 1 and 2) designed to process words or their syntactic types (tags) to produce a correct syntactic interpretation, or parse, of complete sentences. Parsers vary greatly in the way in which they tackle syntactic processing, and this is reflected in their modularity (or non-modularity) and in whether they combine neural networks with conventional symbolic processing to provide a hybrid solution, or adopt a purely connectionist approach.

Modularity and hybridity are reviewed as key attributes of connectionist parsers concerned with how the parsing problem is decomposed into (usually simpler) modules to form a parsing system, consisting of one or more connectionist modules and zero or more non-connectionist (e.g. symbolic) modules. The level of realism of parsers is assessed, by which we mean the ability of connectionist parsing systems to capture naturally occurring linguistic structures, behaviours and processing limitations. We discuss the extent to which parsers are able to capture the syntactic constraints and structures that naturally occur in

language, as opposed to being limited to artificial grammars that restrict them to processing very small sub-domains of the language.

Modularity and hybridity

Psycholinguistic and fMRI-based evidence suggests that there is a significant component of purely syntactic processing of language that precedes and is independent of semantic processing [1]. This evidence can be taken as support for a syntactic module in a Fodorian sense [2], in that it is consistent with information encapsulation and spatial separation of processes. However, there is little consensus in terms of the details of how the syntactic module and its processing might be decomposed into separate modules performing distinct tasks. Within the connectionist research programme, many versions of modular architecture have been proposed, and as the neurocognitive evidence is still unclear, connectionists are free to explore the computational plausibility of different architectures. When a proposed architecture shows human-like performance on some aspect of syntax, it can be claimed as evidence of cognitive plausibility, but thus far no systems have been extended to a truly convincing range of language structures. What is clear, however, is that modular and hybrid parsers trained on corpora continue to make significant progress. Indeed, for large scale parsing, although there is little support for the eliminative connectionist viewpoint, which claims that purely connectionist systems are in principle capable of cognitive functions including language

Box 1. Artificial neural network architectures

Multi-layer perceptron (MLP)

In artificial neural networks (ANN), each node (or neuron) calculates the level of a feature that is present on its inputs, which comes either from the outside world or from other nodes. The particular feature is usually dependent on the weights on the inputs, and these weights can be adapted during a training phase, so that the feature represented by each node can be learned experientially, rather than hardwired in advance.

MLPs are the most commonly used and general purpose ANN architectures. They have an input layer, one or more hidden layers, and an output layer (see Fig. I). The standard MLP has complete connectivity between all the nodes of one layer and the nodes of the next. Each input to a node is multiplied by a weight associated with the connection. The output of a node is the sum of these weighted inputs after it has been put through a transfer function. The transfer function is usually non-linear (the sigmoid function is the most commonly used). This allows the network to model non-linear relationships between the inputs and outputs. A common form of training involves propagating the difference between the desired and actual outputs for a given input (i.e. the output error), back through the network, adjusting the weights so as to reduce this error. This is known as back-propagation and is done repeatedly for a set of training input-output pairs.

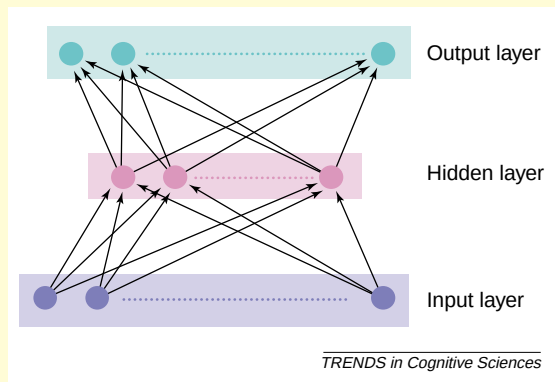


Fig. I. MLP architecture with one hidden layer. (See text for details.)

