

Middlesex University Research Repository

An open access repository of

Middlesex University research

<http://eprints.mdx.ac.uk>

Huyck, Christian R. ORCID: <https://orcid.org/0000-0003-4015-3549> (2009) A psycholinguistic model of natural language parsing implemented in simulated neurons. *Cognitive Neurodynamics*, 3 (4) . pp. 316-330. ISSN 1871-4080 [Article]
(doi:10.1007/s11571-009-9080-6.)

Published version (with publisher's formatting)

This version is available at: <http://eprints.mdx.ac.uk/4302/>

Copyright:

Middlesex University Research Repository makes the University's research available electronically.

Copyright and moral rights to this work are retained by the author and/or other copyright owners unless otherwise stated. The work is supplied on the understanding that any use for commercial gain is strictly forbidden. A copy may be downloaded for personal, non-commercial, research or study without prior permission and without charge.

Works, including theses and research projects, may not be reproduced in any format or medium, or extensive quotations taken from them, or their content changed in any way, without first obtaining permission in writing from the copyright holder(s). They may not be sold or exploited commercially in any format or medium without the prior written permission of the copyright holder(s).

Full bibliographic details must be given when referring to, or quoting from full items including the author's name, the title of the work, publication details where relevant (place, publisher, date), pagination, and for theses or dissertations the awarding institution, the degree type awarded, and the date of the award.

If you believe that any material held in the repository infringes copyright law, please contact the Repository Team at Middlesex University via the following email address:

eprints@mdx.ac.uk

The item will be removed from the repository while any claim is being investigated.

See also repository copyright: re-use policy: <http://eprints.mdx.ac.uk/policies.html#copy>

Dear Author,

Here are the proofs of your article.

- You can submit your corrections **online** or by **fax**.
- For **online** submission please insert your corrections in the online correction form. Always indicate the line number to which the correction refers.
- Please return your proof together with the **permission to publish** confirmation.
- For **fax** submission, please ensure that your corrections are clearly legible. Use a fine black pen and write the correction in the margin, not too close to the edge of the page.
- Remember to note the journal title, article number, and your name when sending your response via e-mail, fax or regular mail.
- **Check** the metadata sheet to make sure that the header information, especially author names and the corresponding affiliations are correctly shown.
- **Check** the questions that may have arisen during copy editing and insert your answers/ corrections.
- **Check** that the text is complete and that all figures, tables and their legends are included. Also check the accuracy of special characters, equations, and electronic supplementary material if applicable. If necessary refer to the *Edited manuscript*.
- The publication of inaccurate data such as dosages and units can have serious consequences. Please take particular care that all such details are correct.
- Please **do not** make changes that involve only matters of style. We have generally introduced forms that follow the journal's style. Substantial changes in content, e.g., new results, corrected values, title and authorship are not allowed without the approval of the responsible editor. In such a case, please contact the Editorial Office and return his/her consent together with the proof.
- If we do not receive your corrections **within 48 hours**, we will send you a reminder.

Please note

Your article will be published **Online First** approximately one week after receipt of your corrected proofs. This is the **official first publication** citable with the DOI. **Further changes are, therefore, not possible.**

After online publication, subscribers (personal/institutional) to this journal will have access to the complete article via the DOI using the URL: [http://dx.doi.org/\[DOI\]](http://dx.doi.org/[DOI]).

If you would like to know when your article has been published online, take advantage of our free alert service. For registration and further information go to: www.springerlink.com.

Due to the electronic nature of the procedure, the manuscript and the original figures will only be returned to you on special request. When you return your corrections, please inform us, if you would like to have these documents returned.

The **printed version** will follow in a forthcoming issue.

**Fax to: +44 207 806 8278 or +44 870 762 8807 (UK)
or +91 44 4208 9499 (INDIA)**

To: Springer Correction Team

6&7, 5th Street, Radhakrishnan Salai, Chennai, Tamil Nadu, India – 600004

Re: Cognitive Neurodynamics DOI:10.1007/s11571-009-9080-6

A psycholinguistic model of natural language parsing implemented in simulated neurons

Authors: ChristianR. Huyck

Permission to publish

I have checked the proofs of my article and

- I have no corrections. The article is ready to be published without changes.
- I have a few corrections. I am enclosing the following pages:
- I have made many corrections. Enclosed is the complete article.

Date / signature _____

ELECTRONIC REPRINT ORDER FORM

After publication of your journal article, electronic (PDF) reprints may be purchased by arrangement with Springer and Aries Systems Corporation.

The PDF file you will receive will be protected with a copyright system called DocuRights®. Purchasing 50 reprints will enable you to redistribute the PDF file to up to 50 computers. You may distribute your allotted number of PDFs as you wish; for example, you may send it out via e-mail or post it to your website. You will be able to print five (5) copies of your article from each one of the PDF reprints.

Please type or print carefully. Fill out each item completely.

1. Your name: _____
 Your e-mail address: _____
 Your phone number: _____
 Your fax number: _____
2. Journal title (vol, iss, pp): _____
3. Article title: _____
4. Article author(s): _____
5. How many PDF reprints do you want? _____
6. Please refer to the pricing chart below to calculate the cost of your order.

Number of PDF reprints	Cost (in U.S. dollars)
50	\$200
100	\$275
150	\$325
200	\$350

NOTE: Prices shown apply only to orders submitted by individual article authors or editors. Commercial orders must be directed to the Publisher.

All orders must be prepaid. Payments must be made in one of the following forms:

- a check drawn on a U.S. bank
- an international money order
- Visa, MasterCard, or American Express (no other credit cards can be accepted)

PAYMENT (type or print carefully):

Amount of check enclosed: _____ (payable to Aries Systems Corporation)

VISA _____

MasterCard _____

American Express _____

Expiration date: _____ Signature: _____

Print and send this form with payment information to:

Aries Systems Corporation
 200 Sutton Street
 North Andover, Massachusetts 01845
 Attn.: Electronic Reprints
 — OR —
 Fax this to Aries at: 978-975-3811

Your PDF reprint file will be sent to the above e-mail address. If you have any questions about your order, or if you need technical support, please contact: support@docurights.com

For subscriptions and to see all of our other products and services, visit the Springer website at:
<http://www.springeronline.com>

Metadata of the article that will be visualized in OnlineFirst

ArticleTitle	A psycholinguistic model of natural language parsing implemented in simulated neurons	
Article Sub-Title		
Article CopyRight - Year	Springer Science+Business Media B.V. 2009 (This will be the copyright line in the final PDF)	
Journal Name	Cognitive Neurodynamics	
Corresponding Author	Family Name	Huyck
	Particle	
	Given Name	Christian R.
	Suffix	
	Division	
	Organization	Middlesex University
	Address	London, UK
	Email	c.huyck@mdx.ac.uk

Schedule	Received	26 January 2009
	Revised	26 February 2009
	Accepted	26 February 2009

Abstract	<p>A natural language parser implemented entirely in simulated neurons is described. It produces a semantic representation based on frames. It parses solely using simulated fatiguing Leaky Integrate and Fire neurons, that are a relatively accurate biological model that is simulated efficiently. The model works on discrete cycles that simulate 10 ms of biological time, so the parser has a simple mapping to psychological parsing time. Comparisons to human parsing studies show that the parser closely approximates this data. The parser makes use of Cell Assemblies and the semantics of lexical items is represented by overlapping hierarchical Cell Assemblies so that semantically related items share neurons. This semantic encoding is used to resolve prepositional phrase attachment ambiguities encountered during parsing. Consequently, the parser provides a neurally-based cognitive model of parsing.</p>
----------	---

Keywords (separated by '-')	Fatiguing Leaky Integrate and Fire (fLIF) neurons - Natural language parsing - Timing - Prepositional phrase attachment
-----------------------------	---

Footnote Information	
----------------------	--

Journal: 11571
Article: 9080

Author Query Form

**Please ensure you fill out your response to the queries raised below
and return this form along with your corrections**

Dear Author,

During the preparation of your manuscript for typesetting, some questions have arisen. These are listed below. Please check your typeset proof carefully and mark any corrections in the margin of the proof or compile them as a separate list. This form should then be returned with your marked proof/list of corrections to spr_corrections1@sps.co.in

Disk use

In some instances we may be unable to process the electronic file of your article and/or artwork. In that case we have, for efficiency reasons, proceeded by using the hard copy of your manuscript. If this is the case the reasons are indicated below:

- Disk damaged Incompatible file format LaTeX file for non-LaTeX journal
 Virus infected Discrepancies between electronic file and (peer-reviewed, therefore definitive) hard copy
 Other:

We have proceeded as follows:

- Manuscript scanned Manuscript keyed in Artwork scanned
 Files only partly used (parts processed differently:

Bibliography

If discrepancies were noted between the literature list and the text references, the following may apply:

- The references listed below were noted in the text but appear to be missing from your literature list. Please complete the list or remove the references from the text.
 Uncited references: This section comprises references that occur in the reference list but not in the body of the text. Please position each reference in the text or delete it. Any reference not dealt with will be retained in this section.

Queries and/or remarks

Section/paragraph	Details required	Author's response
Affiliation	Please check and confirm the author names and initials are correct. Also, kindly confirm the details in the metadata are correct.	
References	Please check and confirm Author names and initials in references Von der Malsburg 1981, 1986 and van Gompel et al. 2001 (citations as wells in ref. list). Please check and confirm the page number in reference Khan et al. 2008.	

A psycholinguistic model of natural language parsing implemented in simulated neurons

Christian R. Huyck

Received: 26 January 2009 / Revised: 26 February 2009 / Accepted: 26 February 2009
© Springer Science+Business Media B.V. 2009

Abstract A natural language parser implemented entirely in simulated neurons is described. It produces a semantic representation based on frames. It parses solely using simulated fatiguing Leaky Integrate and Fire neurons, that are a relatively accurate biological model that is simulated efficiently. The model works on discrete cycles that simulate 10 ms of biological time, so the parser has a simple mapping to psychological parsing time. Comparisons to human parsing studies show that the parser closely approximates this data. The parser makes use of Cell Assemblies and the semantics of lexical items is represented by overlapping hierarchical Cell Assemblies so that semantically related items share neurons. This semantic encoding is used to resolve prepositional phrase attachment ambiguities encountered during parsing. Consequently, the parser provides a neurally-based cognitive model of parsing.

