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].

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 ____

Springer
the language of science

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	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 (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.	

Cogn Neurodyn
DOI 10.1007/s11571-009-9080-6

RESEARCH ARTICLE

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

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.

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

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

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.

Journal : Large 11571 Dispatch : 9-3-2009 Pages : 14
Article No. : 9080 LE TYPESET
MS Code : CODY99 CP DISK

A range of non-neural connectionist binding mechanisms also exist. Tensor product binding has been used (Smolensky 1990). Recurrent multilayer perceptrons learning via backpropagation (Mikkulainen 1993) have also been used.

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

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

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

Neural and other connectionist parsers

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

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

Another parser (Henderson 1994) uses a connectionist system (Shastri and Aijanagadde 1993) based on associations. These associations use a frame system with dynamic binding via synchrony. It is known that certain constructs, like multiple centre embedded sentences, are difficult for humans to parse. As the number of bindings that the system supports is limited, the system also finds it difficult to parse these types of sentences. Activation decay and simulated annealing have been used to resolve attachment decisions (Kempen and Vosse 1991). One of the problems of most non-neural connectionist parsing models is that there is little notion of time; while such parsers must use word order information, this does not provide timing data.

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

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

Further evidence of the ability to develop parsers using simulated neurons is the CABot1 parser (Huyck and Fan 2007). The CABot1 and CABot2 parsers have many similarities, but the earlier parser uses a stack and there are some indications that the human parser does not (Lewis and Vasishth 2005). Both the CABot parsers make extensive use of Cell Assemblies.

Neuropsychology

Hebb introduced Neuropsychology (Hebb 1949) and provided science with an intellectual bridge between neurons and psychology. One of his key concepts was that of the Cell Assembly (CA).

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

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

Journal : Large 11571 | Dispatch : 9-3-2009 | Pages : 14
Article No. : 9080 | LE | TYPESET
MS Code : CODY99 | CP | DISK

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

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

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

Psychology and linguistics

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

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

Some linguistic theories are almost universally accepted. These include the use of case frames to represent the semantics of a sentence (Filmore 1968) and bar-levels (Jackendoff 1977) to account for simple and complex phrases. Perhaps more importantly, and related to universal grammar, is the notion of a combinatorial system. In this, language is composed of components that can be combined in a tree-like structure that has a practically infinite number of possible topologies. Connectionist systems have been criticised for a lack of compositional syntax and semantics (combinatoriality) (Fodor and Pylyshyn 1988), but this criticism is largely addressed by neural mechanisms used for implementing variable binding (see section "The binding problem").

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

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

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

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

The neural model

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

Journal : Large 11571 | Dispatch : 9-3-2009 | Pages : 14
Article No. : 9080 | ☐ LE | ☐ TYPESET
MS Code : CODY99 | ☑ CP | ☑ DISK

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

$$A_{i_t} = \frac{A_{i_{t-1}}}{D} + \sum_{j \in V_i} w_{ji} \qquad (1)$$

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

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

$$A_{i_t} - F_{i_t} \geq \theta \qquad (2)$$

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

The LIF model is widely used because it is a simple model of a neuron that is relatively accurate biologically. The fLIF model is slightly more complex, and is a slightly better model. A model similar to the one described in this paper (Chacron et al. 2003) has been shown to mimic biological neural responses, particularly with respect to neuronal adaptation, and does provide a more accurate simulation than the simpler LIF models.