Simple recurrent network (SRN) [a]

The more specialized SRN architecture is aimed at processing inputs that consist of sequences of patterns of variable length. It does this by including in the input the activations of the hidden layer for the previous step in the sequence (see Fig. II). This allows the network to include

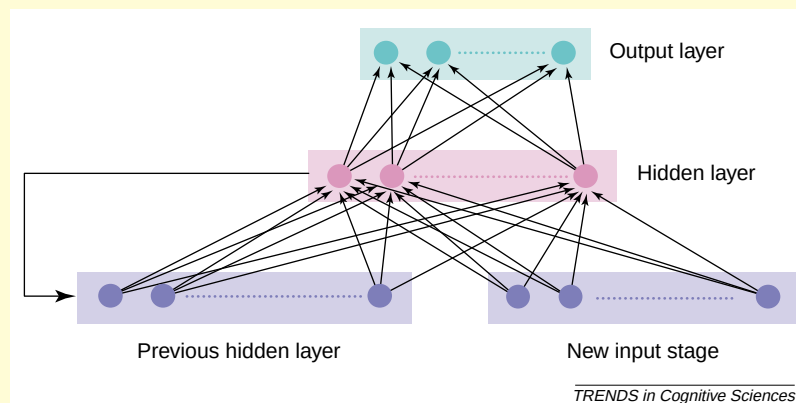


Fig. II. SRN architecture.

information relating to all the previous steps in a sequence in its processing of the current stage. The network 'remembers' what has gone before, forgetting only gradually as it progresses through the sequence. To do the same in an MLP, the input window would have to accommodate, instantaneously, all of the input patterns for the longest sequence. However, even in an SRN, successful training can be difficult to achieve because of significant variation in the length of sequences that it is required to learn.

Recursive Autoassociation Memory (RAAM) [b]

An autoassociative network is an ANN trained to generate the input activations at the output. If this is achieved with an MLP having one hidden layer, the hidden layer activations for a given input will then become an alternative representation of that input. Such a network can be used to encode and decode compressed representations of symbolic structures such as trees, and it is then known as a recursive autoassociation memory, or RAAM. The compressed representation generated at the output for two daughter nodes of a tree, for example, becomes the input representation of their mother node when the representation for the next stage of the tree is generated, hence the term 'recursive'.

Localist networks

In a localist network (as opposed to a distributed one; see Box 2), the nodes are localist in nature – that is, excitation on a given node indicates the presence of the corresponding linguistic structure. Localist networks for parsing usually incorporate 'spreading activation' and 'lateral inhibition'; that is, they support bidirectional excitatory connections between nodes in adjacent layers, and inhibitory connections between nodes within a layer (and between layers) (see Fig. III). This allows the node activations to settle into mutually consistent patterns. The exact connectivity of a given network is specific to the application, and is often designed to support a particular grammatical formalism.

References

- a Elman, J. (1990) Finding structure in time. *Cogn. Sci.* 14, 179–211
- b Pollack, J. (1990) Recursive distributed representations. *Artif. Intell.* 46, 77–105

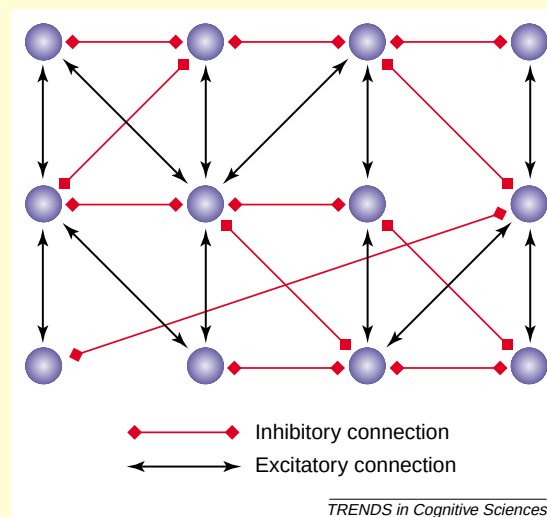


Fig. III. An example localist architecture

processing, there is some support for those such as Pinker and Marcus [3], who point out that essentially symbolic processing devices such as registers need to be combined with neural networks in systems such as parsers.

Singular architectures

A common, singular (non-modular) localist (see Box 2) approach has been to represent the syntactic structure (or parse tree) of a sentence explicitly by patterns of activation between layers of units. The units represent

Box 2. Localist and distributed representations

ANNs are said to represent specific concepts either in a localist way or in a distributed way. In a localist representation there is a one-to-one mapping of concepts or features onto nodes. For example, Fig. 1a shows a localist representation of the vowels: one vowel maps onto one and only one feature. In a distributed representation (Fig. 1b) the representations for several vowels are determined by levels of activation (indicated here by variation in grey level) across common nodes.