Keywords Fatiguing Leaky Integrate and Fire (fLIF) neurons · Natural language parsing · Timing · Prepositional phrase attachment

Introduction

In 1988, Smolensky claimed that “neural models of cognitive processes are ... currently not feasible” (Smolensky 1988). This paper describes a neural simulation of a sophisticated, modern cognitive model of parsing which leads to the conclusion that, while Smolensky’s statement may have been true in 1988, it is now possible to model cognitive processes with simulated neurons.

A natural language parsing system implemented entirely in simulated neurons is described. The paper does not describe the full details of the parser, but the code, written in Java, can be found at <http://www.cwa.mdx.ac.uk/CABot/parse4.html>. The parser is a component in the second Cell Assembly Robot (CABot2) agent (see section “CABot”).

While a synapse by synapse, or neuron level, description of the system would be far too long and inappropriate here, a higher level description at the level of the Cell Assemblies (CAs) (see section “Neuropsychology”) that operate by the systematic firing of the simulated neurons is provided. This description and simulation evidence shows that the parser meets the following four goals:

1. *The system parses in a manner that is linguistically, cognitively and neurally plausible* While linguists do not agree on all aspects of language, there is broad agreement on some areas, and the parser should be consistent with these areas of agreement. In other linguistic areas there is not agreement, and in this case the parser should be consistent with at least one theory. There are a range of cognitive phenomena that could be modelled; the parser does not need to account for all of them, but should account for some of the important ones. There are a range of neural models, but there is also a trade-off between level of detail and the speed of the simulation. The mapping between the simulated neural model and biological neural topology needs to be reasonably accurate, but the simulation needs to be efficient. Where neural mapping inaccuracies are imposed, due to, for instance, the number of neurons that can be simulated in real-time, then there should be a path to duplicating performance, in this instance, if there were more neurons in the simulation. It is

A1 C. R. Huyck (✉)
A2 Middlesex University, London, UK
A3 e-mail: c.huyck@mdx.ac.uk

hypothesised that similarity in the substrate that supports cognition and language in human and AI systems will directly improve the latter's parsing capabilities, i.e. making performance closer to that of people.

2. *The system resolves Prepositional Phrase (PP) attachment ambiguity* PP attachment ambiguity is a difficult problem for natural language parsing. Example 1 is a commonly used sentence with a PP attachment ambiguity.

(example 1) I saw the girl with the telescope.

The PP *with the telescope* can be attached to (modify) the verb *saw* so that it is an instrument, or can attach to the noun phrase *the girl* so that she has it. Resolving PP attachment ambiguity is important because it is one of the many instances of semantics being needed to resolve syntactic decisions. In example 1, attachment to *saw* is more probable since telescopes are normally used for seeing, but replacing *tool* for *telescope* might shift the attachment to the noun phrase *the girl*.

3. *The system parses relatively effectively* The system must parse a reasonable subset of English to be convincing, and it must be clear how it could scale up in a relatively straightforward manner. Similarly, the system is a part of the functioning CABot2 agent and must be effective for it.
4. *All of the above involve semantics, so the system must represent semantics in a reasonable fashion* For a system to handle natural language effectively, it must deal with its combinatorial nature. It must be able to cope with a practically unlimited number of sentences, and, in particular, generate a different semantic representation for sentences that do have different meanings.

The remainder of the paper is broken into sections, starting with its background and related work. This is followed by a section on the neural model, and then a section on the CABot2 parser itself. These are followed by a section on the empirical results of the parser on the test materials. The results show that it correctly parses, in times similar to those found in human subjects, and that it resolves PP attachment ambiguities correctly. The paper concludes with a discussion of the quality of the CABot2 parser as a functioning system and as a cognitive model, and how it can be improved.

Background

The over-arching long-term goal for research in this area is to develop an AI system that is capable of understanding and producing language at a level that is at or near the level

of an adult human. It is hypothesised that the best way to do this is to develop a model that behaves in a fashion that is both psychologically and neurally close to that of the human one. For such a system to succeed, however, it must have an understanding of semantics that is similar to a human's. It must, among many other things, ground symbols (Harnad 1990), have sensory input, and function in an environment. As a fully intelligent system is a huge goal, this paper describes a system that starts to solve a particular subgoal.

The particular subgoal is to develop a parser based on neurons that parses in a neurally and psycholinguistically plausible manner. Moreover, as this parser is a component in a working agent (Huyck 2008), it must be effective, efficient to simulate, and able to work with other subsystems.

An earlier version of the system (Huyck and Fan 2007) was based on a stack and was used for CABot1, the first version of the CABot agent. Unfortunately, the dynamics of this earlier system led it to spending a large amount of time managing the stack (see section "The binding problem"). Moreover, the simulation time was too long and the putative biological time of the CABot1 parser was also well beyond that of human parsing.

Consequently, a stackless parser was developed for CABot2. This is similar to a range of psycholinguistic parsers including one (Lewis and Vasishth 2005) based on the ACT-R system. Nonetheless, this stackless parser still had to account for a traditional problem of neural parsing systems, the variable binding problem.

The binding problem

The binding problem (Von Der Malsburg 1986) needs to be resolved to allow compositional semantics and syntax (Fodor and Pylyshyn 1988). For instance, a standard mechanism for representing the semantics of a sentence is a case frame representation (Filmore 1968), where a sentence like example 2 is represented by the head verb *see* and two slot filler pairs, the actor *I*, and the object *the girl*. However, the slots need to be bound to the appropriate filler for successful sentence parsing. For example, the object slot would need to be filled by *the boy* in example 3.

(example 2) I saw the girl.

(example 3) I saw the boy.

Binding is simple for symbolic systems, because a variable can easily be given a value and subsequently have that value replaced. It is a basic operation on all standard computers.

166 A range of non-neural connectionist binding mecha-
167 nisms also exist. Tensor product binding has been used
168 (Smolensky 1990). Recurrent multilayer perceptrons
169 learning via backpropagation (Mikkulainen 1993) have
170 also been used.

171 The most commonly used mechanism for binding in
172 neural simulations is binding by synchrony (Von Der
173 Malsburg 1981), where bound neurons fire with a similar
174 oscillatory pattern. Another option is binding by active
175 links (van der Velde and de Kamps 2006), where special
176 reusable circuits are developed to bind items.

177 In related work (Huyck and Belavkin 2006), a system was
178 developed that bound via Long-Term Potentiation (LTP).
179 However, this interfered with other learning, leading to
180 the stability-plasticity dilemma (Carpenter and Grossberg
181 1988). This dilemma is the ability of a neural system to learn
182 new information, while retaining the appropriate older
183 information.

184 As in the earlier CABot1 parser (Huyck and Fan 2007),
185 the CABot2 parser uses Short-Term Potentiation (STP) to
186 bind. STP is a form of Hebbian learning that occurs in
187 biological neural systems (Hempel et al. 2000; Buonomano
188 1999). Hebbian learning implies that the co-firing of two
189 neurons tends to increase the synaptic strength between
190 them. With STP, this synaptic strength returns to its initial
191 value automatically over a relatively short period of disuse
192 (s or min), thus the binding is quick (approx. 40 ms) and it
193 can be reused.

194 Neural and other connectionist parsers

195 Interest in neural and other connectionist parsers is not
196 new. While non-neural connectionist parsers may have no
197 direct link to neural processing, they may provide a useful
198 set of metaphors.

199 One early parser was a component of a larger connec-
200 tionist natural language processing system (Mikkulainen
201 1993). This system used a recurrent back-propagation
202 network to parse. Unfortunately, these types of systems
203 have problems with longer sentences since earlier portions
204 of the sentence must be retained in the activation patterns
205 of the context nodes. Moreover, the overall system used
206 several different types of connectionist system, so the
207 overall architecture is quite ad hoc.

208 Another parser (Henderson 1994) uses a connectionist
209 system (Shastri and Aijanagadde 1993) based on associ-
210 ations. These associations use a frame system with
211 dynamic binding via synchrony. It is known that certain
212 constructs, like multiple centre embedded sentences, are
213 difficult for humans to parse. As the number of bindings
214 that the system supports is limited, the system also finds it
215 difficult to parse these types of sentences. Activation
216 decay and simulated annealing have been used to resolve

217 attachment decisions (Kempen and Vosse 1991). One of
218 the problems of most non-neural connectionist parsing
219 models is that there is little notion of time; while such
220 parsers must use word order information, this does not
221 provide timing data.

222 However, one hybrid-connectionist parser (Tabor and
223 Tanenhaus 1999) uses attractor basins, and the time the
224 parser takes to descend into a basin corresponds to the time
225 to make a parsing decision, that is, how long it takes to apply a
226 parsing rule. This is in the spirit of the CABot2 parser, as Cell
227 Assembly ignition (see section "Neuropsychology") is
228 equivalent to descent into an attractor basin (Amit 1989).
229 Similarly, simulated annealing (Kempen and Vosse 1991) is
230 related to statistical mechanics, which is used to formalize
231 attractor basins.

232 Non-neural connectionist parsers may provide insight
233 into parallel processing, but lack any direct link to neurons.
234 Even though it may be more difficult to develop systems
235 based on models with direct links to neurons, there have
236 been prior neural parsers. One such parser used a spiking
237 neural model to parse a regular language (Knoblauch et al.
238 2004). Importantly, like the CABot2 parser, this parser was
239 embedded in an agent. This shows that parsers can be
240 developed in simulated neurons.

241 Further evidence of the ability to develop parsers using
242 simulated neurons is the CABot1 parser (Huyck and Fan
243 2007). The CABot1 and CABot2 parsers have many sim-
244 ilarities, but the earlier parser uses a stack and there are
245 some indications that the human parser does not (Lewis
246 and Vasishth 2005). Both the CABot parsers make exten-
247 sive use of Cell Assemblies.