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

The CABot2 parser

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

Journal : Large 11571 Dispatch : 9-3-2009 Pages : 14
Article No. : 9080 LE TYPESET
MS Code : CODY99 CP DISK

Overall topology

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

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

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

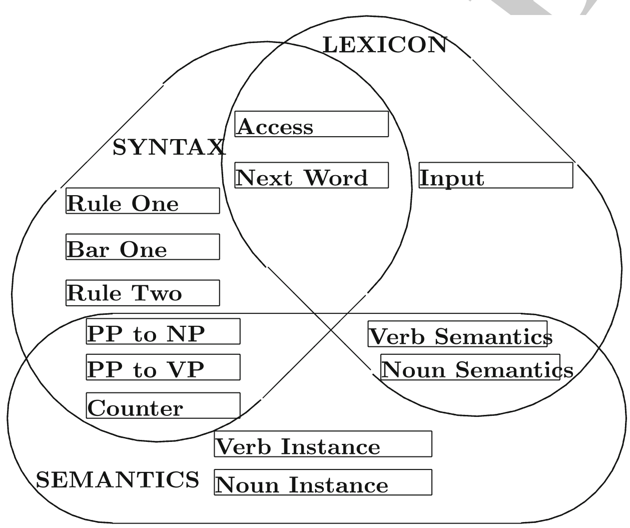

Fig. 1 Gross topology of the CABot2 parser. Each *box* represents a subnet with similar subnets grouped together according to Jackendoff's Tripartite theory

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

Input, access and semantics

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

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

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

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

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

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

Journal : Large 11571 Dispatch : 9-3-2009 Pages : 14
Article No. : 9080 LE TYPESET
MS Code : CODY99 CP DISK

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

Simple rule activation and instantiation

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

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

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

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

Figure 2 shows firing behaviour in the *Verb Instance*, *Rule One* and *Rule Two* subnets. Each dot represents a neuron firing in a particular cycle. The *Verb Instance* neurons are the bottom 500 neurons, and it can be seen that it begins around cycle 65, and persists through to the end of the parse. It can also be seen that different rules ignite at different times.

Instances can be in one of four states. The first is inactive, meaning that no neurons are firing. The second is open, meaning that a simple noun phrase is under construction. Part of X-bar theory states that there is at most one simple phrase currently under construction at any time (Jackendoff 1977). The third and fourth states of instances are the active complex phrase state, and the done state (see section "Complex rules and multi-valued cell assemblies"). When an instance is started, it is open, and this is marked by having a specific feature in the instance firing. When the instance is closed, this feature is turned off.

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

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

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

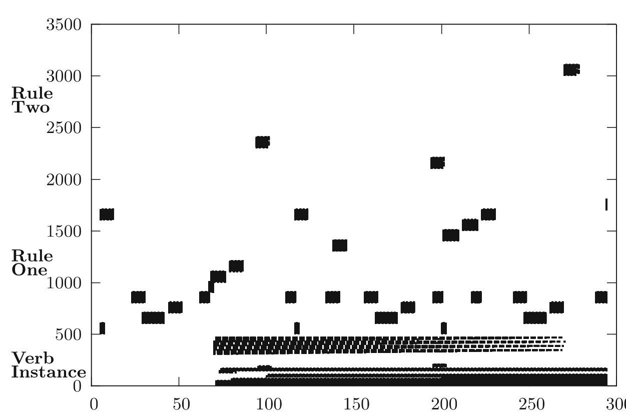

Fig. 2 Rastergram of the verb instance and rule SubNets. The first 500 neurons are verb instances, the next 1,200 from Rule One, and the remainder from Rule Two

Journal : Large 11571 Dispatch : 9-3-2009 Pages : 14
Article No. : 9080 LE TYPESET
MS Code : CODY99 CP DISK

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

As noted in section "Input, access and semantics", *girl* now comes into the *Input* subnet. It then, in collaboration with other subnets, turns on the *girl* CAs in the *Noun Access* and *Noun Semantics* subnets.

As before, the *Word Active* CA is on; in collaboration with the *girl Noun Access* CA, the *NP adds N* rule ignites in the *Rule One* subnet. This turns on the *Main Noun* feature in the open noun instance. This feature is represented by some neurons that learn via STP (see section "The binding problem"). These neurons connect to all of the noun instances, and as the only noun instance that is active is *girl*, this instance is bound to girl after a few cycles of co-firing.