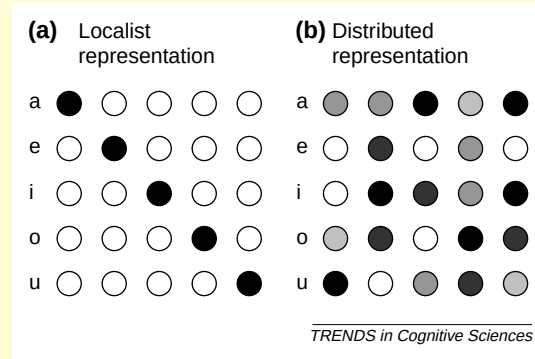


Fig. 1. The vowels represented in (a) localist and (b) distributed forms. (See text for details.)

symbols or concepts drawn from either symbolic grammar rules [4–7], word senses, syntactic roles or semantic roles [8–10]. Many distributed parsing models employing a single neural network have also been developed, using either a feedforward multi-layer perceptron (FF-MLP) network [11,12], a simple recurrent network (SRN) [13,14] or a fully recurrent network [15] (see Box 1). To process linguistic input, which is sequential by nature, FF-MLPs typically require a temporal (or sliding) input window of sufficient width (i.e. with a sufficient number of input units) to process the largest grouping of constituent symbols that could signify a parser action. This approach is inefficient as the window width is defined by the worst (longest) case, and for all other cases there is redundancy involved in processing the entire window. These limitations were reflected in early distributed connectionist parsers with temporal input windows [11,12].

The simple recurrent network (SRN)

SRN-based architectures (Box 1) belong to the class of recurrent neural networks (RNNs) that implement temporal processing with feedback connections which allow them to establish temporal relationships across sequential input items. SRNs are computationally more powerful than FF-MLP networks for automatically inducing grammatical constraints and structure through supervised learning [13,14,16]. For example, SRNs (and RNNs generally) are able to learn regular languages by forming a continuously-valued representation of a corresponding discrete finite-state machine (FSM) when trained to predict the next symbol (e.g. word) within a sequence (e.g. sentence) [13,17]. However, it is generally accepted that natural languages

are not strictly regular languages but actually lie between the classes of context-free and context-sensitive languages [18]. This would suggest that additional *external* apparatus, such as symbolic or neural stacks or registers, are required to extend SRNs to non-regular languages that support embedded and context-sensitive structures. Some have indeed used SRNs with external stacks for language parsing [19–24], although others have used alternative RNNs with external stack memories [25].

Rather than introducing additional apparatus to process natural language, several researchers have focused their attention on using dynamical systems theory [26] to understand and interpret the representations formed and dynamic behaviours (transitions made between hidden states) exhibited by RNNs when they are expected to process irregular languages. Such studies have shown that SRNs learn simple counting functions to process deterministic context-free languages [27]. More recently it has been shown that SRNs are able to generalize to a subset of context-sensitive languages beyond the training data [28], however performance degrades rapidly with respect to the length of the input. Although these studies also show that the behaviour of SRNs is unstable with respect to complexities associated with irregular languages, SRNs *are* able to generalize to structures considered beyond the computational power of an FSM and the memory limitations exhibited have been shown to be comparable to human memory limitations during sentence processing.

Modules

The first modular distributed parsers appeared in the early 1990s and typically consisted of combinations of FF-MLP, SRN and RAAM architectures. The motivation to decompose the parsing task into sub-modules is often: (1) to simplify the network's learning task, and to reduce training set size and complexity; and/or (2) to evaluate the computational and cognitive plausibility of a given composition of modules. The wide range of modules that has been tried reflects the respective strengths and weaknesses of FF-MLPs, SRNs, and RAAMs. FF-MLPs are general-purpose pattern recognizers and function approximators, and therefore it is unsurprising that they have been used to compress [20], transform [29] and compare patterns representing words, phrases, clauses, and semantic and syntactic roles [30]. SRNs, by virtue of sequential input and feedback to context nodes, are more suited than FF-MLPs to recognizing a sequence or identifying an action based on a sequence, where the sequence is presented incrementally through time. Thus, SRNs lend themselves naturally to segmenting sentences into case-roles [21,31] and phrases [30,32], to signifying shift-reduce actions [22], and to recognizing phrases through time [23,24,30]. RAAMs, because of their autoassociative coding and decoding properties, have been used to encode (storing them in a compressed form) and decode symbolic parse trees

and phrases [21,23,34,35]. Neural networks developments in the form of FF-MLPs, SRNs and RAAMs, and the learning processes they use have strongly influenced the modular decomposition of connectionist parsing.