248 Neuropsychology

249 Hebb introduced Neuropsychology (Hebb 1949) and pro-
250 vided science with an intellectual bridge between neurons
251 and psychology. One of his key concepts was that of the
252 Cell Assembly (CA).

253 The CA hypothesis is that a CA is the neural basis of a
254 concept (Hebb 1949). A CA is a set of neurons that have
255 high mutual synaptic strength. There is now extensive
256 evidence that the brain does contain CAs (e.g. Abeles et al.
257 1993; Bevan and Wilson 1999; Pulvermuller 1999).
258 Moreover, the CA concept has also been extensively used
259 to explain and model cognitive tasks (e.g. Kaplan et al.
260 1991; Von Der Malsburg 1986; Wennekers and Palm
261 2000).

262 When a small subset of the neurons in a CA fire, a
263 cascade of activation ensues that leads to CA ignition
264 (Wennekers and Palm 2000); the CA can then persist after
265 the initial stimulus (the initial small subset of triggered
266 neurons) has ceased. This persistence is the neural imple-
267 mentation of psychological phenomena such as short-term

268 or working memory. The formation of the CA in the first
269 place is done via Hebbian learning, and this neural for-
270 mation constitutes a long-term memory.

271 The CA hypothesis gives two types of cognitive neu-
272 rodynamics. The first and faster dynamic is CA ignition,
273 where neurons fire and start a cascade that can then persist.
274 The second and slower dynamic is CA formation, which is
275 an emergent phenomena from a large number of synaptic
276 changes. This is often called the dual-trace mechanism.

277 It has been proposed that CAs gradually lose activity
278 over time (see section “[Complex rules and multi-valued](#)
279 [cell assemblies](#)”). One proposition that bridges the gap
280 between neuropsychology and parsing is that the stack that
281 is typically used for parsing is implemented by this gradual
282 loss of activity (Pulvermuller 2000). This proposal is in the
283 spirit of memory based parsers (see section “[Psychology](#)
284 [and linguistics](#)”) including the stackless CABot2 parser.

285 Psychology and linguistics

286 The research literature in both psychology and linguistics is
287 far too vast to summarize here. There is, however, some
288 research that attempts to unify these research fields. For
289 example, cognitive architectures (e.g. Anderson and Lebiere
290 1998) are systems whose ultimate goal is to be able to model
291 all cognitive functions. Similar work based on neural models
292 (e.g. Rolls 2008) is in its infancy and here the ultimate goal of
293 these systems is to show how the brain’s neural systems can
294 be modelled with simulated neurons to perform all cognitive
295 functions.

296 Similarly, in linguistics there are unifying theories. The
297 most famous is universal grammar (Chomsky 1965), but
298 this is largely about the way that humans learn language.
299 The tripartite theory (Jackendoff 2002) fits parsing into a
300 larger linguistic system, and then into a psychological
301 model. This theory shows how different aspects of lin-
302 guistics (e.g. semantics, syntax and the lexicon) inter-
303 relate. The theory is not, however, universally accepted.

304 Some linguistic theories are almost universally accep-
305 ted. These include the use of case frames to represent the
306 semantics of a sentence (Fillmore 1968) and bar-levels
307 (Jackendoff 1977) to account for simple and complex
308 phrases. Perhaps more importantly, and related to universal
309 grammar, is the notion of a combinatorial system. In this,
310 language is composed of components that can be combined
311 in a tree-like structure that has a practically infinite number
312 of possible topologies. Connectionist systems have been
313 criticised for a lack of compositional syntax and semantics
314 (combinatoriality) (Fodor and Pylyshyn 1988), but this
315 criticism is largely addressed by neural mechanisms used
316 for implementing variable binding (see section “[The](#)
317 [binding problem](#)”).

318 There has been a vast range of psycholinguistic work on
319 parsing. In linguistics, a distinction is often made between
320 performance and competence, with many psycholinguists
321 expressing the view that performance is not their concern,
322 so, in such circumstances, parsing is performance. While
323 many linguists may express a lack of interest in perfor-
324 mance, they are not saying it is not of interest in general.

325 Psycholinguistic work in parsing can be divided into
326 work that focuses on ambiguity, and work that focuses on
327 memory. One, of many approaches, that focus on resolving
328 ambiguity is a constraint based algorithm (MacDonald
329 et al. 1994) which simultaneously resolves lexical and
330 syntactic ambiguity. This has been tested on PP attachment
331 ambiguity among other phenomena. Also, work in eye
332 movement studies (Rayner 1998) (see section “[Timing](#)”) has
333 been extensively used to deal with back-tracking and to
334 show that humans make incorrect parsing decisions, and
335 have to go back and repair them. The incorrect decisions
336 illustrate some of the biases of the human parser.

337 A modern example of a memory based parser (Lewis
338 and Vasishth 2005) is based on the ACT-R cognitive
339 architecture (Anderson and Lebiere 1998). In this model,
340 each word and phrase is represented by a symbolic memory
341 chunk that has an associated activation level. This level
342 decreases over time, although it is reactivated when the
343 memory is re-accessed and this level is guided by ACT-R’s
344 equations. These equations have been used in a wide range
345 of other psychological models, both linguistic and non-
346 linguistic. The activation levels are then used to resolve
347 attachment decisions. For instance, this mechanism can be
348 used to account for center embeddings and to fail to
349 interpret center embedded sentences that people find dif-
350 ficult to interpret.

351 The CABot2 parser is a memory based parser that is able
352 to resolve the types of ambiguity discussed above. By the
353 use of frames, it is capable of generating a combinatorial
354 representation of semantics.

355 The neural model

356 This paper’s results and conclusions depend on the simu-
357 lated neural model being a reasonably accurate biological
358 model. The neural model that forms the basis of CABot2 is
359 a fatiguing Leaky Integrate and Fire (fLIF) model. It is not
360 as accurate as compartmental models (e.g. Hodgkin and
361 Huxley 1952; Dayan and Abbott 2005), but is much more
362 efficient to simulate. As parsing is complex, efficiency of
363 simulation is important. The fLIF model is an extension of
364 the more popular Leaky Integrate and Fire (LIF) model
365 (Tal and Schwartz 1997), which is in turn an extension of
366 the Integrate and Fire model (McCulloch and Pitts 1943).

367 A brief description of the fLIF model is given below, and a more detailed one can be found elsewhere (Huyck 368 1999, 2007). In the Integrate and Fire model, a neuron 369 collects activation from other neurons, and fires when it has 370 sufficient activation to surpass a threshold θ . When the 371 neuron fires, it sends activation to each neuron to which it 372 has synapses, and the activation is directly proportional to 373 the weight associated with each synapse. The fLIF model 374 uses discrete cycles, so the activation that is sent from a 375 neuron that fires in a cycle is not collected by the post- 376 synaptic neuron until the next cycle. If a neuron fires, it 377 loses all its activation, but if it does not fire, it retains some, 378 while some activation leaks away (decay); this is the leaky 379 component and is modelled by a factor $D > 1$, where the 380 activation is divided by D to get the initial activation at the 381 next step. 382

$$A_i = \frac{A_{i-1}}{D} + \sum_{j \in V_i} w_{ji} \quad (1)$$

384 Equation 1 shows the activity of a neuron at time t . The 385 neuron combines the retained activation after leak and the 386 new activation from the active inputs of all neurons $j \in$ 387 V_i , V_i being the set of all neurons that fired at $t - 1$ that are 388 connected to i , weighted by the value of the synapse from 389 neuron j to neuron i .

390 The LIF model is a widely used model of biological 391 neurons, although the extension of having neuron fatigue is 392 relatively novel. The idea of fatigue is that the more a 393 neuron fires, the harder it becomes to fire, that is, neurons 394 tire. This is modelled, in this paper, by each neuron having 395 an additional fatigue value which is increased by a constant, F_c , 396 in cycles in which the neuron fires, and decreased 397 by a constant, F_r , in cycles where the neuron does not fire. 398 The value never goes below zero, and the neuron's fatigue 399 value is added to the threshold, θ , to establish if a neuron 400 fires. So, if a neuron becomes highly fatigued, then it will 401 need a great deal of activation to fire. This is shown in 402 Eq. 2, where the neuron fires at time t if its activity A 403 minus fatigue F is greater than the threshold.

$$A_i - F_i \geq \theta \quad (2)$$

405 One emergent property of fatigue across all the neurons 406 in a CA is that fatigue can cause a CA to stop firing. 407 Practically, it is used in the CABot2 parser to show how 408 long a memory item has been active (section “Complex 409 rules and multi-valued cell assemblies”), and to 410 automatically shut down rules (section “Simple rule 411 activation and instantiation”).

412 The LIF model is widely used because it is a simple 413 model of a neuron that is relatively accurate biologically. 414 The fLIF model is slightly more complex, and is a slightly 415 better model. A model similar to the one described in this 416 paper (Chacron et al. 2003) has been shown to mimic

biological neural responses, particularly with respect to 417 neuronal adaptation, and does provide a more accurate 418 simulation than the simpler LIF models. 419

420 CAs composed of fLIF neurons can interact with each 421 other in a range of ways. Perhaps the simplest is for one 422 CA to cause another to ignite, which is done by having 423 neurons from the first send sufficient activation, via 424 synapses, to the second to ignite it. A more complex 425 mechanism is to require two CAs to be on to ignite a third, 426 while neither of the original alone is sufficient to ignite the 427 third. Requiring two CAs to be active to ignite a third is a 428 mechanism for controlling spreading activation. This 429 mechanism can be used to implement finite state automata 430 (Fan and Huyck 2008). A third type of interaction is to 431 have a CA suppress another so that its neurons stop firing 432 (the second CA is extinguished). The processing of the 433 CABot2 parser is driven by these types of CA interactions.

The CABot2 parser 434

435 The CABot2 parser is merely a network of fLIF neurons 436 with a symbolic interface to allow each word in a sentence 437 to be input. There is also a mechanism for converting the 438 subsymbolic semantic representation into a symbolic one 439 for output. CABot2 has a network of 30,000 neurons which 440 have been divided into 15 subnetworks. The threshold, θ ; 441 decay, D ; fatigue; F_c ; and fatigue recovery, F_r remain 442 constant within a subnetwork but may differ between 443 subnets (see Table 1). These subnets have been used to 444 facilitate the system's development, but they also fit a 445 logical, and to lesser extent a psycholinguistic, structure.

Table 1 Subnetwork Constants and Sizes for the CABot2 Parser

Name	Threshold	Decay	Fatigue	Fatigue rec.	Neurons
Input	4	1.5	0	0	3,000
Noun Access	4	2.0	0.8	0.8	1,800
Verb Access	4	2.0	0.8	0.8	900
Other Access	4	2.0	0.8	0.8	900
Next Word	4	12.0	0	0	200
Bar One	4	1.5	0.8	0.8	200
Rule One	4	2.0	0.5	0.4	1,200
Noun Semantics	4	2.0	0.8	0.8	10,200
Verb Semantics	4	2.0	0.8	0.8	5,400
Noun Instance	4	1.5	0.01	0.011	2,000
Verb Instance	4	1.5	0.01	0.011	1,000
Counter	4	2.0	0	0	600
Rule Two	4	1.2	0.5	0.45	1,800
PP to NP	4	1.25	0	0	400
PP to VP	4	1.25	0	0	400

446 Overall topology

447 Figure 1 is a schematic of the network. Each box refers to a
 448 subnet except the *Access* box, which refers to three separate
 449 subnets: the noun access, verb access, and other lexical
 450 item access subnets. Information largely flows from the top
 451 to the bottom with *Input* leading to *Access* and *Semantics*
 452 then being activated. Composite structures are built in the
 453 *Instances* with the *Rules* and *Bar One* subnets explicitly
 454 invoking state changes.

455 The overall topology adheres to a tripartite linguistic
 456 theory (Jackendoff 2002). In this theory there are separate
 457 lexical, syntactic, and semantic systems. These communicate
 458 by special communication systems (e.g. the lexical
 459 syntactic communication system). The lexical system is on
 460 the top right of the figure with the *Input* subnet entirely
 461 within that system. The syntax system is on the top left
 462 with the rules entirely within the syntax system. The
 463 semantics system is on the bottom with the instances
 464 entirely within that system. The other subnets cross these
 465 systems boundaries. For example, the access subnets are
 466 part of the lexical syntactic communication system. Note
 467 that the focus of the CABot2 parser has been on the syntax
 468 system; the lexical system in particular is under specified,
 469 and the semantic system is somewhere in between. The
 470 tripartite theory also allows extra links from these systems
 471 to others, e.g. from semantics to other systems such as
 472 perception, planning and action.

473 The particulars of these subnets are more fully explained
 474 below. The number of neurons in the subnets and the
 475 parameters are largely driven by expediency. That is,
 476 engineering decisions had a large role in determining these
 477 parameters. The explanation of the subnets starts with the

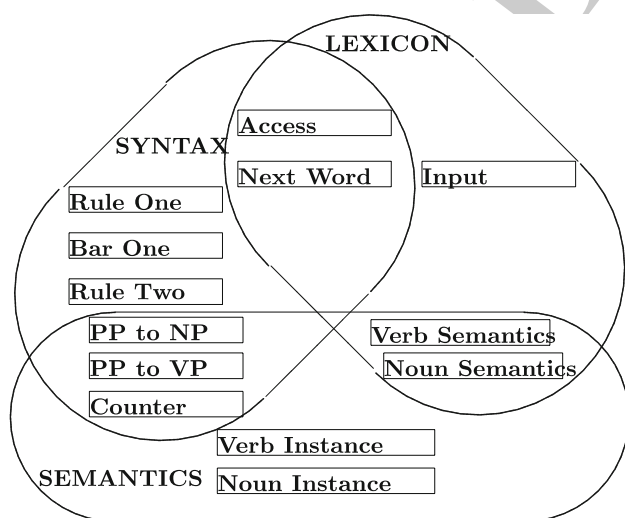


Fig. 1 Gross topology of the CABot2 parser. Each *box* represents a subnet with similar subnets grouped together according to Jackendoff's Tripartite theory

initial input and traces the processing of the example 4 478
 sentence in the following sections. 479

Input, access and semantics 480

Input is a symbolic action that is achieved by activating the 481
 CA for the input word, and only one word is active at a 482
 time. This is done when a particular rule CA in the *Rule* 483
One subnet ignites: the *Read Next Word* rule. This rule also 484
 ignites the first CA of the *Bar One* subnet called *Word* 485
Active. So to start parsing, the *Read Next Word* rule is 486
 ignited. The next *Input* CA, consisting of 100 neurons, 487
 is then ignited, and remains active until the next input is 488
 received. 489

The *Bar One* subnet has two CAs of 100 neurons. The 490
 first is called *Word Active*, and is active while the input 491
 word is directly activating CAs in other subnets. The second 492
 CA in *Bar One* is *Bar One Active*. This relates to X-bar 493
 theory (Jackendoff 1977); roughly, there are simple and 494
 complex phrases, with the simple phrases being bar one, so 495
 the *Bar One Active* CA is firing while the simple phrase is 496
 being constructed. 497

(example 4) The girl saw the dangerous pyramid with 498
 the stalactite. 499

Parsing the sentence from example 4 starts with the 500
Read Next Word rule being ignited, which turns on the 501
Input CA for *The*, and the *Word Active* CA in the *Bar One* 502
 subnet. The combined activation from these two CAs is 503
 enough to cause the *The* CA in the *Other Access* subnet to 504
 ignite. Each word has an element in one of the access 505
 subnets; there is no lexical ambiguity resolution in the 506
 CABot2 parser, so, for instance, *left* is always a noun and 507
centre is always a verb. 508

Later, the word *girl* is read. This causes the *girl* input 509
 CA and the *Word Active* CA to ignite. These combine to 510
 ignite the *Noun Access* CA for *girl*. This sends activation to 511
 the *Noun Semantics* subnet, which ignites the semantics for 512
girl. Each noun and verb is semantically represented by a 513
 hierarchical series of features. In the case of *girl*, this 514
 consists of *girl*, *person*, *living-thing*, *object*, and *physical-* 515
entity. For nouns, this hierarchy is derived from WordNet 516
 (Miller 1990). For verbs, this hierarchy is derived from a 517
 verb hierarchy available locally. This type of hierarchical 518
 encoding can be learned (Huyck 2007), but for reasons of 519
 technological expediency when implementing it on a PC, 520
 CABot2 had its hierarchical encoding hard-coded. 521

It should be noted that this topology of CAs is inconsis- 522
 tent with current understanding of CAs in biological 523
 systems. Firstly, aside from the two semantic subnets, 524
 where CAs share neurons, CAs are orthogonal with each 525
 neuron being in only one CA. Secondly, CAs are by and 526
 large composed of sets of features that are in turn 527

528 composed of 10 neurons. These neurons once on, oscillate
529 from one set of five firing, to the other five firing. This is a
530 persistent CA, but is not the kind of CA that has been
531 learned in past simulations. This was done to minimise the
532 number of neurons used, and to mathematically guarantee
533 behaviour. Unfortunately, simulation time slows markedly
534 as the number of neurons and synapses increase, so the
535 simulations are forced to use a relatively small number of
536 neurons. It is expected that the same behaviour could be
537 generated using larger CAs that are less uniform.

538 Simple rule activation and instantiation

539 The syntax system builds simple phrases and complex
540 phrases. The result of rule applications are stored in the
541 instance nets, in bindings between the instance nets, and
542 bindings from the instance nets to the access nets.

543 For example, the combined activation of *Word Active*
544 and the *the Access* CAs causes two rules to ignite, both in
545 the *Rule One* subnet. One is the *New Noun Instance* rule.
546 This causes a new instance to become active in the *Noun*
547 *Instance* subnet. Instances are the data structures that are
548 populated by parsing.

549 Instances are managed by the *Count* subnet, whose sole
550 purpose is to note the next free noun instance and verb
551 instance. Initially there are no verb or noun instances. This
552 is represented in the *Count* subnet by having a CA associ-
553 ated with zero for each of these. These CAs prime, but do
554 not ignite, the CAs associated with a count of one. When
555 the *New Noun Instance* rule ignites, it stimulates all of the
556 *Count*'s noun CAs. As the only one that is primed is the
557 *one* CA, it ignites, and in turn extinguishes the *zero* noun
558 count. This count CA in turn ignites the first noun instance
559 in the *Noun Instance* subnet. As yet, there is no information
560 in the instance, but it is now active. A duplicate mecha-
561 nism is used to get a new verb instance when the *New Verb*
562 *Instance* rule is applied. Instances follow case frame theory
563 (Filmore 1968), and the overall grammar with features is
564 amenable to analysis from unification-based grammar
565 (Shieber 1986), and head driven phrase structure grammar
566 (Pollard and Sag 1994).

567 As noted above, two rules ignite simultaneously. The
568 second rule that ignites along with the *New Noun Instance*
569 rule is the *NP adds det* rule. This switches on the *deter-*
570 *miner* feature of the open noun instance. This is done again
571 by having two CAs on and these two turn on a third CA, or,
572 as in this case, a third subCA. Features are represented by
573 neurons in the instance. The rule stimulates the *determiner*
574 features of all the noun instances and the open noun
575 instance stimulates all of its bar one features. Together,
576 these turn on the *determiner* feature.

577 Figure 2 shows firing behaviour in the *Verb Instance*,
578 *Rule One* and *Rule Two* subnets. Each dot represents a

579 neuron firing in a particular cycle. The *Verb Instance*
580 neurons are the bottom 500 neurons, and it can be seen that
581 it begins around cycle 65, and persists through to the end of
582 the parse. It can also be seen that different rules ignite at
583 different times.