Hybridity

Only pure connectionist parsers such as those of [21,34,36] attempt every aspect of the parse with neural networks and perform no symbol manipulation other than to interpret system outputs. When symbolic modules or processes are included, the result is a hybrid connectionist parser. In hybrid parsers the symbolic modules have commonly been assigned to tasks such as: (1) short-term storage to hold intermediate parse states, current input states and full sentential parses (stacks and buffers); (2) long-term storage to hold structured knowledge about language, such as grammar rules, semantic networks, and tree structures; and (3) symbol manipulation and communication to control the parsing process and coordinate interactions between (connectionist) modules. Pure connectionist parsers have attempted to avoid using symbolic modules by implementing connectionist architectures to act as stack memories and associative memories.

Realism

Convergence of psycholinguistic studies and connectionist parsing

A dominant theory of Human Sentence Processing (HSP) is a modular two-stage model [37] in which an independent processor provides an initial syntactic structure using only syntactic information. A second processing phase allows semantic and discourse information to be used to enable the parser to revise the initial structure. Lexical and syntactic processing are considered to be separate because it is hypothesized that lexical representations are retrieved whereas syntactic representations are constructed [38]. However, there is an increasing body of research that rejects this view and argues that HSP is a multiple constraint-satisfaction process that allows syntactic, semantic and discourse information to simultaneously interact (to varying degrees) during on-line processing [39,40,41]. The psycholinguistic evidence suggests that syntactic structure is projected from a mental lexicon enriched with word representations containing syntactic, semantic, and discourse information together with the frequency and probabilistic relationships that hold amongst them. Syntactically ambiguous sentences are commonplace. In the sentence, '*Gill saw the man with the telescope*', we don't know whether the man or Gill has the telescope, and the two possibilities are syntactically distinct, suggesting the need for semantic intervention in the parsing process. Furthermore, psychologists have for decades accumulated evidence for the emergence of tentative interpretations of sentences occurring before a definitive parse [42].

Connectionist models can be employed as the processing mechanisms to develop and evaluate constraint-satisfaction theories [39,43] and there is some evidence that connectionist research into syntactic processing is beginning to converge with psycholinguistic theories. Early localist parsing models were able to provide some account of human syntactic and semantic attachment preferences, whereby alternative interpretations remain active until they become incompatible with the rest of the sentence [8,9,10]. However, these techniques were based on a limited set of grammar rules. Stevenson's CAPERS model [44] extended localist parsing techniques further using a symbolic module, based on the principles defined in Chomsky's Government and Binding (GB) Theory [45]; activation representing grammatical constraints is communicated between neighbouring units within a localist network that settles to a steady state of activations explicitly representing a parse tree. Stevenson demonstrated that CAPERS is able to establish long-distance dependencies between constituents without specially designed heuristics and addresses some of the psycholinguistic data reported by Stowe [46] and Nicol (J. Nicol, Ph.D thesis, MIT, Boston, MA, 1988). More recently, Stevenson and Merlo have used CAPERS to show that differences between unergative verbs (e.g. '*the horse raced past the barn fell*') and unaccusative verbs (e.g. '*the butter melted in the pan was brown*') at the lexical level significantly influence processing difficulty of sentences with ambiguities.

Although perhaps not directly motivated by the constraint-based theories per se, Hadley *et al.*'s [47] localist model supports the interaction of syntactic and semantic information. They proposed a self-organising, Hebbian-inspired, competitive network that was able to predict the coarse semantic features of the next word within a sentence for a small language domain. It was also able to predict syntactically key words, such as prepositions and relative pronouns. The model is significant because it generalised to structures of a complexity beyond that found in the training data and demonstrated strong syntactic systematicity in that it was able to recognize noun phrases in new (untrained) syntactic positions.

Connectionists have also exploited the advantages of distributed representations. St John and McClelland presented a modular connectionist account of the multiple constraint-based model using two MLP-based networks [48]. The model assigns thematic roles (e.g. agent and action) to predetermined syntactic phrases (constituents) of simple single-clause sentences based on syntactic and semantic constraints. It builds a single distributed representation of the sentence, called the 'sentence gestalt', which is adjusted as each constituent is processed. Given a word from a sentence it is able to reproduce the thematic role assigned to that word. Although evaluated using a very limited language, it was able to anticipate future constituents, and also learnt semantic regularities about the age of

agents to predict appropriate objects. The main limitation of this approach is the inefficient localist representation used on input/output layers, which provides little promise of scaling-up to work on realistic language sources.