584 Instances can be in one of four states. The first is
585 inactive, meaning that no neurons are firing. The second is
586 open, meaning that a simple noun phrase is under con-
587 struction. Part of X-bar theory states that there is at most
588 one simple phrase currently under construction at any time
589 (Jackendoff 1977). The third and fourth states of instances
590 are the active complex phrase state, and the done state (see
591 section “Complex rules and multi-valued cell assem-
592 blies”). When an instance is started, it is open, and this is
593 marked by having a specific feature in the instance firing.
594 When the instance is closed, this feature is turned off.

595 The *NP adds det* rule also turns off the *Word Active* CA
596 in the *Bar One* subnet, meaning that the net has finished, or
597 is about to finish, with processing a word. It also turns off
598 the *The* CA in the *Other Access* subnet.

599 The *NP add det* rule then switches off automatically
600 through a combination of loss of external input (*Word*
601 *Active* is now off), and fatigue. The fatigue constant is
602 greater than the fatigue recovery one (see Table 1), and
603 neurons are on in only half of the cycles. This causes
604 fatigue to accumulate, eventually, as the neurons do not
605 have enough activation to surpass the threshold plus fati-
606 gue, so they stop firing after nine cycles, and so the CA is
607 extinguished.

608 The *Next Word* subnet now comes into play. The system
609 will try to apply any rules that it can. However, if no rule
610 has applied in 10 cycles, the *Next Word* rule in the *Rule*
611 *One* subnet will come on. This is done by the *Next Word*
612 subnet which is a counter. It counts 10 cycles using 10
613 pseudo-CAs. Each of these pseudo-CAs turns on the next,

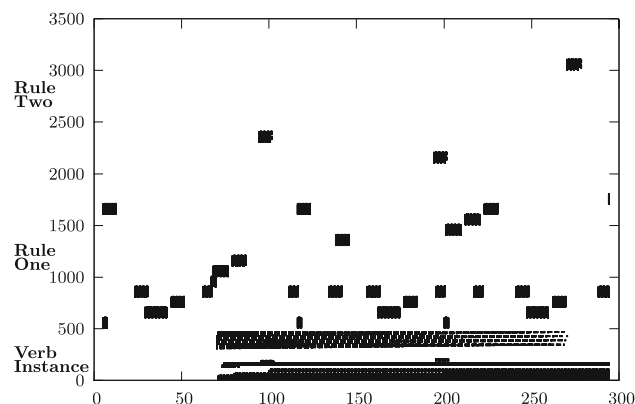


Fig. 2 Rastergram of the verb instance and rule SubNets. The first 500 neurons are verb instances, the next 1,200 from Rule One, and the remainder from Rule Two

614 and turns off the prior one. The first pseudo-CA also turns
615 off all of the others except the second. All of the rules
616 turn on the first one, and this implements the counter. The
617 last of the pseudo-CAs turns on the *Next Word* rule in the
618 *Rule One* subnet.

619 As noted in section “[Input, access and semantics](#)”, *girl*
620 now comes into the *Input* subnet. It then, in collaboration
621 with other subnets, turns on the *girl* CAs in the *Noun*
622 *Access* and *Noun Semantics* subnets.

623 As before, the *Word Active* CA is on; in collaboration
624 with the *girl Noun Access* CA, the *NP adds N* rule ignites
625 in the *Rule One* subnet. This turns on the *Main Noun*
626 feature in the open noun instance. This feature is repre-
627 sented by some neurons that learn via STP (see section
628 “[The binding problem](#)”). These neurons connect to all of
629 the noun instances, and as the only noun instance that is
630 active is *girl*, this instance is bound to *girl* after a few
631 cycles of co-firing.

632 Next the *NP Done* rule is applied. This turns off all noun
633 access CAs, and both *Bar One* CAs. This means the system
634 is done with the word, and done with the instance as a
635 simple phrase. The rule also turns off the noun instance by
636 switching on the *Bar One Done* feature.

637 Note that there are parallel features for *Bar One Done*
638 and *Bar One Open*. The open feature is turned off when the
639 instance is done, but the open feature has fast bind neurons
640 that bind to the rest of the features if they are turned on.
641 This provides memory within the instance. A duplicate
642 mechanism is used to support features in the verb instances.

643 A similar process now occurs with the word *saw*. The
644 *Next Word* rule comes on, which causes *saw* to be propa-
645 gated through to the verb access and semantics subnets. A
646 new verb instance is created, and *saw* is made the main
647 verb when the *VP adds Main Verb* rule is applied. The
648 *Verb Done* rule is then ignited, and the verb instance is
649 closed.

650 Complex rules and multi-valued cell assemblies

651 Having processed *saw*, two instances are available and the
652 system can now apply phrase combination rules. These
653 rules are in the *Rule Two* subnet, and are quite similar to
654 the simple phrase creation rules. These rules will not be
655 applied when a simple phrase is under creation because
656 they are inhibited by the *Bar One Active* CA.

657 A few cycles after the *VP Done* rule for *saw* ceases
658 firing, the *VP → NPactor VP* rule is applied. This rule
659 receives activation from both the verb instance and the
660 noun instance. When it ignites, it firstly turns on the verb’s
661 *actor* slot (feature). This slot has neurons that learn via
662 STP, and these neurons have connections to all of the noun
663 instances. The only active noun is the instance that is
664 bound to *the girl*, and so it is bound as the actor after a few

665 cycles. The *actor* slot also turns on the *actor-done* feature
666 which inhibits further application of the rule, and turns off
667 the *actor* slot so that no further binding will occur. Addi-
668 tionally, the noun instance has its *bound* feature turned on,
669 so that it will no longer be used as a slot filler. Note that the
670 application of *VP → NPactor VP* can be seen in Fig. 2; it
671 can be seen at neuron 2,300 near cycle 100. Other rules can
672 be picked out.

673 This rule application is quite similar to the application of
674 simple noun phrase rules. However, two problems arise:
675 the first is that there needs to be multiple rules for each slot;
676 the second is that without a stack, some mechanism is
677 needed to select between rules.

678 Without a stack, some mechanism is needed to select
679 between rules. In the above example, the *VP → NPactor*
680 *VP* rule is selected instead of the *VP → VP NPobject* rule.
681 The system has no explicit idea of order, so how does it
682 know to select the actor rule?

683 The answer to this lies in the third state of instances (see
684 section “[Simple rule activation and instantiation](#)”). Having
685 been completed as a simple phrase, both instances are in
686 the third state (active complex phrase). Also, when an
687 instance is created, a set of its neurons are activated that act
688 as a counter for how long the instance has been active. This
689 system is set up in groups of eight neurons with six neurons
690 firing in each cycle. As $(F_c * 3) < F_r$ (see Table 1), the
691 circuit accumulates fatigue. Due to fatigue these counter
692 neurons gradually stop firing; this is how it acts as a
693 counter. For a more complete explanation see (Passmore
694 and Huyck 2008). Each instance CA has a set of counter
695 neurons. This can be seen in Fig. 2, where the counter
696 neurons are between 300 and 500. These start out firing,
697 and then gradually decline.

698 The counter neurons are used as input to the actor and
699 object rules. For the actor rule, extra activation comes from
700 the verb because it is more active. For the object rule, extra
701 activation comes from the noun because it is more active
702 when the object rule is applied. Passive constructions could
703 be folded in with a passive feature on the verb instance.
704 Additionally, rule CAs have mutual inhibition, so while
705 one is active, others must wait until it has completed.

706 To return to the example, *the dangerous pyramid* is
707 processed in a similar manner to *The girl*, and a new
708 instance is duly created for it. When this instance is com-
709 pleted, the *VP → VP NPobject* rule is applied, and it is
710 bound to the object slot of the verb. Similarly, *with the*
711 *stalactite* is made into a noun instance with the preposition
712 feature set.

713 At this stage, there is a PP attachment ambiguity that is
714 resolved to attaching *with the stalactite* to *the pyramid*.
715 That is, the rule *NP → NP PP* is applied. Note that while
716 the instance for *the pyramid* has its bound feature on, it is
717 still open to having something bound to it. In this case the

718 *PP modifier* feature of *the pyramid* is bound to the noun
719 instance for *with the stalactite*.

720 This is a proactive form of attachment that has been
721 used in other natural language processing models. Unlike
722 traditional context free parsers, it focuses on attaching
723 items as soon as possible. For words, it has been suggested
724 that each word is incorporated into the sentence immedi-
725 ately (Milward and Cooper 1994). For phrases, this is a
726 form of left-corner parsing, e.g. (Roark and Johnson 1999).

727 Another problem is closing off phrases so that they
728 cannot have another phrase attached to them. This happens
729 to the first noun phrase in example 5.

730 (example 5) I saw with the telescope.

731 Here the noun phrase *I* is incorporated into the verb frame
732 by the application of the *VP* → *NPactor VP* rule, however,
733 the noun instance is still active, and thus the prepositional
734 phrase *with the telescope* could be attached to it. This is
735 prevented by a feature in the noun instance that is turned on
736 by the actor rule. When this feature is on, the noun
737 instance, has moved to the fourth state, done.

738 Attention should be drawn to one major difference
739 between the two rule subnets. They have different decay
740 rates with *Rule One* having a decay of 2, and *Rule Two* a
741 decay of 1.2. This means that in each cycle when a neuron
742 does not fire, more activation leaks away from a neuron in
743 the *Rule One* subnet than from a neuron in the *Rule Two*
744 one. This also means that evidence can take longer to
745 accumulate for the phrase combination rules in *Rule Two*.
746 Figure 2 also shows that the number of *Rule Two* appli-
747 cations is much smaller than the number of those from
748 *Rule One*.

749 This evidence is used to make more complex decisions
750 in, for example, PP attachment. Here the system makes use
751 of known attachment decisions to decide how to attach a
752 PP. There are two subnets, the *PP to NP* and *PP to VP*
753 subnets which are used for making attachment decisions.
754 These subnets get activation from the *Noun Semantics* and
755 *Verb Semantics* subnets that is sufficient to ignite particular
756 CAs when the appropriate words are active. For example,
757 one CA in the *PP to VP* subnet gets activation from *saw*,
758 *girl*, and *telescope*, that is sufficient to ignite it. This CA in
759 the *PP to VP* subnet in turn sends activation to the *VP* →
760 *VP PP instrument* rule, causing it to ignite and be applied.
761 Similarly, one CA in the *PP to NP* subnet gets activation
762 from *move*, *door* and *handle*, and sends activation to *NP* →
763 *NP PP*.

764 Results

765 The CABot2 parser is not capable of parsing all English
766 sentences, but it does parse several common constructs

correctly. It is a relatively capable parser which can handle 767
the basic commands that are needed within the CABot2 768
computer game environment and produce correct semantic 769
output. More importantly, it is based on a neural model 770
with a link to biological time, and parses in times that are 771
similar to human timing data. It resolves PP attachment 772
ambiguity in a way that appears to be similar to the way 773
humans resolve these ambiguities, and demonstrates one 774
way that semantics can be involved in making parsing 775
decisions. Finally it can be readily incorporated into a 776
neural agent, and thus can make use of evidence that is not 777
normally available to other computational parsers, but is 778
available to the human parser. 779

Semantic output 780

The CABot2 parser has been tested on 27 sentences, and 781
produces the correct semantic output for all of these. This 782
is a small number of sentences, but does include a range of 783
constructs including imperative sentences, multiple PP and 784
NP slots, and PP attachment ambiguities. Aside from PP 785
ambiguities, all sentences that have the same lexical format 786
will parse correctly. Even with the small number of words 787
in the current lexicon, 28, this means thousands of sen- 788
tences can be parsed by the system. 789

The semantic results of a parse are calculated by turning 790
all neurons off, then turning the first verb instance on. This 791
spreads activation through the bindings to other instances, 792
and then on to the *Access* subnets. After 45 cycles, by 793
which time the system will be stable if a parse is suc- 794
cessful, the nets are measured to create a symbolic version, 795
and this is the semantic output of the sentence. 796

The noun instances have the determiner, preposition, 797
adjective, main noun and prepositional phrase modifier 798
slots. The verb instance has the main verb, actor, object, 799
location and instrument slots. All of these were tested and 800
behaved correctly on the 27 sentences. 801

Timing 802

An important consideration for a neuropsychological parser 803
is that it parses in the correct time, that is, in times 804
equivalent to those obtained from experimental human 805
performance data. The fLIF neural model is based on 806
cycles, and these cycles correspond roughly to 10 ms of 807
biological time. The neurons are not much faster because 808
they ignore refractory periods and synaptic delays, all of 809
which happen generally in under 10 ms. Also, biological 810
neurons generally do not spike more than once in a 10 ms 811
period. 812

Similarly, humans read at a wide range of speeds. None 813
the less, studies have been done using eye tracking to see 814
when people foveate (fix their eyes) on particular words. 815

816 This is one widely used way to measure how people are
817 parsing sentences (Rayner 1998).

818 van Gompel et al. (2001) used eye tracking to see how
819 people read sentences with PP attachment ambiguities.
820 Figure 3 gives a comparison of the CABot2 parser and the
821 human performance data. The *x*-axis represents the word in
822 the sentence that is being read, and the *y*-axis is time in
823 milliseconds. The solid line is the parser's performance
824 assuming that each cycle is 10 ms and that each word's
825 processing is completed before the next word is read. The
826 dotted lines represent human performance; humans do not
827 foveate on each word and the human data was reported by
828 groups of words. In the example, the words were grouped
829 as follows: *The girl, saw, the dangerous pyramid, with the,*
830 *and stalactite*; the additional period is included to show the
831 end of the parse. The reported data was averaged across a
832 range of sentences with the same lexical content. The
833 human data that is reproduced in Fig. 3 is the total time
834 spent on a word group for ambiguous sentences. The parser
835 data was counted from the cycle that a new word was read,
836 indicating that processing of the prior word had been largely
837 completed.

838 The CABot2 parser performs with almost the exact same
839 timings as the human data. The time to parse the complete
840 sentence is 2,940 ms for the parser and 2,931 ms for the
841 human model. The average difference between the five
842 comparable data points is 55.2 ms.

843 This is not to say that the CABot2 parser is a perfect
844 model of human parsing timing. For instance, the parser
845 does not back track, and it is known that on some sentences
846 people do. Nonetheless, it does parse in roughly the correct
847 time, giving some support to the notion that it is doing
848 something like the human parser.

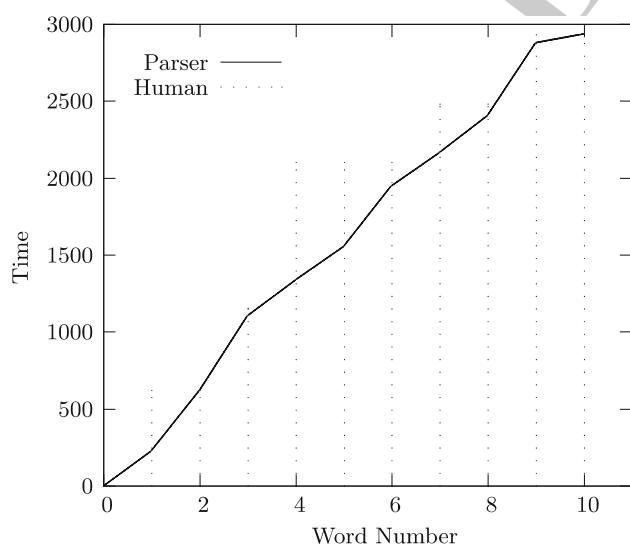


Fig. 3 Time spent to parse by word the sentence *The girl saw the dangerous pyramid with the stalactite*

PP-attachment

849

850 The CABot2 parser does resolve PP attachment ambiguities. Seven sentences were tested and all were attached correctly. The sentences are shown in Table 2. The first column represents the attachment decision that the parser makes for the sentence in the third column. The second column represents how the parser makes the decision and is further elaborated below.

851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885

The first sentence is the standard PP attachment example. People typically resolve this sentence, in the null context, by attaching the PP to the Verb so that *the telescope* is used as an instrument for *seeing* (Ford et al. 1982). As described in section “Complex rules and multi-valued cell assemblies”, there is a particular CA in the *PP to VP* subnet that is used to store the preference to attach this PP as the instrument of the verb. This CA is ignited by a combination of evidence from *see*, *girl* and *telescope*. The ignited preference CA in turn ignites the appropriate grammar rule. As the decision is stored, Table 2 marks this as *stored*.

For the second sentence, *see* and *telescope* still send activation to the preference CA, but alone are insufficient to ignite the rule. However, as the words are stored as a semantic hierarchy, *boy* shares many neurons with *girl*, and those shared neurons also send activation to the preference CA. Consequently, the preference CA is ignited followed by the grammar rule. In this case, the decision is not explicitly stored, but instead derived via a hierarchical relation, so Table 2 marks this decision as *inherited*.

Similarly, the third sentence has the attachment preference stored, but in this case it is stored in the *PP to NP* subnet so that *the door* has *the handle*. Again, this CA is ignited by a combination of the three inputs, and turns on the appropriate grammar rule. The fourth sentence is similar to the third, but the decision is not stored. The semantics of the words *door* and *gate* share neurons, so together they are sufficient to ignite the preference CA.

Table 2 Sentences with PP attachment ambiguities tested, their attachment to noun or verb, and the method of ambiguity resolution used

Attachment	Method	Sentence
Verb	Stored	I saw the girl with the telescope
Verb	Inherited	I saw the boy with the telescope
Noun	Stored	Move the door with the handle
Noun	Inherited	Move the gate with the handle
Noun	Default	Turn the telescope with the pyramid
Verb	Stored	Move it toward the stalactite
Verb	Inherited	Move it toward the pyramid

886 In the case of the fifth sentence, no preference CA is
887 ignited. Consequently, the default behaviour occurs, and
888 the PP is attached as a modifier of the noun. Note that
889 sentences three and five are lexically identical. However it
890 takes two cycles (or 20 ms) longer to begin to apply the
891 $NP \rightarrow NP PP$ rule for sentence five. That is, the default
892 decision takes longer as there is less information available.

893 Finally, the sixth and seventh sentences attach the PP to
894 the verb. The difference here is that they differ on the NP
895 instead of on the PP. This shows that inheritance works
896 on different elements, even though the elements are dif-
897 ferently weighted.

898 Elsewhere (Nadh and Huyck 2009), hierarchical rela-
899 tions have been used to learn attachment preferences,
900 although in a symbolic system. This shows that the basic
901 idea can be translated to a neural system. However, it is not
902 clear how well the neural approach will scale. That is, the
903 use of hierarchical CAs for the semantics of words may in
904 itself be insufficient to resolve a large number of decisions
905 as different preference CAs may begin to conflict. None the
906 less, it is obvious how these preference CAs can be gen-
907 erated for any learned relation.

908 CABot

909 People parse sentences in the context of both related sen-
910 tences and in the broader environment in which the
911 sentences occur. Particularly during conversation, parsing
912 interacts with other cognitive systems both receiving
913 information from them, for example, to resolve referential
914 ambiguity, and providing information to them. The
915 CABot2 parser is a component of an agent called CABot2.
916 The agent exists in a video game, and the agent, including
917 the parser, is implemented entirely in fLIF neurons.

918 At this stage, the agent is relatively simple and has gone
919 through two major versions with associated minor versions.
920 CABot2, the most recent, uses the CABot2 parser while
921 CABot1 used the earlier stack based parser. Timing for
922 CABot1 provides one of the major reasons for the devel-
923 opment of the CABot2 parser: the stack-based parser was
924 too slow. While a command like *Turn toward the pyramid*.
925 takes around 200 cycles in the CABot2 parser, it takes 800
926 in the stack-based parser due to time needed for stack
927 erasing.