Pearlmutter *et al.* [49] proposed an FF-MLP-based model directly inspired by the constraint-based framework presented by MacDonald *et al.* [40,43]. The localist input representation for nouns contained information such as subject animacy, voice and presence of a direct object. Verb representations contained information such as a unique identifier for the verb and a set of simple semantic features. Given this input on the input layer the FF-MLP was expected to produce the appropriate verb argument structure on the output layer. After being trained on a set of 60 unambiguous verbs and associated argument structures extracted from the Wall Street Journal corpus, the FF-MLP was tested on a modified training set. The training set was made ambiguous by removing the influence of voice and direct object inputs. The resulting behaviour of the system provided strong support for constraint-based theories. Pearlmutter *et al.* reported that the frequency of argument structures found within the corpus directly influenced the processing of verb ambiguities.

Realism beyond context-free grammar

Modular connectionist parsers that do not rely on a connectionist module such as RAAM to encode explicit structure have been able to capture realistic natural language structures from natural language corpora annotated with syntactic information. These parsers, for example those of Wermter and Weber [30], Lane and Henderson [19], and Tepper, Powell and Palmer-Brown [32,33] induce implicit grammars (that need never be expressed in the form of symbolic rules) via learning the underlying linguistic constraints contained in the annotations of corpus texts. To achieve this, connectionists have focused on using SRNs to perform tasks such as phrase boundary identification [31,32,33] and recognition of constituent dependencies [19], and on FF-MLPs to perform tasks such as phrase structure classification [32,33]. In these hybrid models, the symbols representing syntactic structure are represented by patterns which are processed and recognized by connectionist modules.

Embedding and cross-serial dependencies

SRNs can process cross-serial dependencies (in which the correct interpretation of the syntactic

role of a word or phrase is dependent on that of another word or phrase separated from it by several words) when there are local dependencies between all the intervening words [13], and SRNs can also carry information over embeddings without local dependencies if the number of intervening words was small [14]. In addition, Christiansen and Chater [50] have demonstrated that an SRN is able to process centre-embedding and cross serial dependencies (albeit based on simple context-free grammars) in a manner analogous to the human performance found in psycholinguistic data, in which the ability to process embeddings and serial dependencies is limited. Miikkulainen, whose parser, CLAUSES [31], used a combination of four SRNs, found it easier to process tail recursion (e.g. 'The woman blamed the man, who hit the girl, who blamed the boy') than relative clauses located in the middle of a sentence (e.g. 'The woman, who the boy, who the girl blamed hit, blamed the man') because the latter case requires longer sequences thus taxing the memory capacity further. The memory degradation in SRNs is due to previous input information being degraded and ultimately lost as more input is recursively encoded; this type of degradation also applies to RAAMs [51], although the level of embedding that a RAAM is capable of encoding can be reliably determined [52].

Conclusion

Connectionism is sometimes characterized as a unified approach to computation and cognition, with obvious strengths, such as the ability to acquire new information, and obvious weaknesses, such as limited recursive capabilities, and lack of systematicity. However, a review of connectionist parsing demonstrates that there are several related approaches, ranging from modular systems that mix symbolic processing with neural network processing, to systems that consist entirely of one or more neural networks and that learn to process syntax entirely from examples. Connectionist parsing is still negotiating its relationship with the world of symbols, but the current evidence supports a hybrid approach. The continued improvement in performance of connectionist parsers seems likely to be dependent on the discovery or development of more powerful neural network learning processes and architectures, that can effectively assimilate the vast quantity of complex information associated with human language processing ability.

References

- 1 Friederici, A.D. (2002) Towards a neural basis of auditory sentence processing. *Trends Cogn. Sci.* 6, 78–84
- 2 Fodor, J.A. (1983) *Modularity of Mind*, MIT Press
- 3 Marcus, G. (2001) *The Algebraic Mind: Integrating Connectionism and Cognitive Science*, MIT Press
- 4 Selman, B. and Hirst, G. (1985) A rule-based connectionist parsing system. In *Proc. 7th Annu. Conf. Cogn. Sci. Soc.*, pp. 212–221, Erlbaum
- 5 Fanty, M.A. (1986) Context-free parsing with connectionist networks. In *Proc. AIP Conf. on Neural Networks for Computers*, pp. 140–145, Springer
- 6 Howells, T. (1988) VITAL: a connectionist parser. In *Proc. 10th Annu. Conf. Cogn. Sci. Soc.*, pp. 18–25, Erlbaum
- 7 Santos, E. (1989) A massively parallel self-tuning context-free parser. *Adv. Neural Inf. Process. Syst.* 1, 537–544
- 8 Cottrell, G.W. and Small, S. (1983) A connectionist scheme for modeling word-sense disambiguation. *Cogn. Brain Theory* 6, 89–120
- 9 Waltz, D. and Pollack, J. (1985) Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cogn. Sci.* 9, 51–74
- 10 Cottrell, G.W. (1985) Connectionist parsing. In *Proc. 7th Annu. Conf. Cogn. Sci. Soc.*, pp. 201–211, Erlbaum