928 The CABot agents act to support a user in the game. The
929 parser interprets commands from the user and uses the results
930 of these commands to set its internal goals. The game
931 requires that CABot2 interpret and implement 12 different
932 imperative commands. The parser generates the correct
933 interpretation for all of these.

934 Various minor versions of the agents have been devel-
935 oped to explore a range of capabilities, and two versions
936 are particularly relevant to this paper. In one version of

CABot1, the labels of some visual semantic categories 937
were learned by presenting them simultaneously with 938
visual instances of the category. This labelling is a portion 939
of the solution to the symbol grounding problem (Harnad 940
1990). Similarly, a second variant of CABot1 used an item 941
in the visual field to resolve the referent of the command 942
Turn toward it, showing the agent supports pronoun reso- 943
lution by context. 944

945 The CABot2 parser is being used for the next version of
946 the agent that is currently under development, CABot3. It
947 will need to understand about 20 new commands, but this
948 should be a straightforward extension to the current parser.
949 CABot3 will also use the above labelling work from the
950 variant of CABot1.

951 Discussion and conclusion

952 The four main goals of the CABot2 parser, laid out in the
953 introduction, have been met. Most importantly, the system
954 parses in a linguistically, psychologically, and neurally
955 plausible manner. That is not to say that it is a perfect
956 model, but it is consistent with current theories and data
957 obtained in all three fields. It is consistent with several
958 linguistic theories (e.g. Filmore 1968; Pollard and Sag
959 1994; Jackendoff 2002), parses a context free grammar,
960 and has a combinatorial representation of semantics that is
961 extensible to all linguistic semantics. It parses in a psy-
962 chologically plausible manner following a psycholinguistic
963 model (Lewis and Vasissth 2005). Short and long-term
964 memories are handled according to the long standing
965 neuropsychological CA hypothesis. Timing of short-term
966 memories and overall timing of parsing is consistent with
967 psychological evidence. The basic fLIF neural model is a
968 reasonably accurate, albeit relatively simple, model of
969 biological neurons. While the simulated neural topology is
970 specified, and in some cases biologically unlikely (e.g. 10
971 oscillating neurons for a feature, and mostly orthogonal
972 CAs, see section “Input, access and semantics”), it does
973 make use of CAs and in some cases hierarchical CAs.
974 These simplifications are caused mainly by a forced limi-
975 tation of size. Although biologically unlikely (and aside
976 from some very strong synapses), the topology does not
977 violate any known neural organisation principles.

978 As is almost certainly the case with people, preposi-
979 tional phrase attachment ambiguity is resolved by
980 semantics. In the cases where the attachment is known, it
981 performs flawlessly, that is, the system is capable of storing
982 pre-calculated decisions. Moreover, it is capable of han-
983 dling novel attachment decisions due to the hierarchical
984 nature of the stored semantics and their activation of
985 attachment preference rules. This use of the four-tuple
986 (verb, noun, preposition, noun) has been shown to be

987 effective in symbolic systems (Ratnaparkhi et al. 1994;
988 Nakov and Hearst 2005; Nadh and Huyck 2009) getting
989 more than 90% of decisions correct. However, it is intended
990 that future parsers, using context information, may perform
991 at or near human levels.

992 The CABot2 parser is relatively effective. It correctly
993 parses all of the test sentences in the current CABot
994 commands, and, as the topology has no randomness, it
995 parses these correctly every time. The expectation is that
996 this can be easily expanded to account for the further 20 or
997 30 commands that the next CABot agent will need to
998 understand. Moreover, the whole parsing process is rela-
999 tively efficient in both simulated and actual time. An
1000 additional and important advantage is that the relatively
1001 few neurons used for parsing leaves more available for
1002 other types of processing (e.g. vision and planning).

1003 Finally, the parser uses a reasonable semantic model.
1004 The representation of words as semantic hierarchies is one
1005 aspect of this, along with noun and verb instances to
1006 implement frames to store the semantics of phrases and
1007 sentences. This storage approach allows specific queries
1008 made of a sentence to interact with other systems, and
1009 CABot uses these instance frames to set its goals.

1010 As the four main goals have been met, the CABot2
1011 parser qualifies as a cognitive model. As a cognitive model,
1012 it provides evidence for the type of grammar that is used
1013 showing that a unification-based grammar can be used. It
1014 shows that PP attachment can use hierarchical relations to
1015 resolve ambiguity. Finally, the timing results show that
1016 proactive attachment can be efficiently implemented.

1017 While the CABot2 parser handles standard, prototypical
1018 English, parses in human-like time, and handles PP attach-
1019 ment ambiguity, it is by no means an industrial grade
1020 parser or even a particularly good psycholinguistic model.
1021 The belief is that by using the same techniques used to
1022 develop the parser, it could readily be scaled up, but this
1023 may not be the best way forward. Instead, a better under-
1024 standing of the neurodynamics of the system could be
1025 gained while developing a parser that learned rules and that
1026 a better parser would result from this. Of course, parallel
1027 improved understanding of the dynamics could also
1028 improve other related and connected systems that would
1029 also improve parsing performance.

1030 These improvements and expansions will run into a
1031 simulation boundary. The CABot2 parser has 30,000 neu-
1032 rons and systems with 100,000 fLIF neurons have been
1033 simulated in real-time on a standard PC, where real-time
1034 means a cycles takes 10 ms to simulate, or 100 cycles
1035 take about a second. Expansion beyond 100,000 neurons
1036 has radically slowed simulations. These sizes could be
1037 improved by improved hardware, distributing the simulator
1038 across PCs, or a more efficiently coded simulator, but it is
1039 expected that special neural hardware (Khan et al. 2008)

1040 will be available within a year or two. This should enable
1041 simulations in real-time of a billion neurons.

1042 Scaling up is relatively straightforward for words and
1043 grammatical constructs. The addition of new words and
1044 lexical classes is merely a linear change in the number of
1045 neurons, that is, each new word will only increase the
1046 number of neurons as much as the last word and perhaps
1047 less than this due to the hierarchical encoding of semantics.

1048 Grammar rules can readily be added, although phe-
1049 nomena like conjunction and gapping need further
1050 exploration. Since the CABot2 parser is based on current
1051 linguistic theories that account for these phenomena,
1052 however, such extensions are about implementation detail
1053 and not fundamental to the neurally based parsing approach
1054 reported.

1055 For example, in the current system, there is a rule for
1056 $VP \rightarrow VP NP_{object}$ that makes the NP the object of the
1057 verb. Unfortunately, there are three versions of this rule,
1058 one for the first NP instance, one for the second, and one
1059 for the third. The problem is that each needs activation
1060 from only one noun instance, and all from the single verb
1061 instance. If there were only one rule, multiple instances
1062 would all contribute activation to the rule and cause it to
1063 activate at the wrong time. This problem might be resolved
1064 by dynamic binding using active links (van der Velde and
1065 de Kamps 2006) or some other hierarchical activation
1066 mechanism, but it is currently a recognised flaw in the
1067 CABot2 system.

1068 The problem with multiple versions of rules for different
1069 instance pairs (see section “Complex rules and multi-valued
1070 cell assemblies”) is currently unsolved and could, in theory,
1071 lead to an explosion of rules as sentences grow longer. There
1072 is, of course, some upper sentence length limit for normal
1073 human parsing. Moreover, in the CABot2 parser, most
1074 instances are turned off early in processing so do not need to
1075 be accounted for. A dynamic binding mechanism can prob-
1076 ably be developed to overcome any remaining problems
1077 concerning multiple rules.

1078 Other linguistic systems, like a lexical system, phonet-
1079 ics, or discourse interpretation, or systems for production of
1080 all of these, could be developed and integrated with the
1081 parser. A lexical system could be used to resolve lexically
1082 ambiguous and polysemous words like *saw*. It is expected
1083 that these efforts would be of a similar degree of com-
1084 plexity to parser development but would be made easier by
1085 the skills, techniques, and knowledge already gained.
1086 Crucially, while these systems would be largely indepen-
1087 dent according to the tripartite theory, they would function
1088 in parallel. Thus the full system would process at roughly
1089 the same, simulated, speed.

1090 While the CABot2 parser could be scaled up, and sys-
1091 tems developed for other tasks, a better approach would be
1092 to develop systems that could learn the underlying rules,

1093 whether syntactic, lexical, phonological or of other types.
 1094 Initial work has begun on rule learning with CAs (Huyck
 1095 and Belavkin 2006; Belavkin and Huyck 2008), but it is
 1096 still in its early stages. Integrating rule learning with vari-
 1097 able binding (see section “The binding problem”) is one
 1098 obvious next step. When this issue is resolved, the system
 1099 will only need to be provided with the basics of Universal
 1100 Grammar (Chomsky 1965) and other systems (e.g. sensing,
 1101 effecting, and semantics) to learn to parse. Of course, the
 1102 difficulty of these tasks is not to be underestimated.

1103 Considering what brain areas these subnets simulate,
 1104 aside from some work on words (Pulvermuller 1999), and
 1105 the knowledge that Broca’s area is heavily involved in
 1106 language processing, at this stage any proposed link would
 1107 be highly speculative, although one could pursue Ander-
 1108 son’s (Anderson and Lebiere 2007) proposals linking
 1109 cognition to eight brain areas.

1110 It does appear that the CABot2 parser is a reasonable
 1111 cognitive model. If so, then this is proof that Smolensky’s
 1112 claim is out of date and that neural models are now capable of
 1113 being used for sophisticated cognitive modelling. More
 1114 importantly, these neural cognitive models may be able to
 1115 address new problems that symbolic and non-neural con-
 1116 nectionist systems cannot, such as timing, word coding, and
 1117 the neural implementation of memory, both short and long-
 1118 term. The link to neural data may also provide simple solu-
 1119 tions to problems that are otherwise difficult to solve. Neural
 1120 models also can solve the symbol-grounding problem that
 1121 cause problems for symbolic systems. It therefore seems
 1122 reasonable to expect that the development of these models
 1123 will lead to better AI systems.

1124 **Acknowledgements** This work was supported by EPSRC grant EP/
 1125 D059720. Thanks go to Dan Diaper and Emma Byrne for comments
 1126 on this paper.

1127 References

- 1128 Abeles M, Bergman H, Margalit E, Vaddia E (1993) Spatiotemporal
 1129 firing patterns in the frontal cortex of behaving monkeys.
 1130 *J Neurophysiol* 70(4):1629–1638
 1131 Amit D (1989) Modelling brain function: the world of attractor neural
 1132 networks. Cambridge University Press, Cambridge
 1133 Anderson J, Lebiere C (1998) The atomic components of thought.
 1134 Lawrence Erlbaum, Mahwah
 1135 Anderson J, Lebiere C (2007) How can the human mind occur in the
 1136 physical universe. Oxford University Press, Oxford
 1137 Belavkin R, Huyck C (2008) The emergence of rules in cell assemblies
 1138 of flif neurons. In: Proceedings of the eighteenth European
 1139 conference on artificial intelligence
 1140 Bevan M, Wilson C (1999) Mechanisms underlying spontaneous
 1141 oscillation and rhythmic firing in rat subthalamic neurons. *Neuro-*
 1142 *science* 19:7617–7628
 1143 Buonomano D (1999) Distinct functional types of associative long-
 1144 term potentiation in neocortical and hippocampal pyramidal
 1145 neurons. *Neuroscience* 19(16):6748–6754

- Carpenter G, Grossberg S (1988) The art of adaptive pattern
 1146 recognition by a self-organizing neural network. *IEEE Comput*
 1147 *21*:77–88
 1148 Chacron M, Pakdaman K, Longtin A (2003) Interspike interval
 1149 correlations, memory, adaptation, and refractoriness in a leaky
 1150 integrate-and-fire model with threshold fatigue. *Neural Comput*
 1151 *15*:253–278
 1152 Chomsky N (1965) Aspects of the theory of syntax. MIT Press,
 1153 Cambridge
 1154 Dayan P, Abbott L (2005) Theoretical neuroscience: computational
 1155 and mathematical modeling of neural systems. MIT Press,
 1156 Cambridge
 1157 Fan Y, Huyck C (2008) Implementation of finite state automata using
 1158 flif neurons. In: IEEE systems, man and cybernetics society,
 1159 pp 74–78
 1160 Filmore C (1968) The case for case. In: Back E, Harms R (eds)
 1161 Universals in linguistic theory. Holt, Rinehart and Winston Inc.,
 1162 New York
 1163 Fodor J, Pylyshyn Z (1988) Connectionism and cognitive archite-
 1164 cture: a critical analysis. *Cognition* 28:3–71
 1165 Ford M, Bresnan J, Kaplan R (1982) A competence-based theory of
 1166 syntactic closure. In: Bresnan J (ed) The mental representation of
 1167 grammatical relations. MIT Press, Cambridge
 1168 Harnad S (1990) The symbol grounding problem. *Phys D* 42:335–346
 1169 Hebb DO (1949) The organization of behavior. Wiley, New York
 1170 Hempel C, Hartman K, Wang X, Turrigiano G, Nelson S (2000)
 1171 Multiple forms of short-term plasticity at excitatory in rat medial
 1172 prefrontal cortex. *J Neurophysiol* 83:3031–3041
 1173 Henderson J (1994) Connectionist syntactic parsing using temporal
 1174 variable binding. *J Psycholinguist Res* 23(5):353–379
 1175 Hodgkin A, Huxley A (1952) A quantitative description of membrane
 1176 current and its application to conduction and excitation in nerve.
 1177 *J Physiol* 117:500–544
 1178 Huyck C (1999) Modelling cell assemblies. Technical report,
 1179 Middlesex University
 1180 Huyck C (2007) Creating hierarchical categories using cell assem-
 1181 blies. *Connect Sci* 19(1):1–24
 1182 Huyck C (2008) CABot1: a videogame agent implemented in
 1183 flif neurons. In: IEEE systems, man and cybernetics society,
 1184 pp 115–120
 1185 Huyck C, Belavkin R (2006) Counting with neurons: rule application
 1186 with nets of fatiguing leaky integrate and fire neurons. In:
 1187 Proceedings of the seventh international conference on cognitive
 1188 modelling, pp 142–147
 1189 Huyck C, Fan Y (2007) Parsing with flif neurons. In: IEEE systems,
 1190 man and cybernetics society, pp 35–40
 1191 Jackendoff R (1977) X-bar syntax: a study of phrase structure. MIT
 1192 Press, Cambridge
 1193 Jackendoff R (2002) Foundations of language: brain, meaning,
 1194 grammar, evolution. Oxford University Press, Oxford
 1195 Kaplan S, Sontag M, Chown E (1991) Tracing recurrent activity in
 1196 cognitive elements (TRACE): a model of temporal dynamics in a
 1197 cell assembly. *Connect Sci* 3:179–206
 1198 Kempen G, Vosse T (1991) Incremental syntactic tree formation
 1199 in human sentence processing: a cognitive architecture based
 1200 on activation decay and simulated annealing. *Connect Sci* 1:
 1201 275–292
 1202 Khan M, Lester D, Plana L, Rast A, Painkras J, Furber S (2008)
 1203 SpiNNaker: mapping neural networks onto a massively-parallel
 1204 chip multiprocessor. In: Proceeding of 2008 international joint
 1205 conference on neural networks, pp 2850–2857
 1206 Knoblauch A, Markert H, Palm G (2004) An associative model of
 1207 cortical language and action processing. In: Proceedings of the
 1208 ninth neural computation and psychology workshop, pp 79–83
 1209 Lewis R, Vasishth S (2005) An activation-based model of sentence
 1210 processing as skilled memory retrieval. *Cogn Sci* 29(3):375–419
 1211

- 1212 MacDonald M, Pearlmutter N, Seidenberg M (1994) Lexical nature of
1213 syntactic ambiguity resolution. *Psychol Rev* 101(4):676–703
- 1214 McCulloch W, Pitts W (1943) A logical calculus of ideas immanent
1215 in nervous activity. *Bull Math Biophys* 5:115–133
- 1216 Mikkulainen R (1993) *Subsymbolic natural language processing*. MIT
1217 Press, Cambridge
- 1218 Miller G (1990) WordNet: an on-line lexical database. *Int J Lexicogr*
1219 3(4):235–312
- 1220 Milward O, Cooper R (1994) Incremental interpretation: applications,
1221 theory and relationships to dynamic semantics. In: 15th inter-
1222 national conference of computational linguistics, pp 748–54
- 1223 Nadh K, Huyck C (2009) Prepositional phrase attachment ambiguity
1224 resolution using semantic hierarchies. In: *Proceedings of Artificial
1225 Intelligence and Applications (AIA 2009)*. Innsbruck, Austria
- 1226 Nakov P, Hearst M (2005) Using the web as an implicit training set:
1227 application to structural ambiguity resolution. In: *Proceedings of
1228 HLT-NAACL*
- 1229 Passmore P, Huyck C (2008) Models of cell assembly decay. In: *IEEE
1230 systems, man and cybernetics society*, pp 251–256
- 1231 Pollard C, Sag I (1994) *Head-driven phrase structure grammar*. Center
1232 for the Study of Language and Information, Stanford
- 1233 Pulvermuller F (1999) Words in the brain's language. *Behav Brain
1234 Sci* 22:253–336
- 1235 Pulvermuller F (2000) Syntactic circuits: how does the brain create
1236 serial order in sentences? *Brain Lang* 71:194–199
- 1237 Ratnaparkhi A, Reynar J, Roukos S (1994) A maximum entropy
1238 model for prepositional phrase attachment. In: *Proceedings of
1239 the ARPA human language technology workshop*, pp 250–252
- 1240 Rayner K (1998) Eye movements in reading and information
1241 processing: 20 years of research. *Psychol Bull* 124(3):372–422
- 1242 Roark B, Johnson M (1999) Efficient probabilistic top-down and left-
1243 corner parsing. In: 37th meeting of the ACL, pp 421–428
- 1244 Rolls E (2008) *Memory, attention, and decision-making: a unifying
1245 computational neuroscience approach*. Oxford University Press,
1246 Oxford
- Shastri L, Aijanagadde V (1993) From simple associations to
systematic reasoning: a connectionist representation of rules,
variables, and dynamic bindings using temporal synchrony.
Behav Brain Sci 16:417–494
- Shieber S (1986) *An introduction to unification-based approaches to
grammar*. Center for the Study of Language and Information,
Stanford
- Smolensky P (1988) On the proper treatment of connectionism.
Behav Brain Sci 11(1):1–22
- Smolensky P (1990) Tensor product variable binding and the
representation of symbolic structures in connectionist systems.
Artif Intell 46:159–216
- Tabor W, Tanenhaus M (1999) Dynamical models of sentence
processing. *Cogn Sci* 23(4):491–515
- Tal D, Schwartz E (1997) Computing with the leaky integrate-and-fire
neuron: logarithmic computation and multiplication. *Neural
Comput* 9(2):305–318
- van der Velde F, de Kamps M (2006) Neural blackboard architectures of
combinatorial structures in cognition. *Behav Brain Sci* 29:1–72
- van Gompel R, Pickering R, Traxler M (2001) Reanalysis in sentence
processing: evidence against current constraint-based and two-
stage models. *J Mem Lang* 45:225–258
- Von Der Malsburg C (1981) *The correlation theory of brain function*.
Technical report, Department of Neurobiology, Max-Planck-
Institute for Biophysical Chemistry
- Von Der Malsburg C (1986) Am i thinking assemblies? In: Palm G,
Aertsen A (eds) *Brain theory*. Springer-Verlag, Berlin, pp 161–176
- Wennekers T, Palm G (2000) Cell assemblies, associative memory
and temporal structure in brain signals. In: Miller R (ed) *Time
and the brain: conceptual advances in brain research*, vol 2.
Harwood Academic Publishers, Amsterdam, pp 251–274

UNCORRECTED