- 11 Kwansy, S.C. and Faisal, K. (1990) Connectionism and determinism in a syntactic parser. *Connect. Sci.* 2, 63–82
- 12 Tepper, J.A. *et al.* (1995) Integrating symbolic and subsymbolic architectures for parsing arithmetic expressions and natural language sentences. In *Proc. 3rd Annu. Symp. Neural Netw.*, pp. 81–84, Springer
- 13 Cleeremans, A. *et al.* (1989) Finite state automata and simple recurrent networks. *Neural Comput.* 1, 372–381
- 14 Elman, J.L. (1991) Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learn.* 7, 195–224
- 15 Christiansen, M.H. and Chater, N. (1999) Connectionist natural language processing: the state of the art. *Cogn. Sci.* 23, 417–437
- 16 Lawrence, S. *et al.* (2000) Natural language grammatical inference with recurrent neural networks. *IEEE Trans. Knowl. Data Eng.* 12, 126–1400
- 17 Servan-Schreiber, D. *et al.* (1991) Graded state machines: The representation of temporal contingencies in simple recurrent networks. *Machine Learn.* 7, 161–193
- 18 Sells, P. *et al.* (1991) *Foundational Issues in Natural Language Processing*, MIT Press
- 19 Lane, P. and Henderson, J. (1998) Simple synchrony networks: learning to parse natural language with temporal synchrony variable binding. In *Proc. 8th Int. Conf. Artif. Neural Netw.*, pp. 615–620, Springer
- 20 Moisil, H. (1992) Connectionist finite state language processing. *Connect. Sci.* 4, 67–91
- 21 Miikkulainen, R. (1995) Subsymbolic parsing of embedded structures. In *Computational Architectures Integrating Neural and Symbolic Processes*, (Sun, R., ed), pp. 153–186, Kluwer Academic Publishers
- 22 Mayberry, M.R. and Miikkulainen, R. (2000) Combining maps and distributed representations for shift-reduce parsing. In *Hybrid Neural Systems* (Wermter, S. and Sun, R., eds.), pp. 144–157, Springer-Verlag
- 23 Reilly, R. (1992) Connectionist technique for on-line parsing. *Neural Netw.* 3, 37–46
- 24 Berg, G. (1992) A connectionist parser with recursive sentence structure and lexical disambiguation. In *Proc. 10th Natl. Conf. Artif. Intell.*, pp. 32–37, MIT Press
- 25 Zeng, Z. *et al.* (1994) Discrete recurrent neural networks for grammatical inference. *IEEE Trans. Neural Netw.* 5, 320–330
- 26 Martelli, M. (1992) *Discrete Dynamical Systems and Chaos*, CRC Press
- 27 Rodriguez, P. *et al.* (1999) A recurrent neural network that learns to count. *Connect. Sci.* 11, 540
- 28 Wiles, J. *et al.* (2000) Representation beyond finite states: alternatives to push-down automata. In *A Field Guide to Dynamical Recurrent Networks*, (Kolen, J.F. and Kremer, S.C., eds), pp. 129–142, IEEE Press
- 29 Sharkey, A. and Sharkey, N. (1992) Connectionism and natural language. In *Connectionist Natural Language Processing* (Part 20) (Sharkey, N., ed.), pp. 1–10, Intellect
- 30 Wermter, S. and Weber, V. (1994) Learning fault-tolerant speech parsing with SCREEN. In *Proc. 12th Natl. Conf. Artif. Intell.*, pp. 670–675, MIT Press
- 31 Miikkulainen, R. (1990) A PDP architecture for processing sentences with relative clauses. *Proc. 13th Intl. Conf. Comput. Linguist.*, pp. 201–206, MIT Press
- 32 Tepper, J.A. *et al.* (2001) Corpus-based connectionist parsing. In *The Second Workshop on Natural Language Processing and Neural Networks (NLPNN2001)*, pp. 8–15, National Centre for Science, Tokyo
- 33 Tepper, J.A. *et al.* (in press) A corpus-based connectionist architecture for large-scale natural language parsing. *Connect. Sci.*
- 34 Ho, E.K.S. and Chan, L.W. (1997) Confluent preorder parsing of deterministic grammars. *Connect. Sci.* 9, 269–293
- 35 Sharkey, N. and Sharkey, A. (1992) A modular design for connectionist parsing. In *Twente Workshop on Language Technology 3: Connectionism and Natural Language Processing*, pp. 87–96, University of Twente, Enschede, The Netherlands
- 36 Giles, C.L. *et al.* (1993) Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Comput.* 4, 393–405
- 37 Frazier, L. (1987) Sentence processing: a tutorial review. In *Attention and Performances XII: The Psychology of Reading* (Coltheart, M., ed.), pp. 559–586, Erlbaum
- 38 Rayner, K. and Morris, R. (1992) Eye movement control in reading: evidence against semantic preprocessing. *J. Exp. Psychol. Hum. Percept. Perform.* 18, 163–172
- 39 Taraban, R. and McClelland, J.L. (1990) Parsing and comprehension: a multiple constraint view. In *Comprehension Processes in Reading*, (Balota, D.A. *et al.*, eds), pp. 231–263, Erlbaum
- 40 MacDonald, M. *et al.* (1994) Syntactic ambiguity resolution as lexical ambiguity resolution. In *Perspectives on Sentence Processing* (Clifton, C. *et al.*, eds), pp. 123–153, Erlbaum
- 41 Tanenhaus, M. and Trueswell, J. (1995) Sentence comprehension. In *Speech, Language, and Communication*, (Miller, J. and Eimas, P., eds), pp. 217–262, Academic Press
- 42 Greene, J. (1986) *Language Understanding*, Open University Press
- 43 McDonald, M.C. *et al.* (1994) The lexical nature of syntactic ambiguity resolution. *Psychol. Rev.* 101, 676–703
- 44 Stevenson, S. (1994) Competition and recency in a hybrid network model of syntactic disambiguation. *J. Psycholing. Res.* 23, 295–322
- 45 Chomsky, N. (1981) *Lectures on Government and Binding*. The Pisa Lectures, Foris, Dordrecht
- 46 Stowe, L.A. (1986) Parsing WH-constructions: evidence for on-line gap location. *Lang. Cogn. Processes* 1, 227–245
- 47 Hadley, R.F. *et al.* (2001) Syntactic systematicity arising from semantic predictions in a Hebbian-competitive network. *Connect. Sci.* 13, 73–94
- 48 St. John, M.F. and McClelland, J.L. (1990) Learning and applying contextual constraints in sentence comprehension. *Artif. Intell.* 46, 217–258
- 49 Pearlmuter, N.J. *et al.* (1994) Modelling the use of frequency and contextual biases in sentence processing. In *Proc. 16th Annu. Conf. Cogn. Sci. Soc.*, pp. 699–704, Erlbaum
- 50 Christiansen, M.H. and Chater, N. (1999) Toward a connectionist model of recursion in human linguistic performance. *Cogn. Sci.* 23, 157–206
- 51 Pollack, J.B. (1990) Recursive Distributed Representations. *Artif. Intell.* 46, 77–105
- 52 Callan, R. and Palmer-Brown, D. (1997) (S)RAAM: an analytical technique for fast and reliable derivation of connectionist symbol structure representations. *Connect. Sci.* 9, 139–159

Do you want to reproduce material from *Trends in Cognitive Sciences*?

This publication and the individual contributions within it are protected by the copyright of Elsevier Science. Except as outlined in the terms and conditions (see p. iv), no part of *Trends in Cognitive Sciences* may be reproduced, either in print or electronic form, without written permission from Elsevier Science.

Please address any permission requests to:

Rights and Permissions,
Elsevier Science Ltd,
PO Box 800, Oxford,
UK OX5 1DX