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

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

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

Complex rules and multi-valued cell assemblies

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

A few cycles after the *VP Done* rule for *saw* ceases firing, the *VP → NPactor VP* rule is applied. This rule receives activation from both the verb instance and the noun instance. When it ignites, it firstly turns on the verb's *actor* slot (feature). This slot has neurons that learn via STP, and these neurons have connections to all of the noun instances. The only active noun is the instance that is bound to *the girl*, and so it is bound as the actor after a few cycles. The *actor* slot also turns on the *actor-done* feature which inhibits further application of the rule, and turns off the *actor* slot so that no further binding will occur. Additionally, the noun instance has its *bound* feature turned on, so that it will no longer be used as a slot filler. Note that the application of *VP → NPactor VP* can be seen in Fig. 2; it can be seen at neuron 2,300 near cycle 100. Other rules can be picked out.

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

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

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

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

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

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

Journal : Large 11571 | Dispatch : 9-3-2009 | Pages : 14
Article No. : 9080 | LE | TYPESET
MS Code : CODY99 | CP | DISK

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

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

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

(example 5) I saw with the telescope.

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

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

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

Results

The CABot2 parser is not capable of parsing all English sentences, but it does parse several common constructs correctly. It is a relatively capable parser which can handle the basic commands that are needed within the CABot2 computer game environment and produce correct semantic output. More importantly, it is based on a neural model with a link to biological time, and parses in times that are similar to human timing data. It resolves PP attachment ambiguity in a way that appears to be similar to the way humans resolve these ambiguities, and demonstrates one way that semantics can be involved in making parsing decisions. Finally it can be readily incorporated into a neural agent, and thus can make use of evidence that is not normally available to other computational parsers, but is available to the human parser.

Semantic output

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

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

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

Timing

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

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

Journal : Large 11571 Dispatch : 9-3-2009 Pages : 14
Article No. : 9080 ☐ LE ☐ TYPESET
MS Code : CODY99 ☑ CP ☑ DISK

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

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

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

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

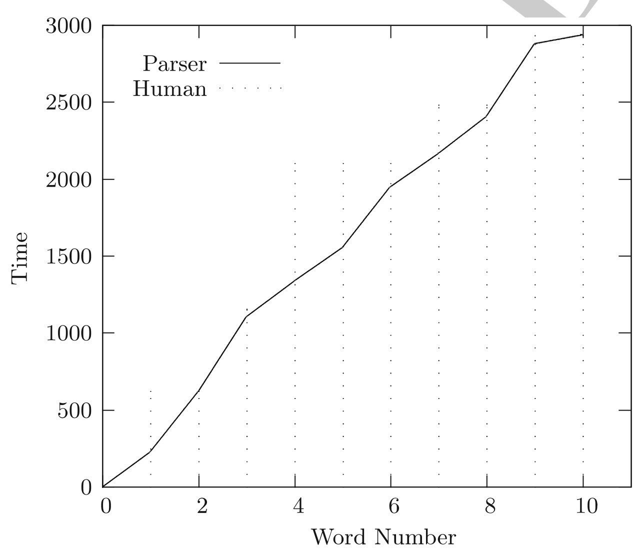

Fig. 3 Time spent to parse by word the sentence *The girl saw the dangerous pyramid with the stalactite*

PP-attachment

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.

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

Journal : Large 11571 Dispatch : 9-3-2009 Pages : 14
Article No. : 9080 □ LE □ TYPESET
MS Code : CODY99 ☑ CP ☑ DISK

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

Finally, the sixth and seventh sentences attach the PP to the verb. The difference here is that they differ on the NP instead of on the PP. This shows that inheritance works on different elements, even though the elements are differently weighted.

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

CABot

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

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

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

Various minor versions of the agents have been developed to explore a range of capabilities, and two versions are particularly relevant to this paper. In one version of CABot1, the labels of some visual semantic categories were learned by presenting them simultaneously with visual instances of the category. This labelling is a portion of the solution to the symbol grounding problem (Harnad 1990). Similarly, a second variant of CABot1 used an item in the visual field to resolve the referent of the command *Turn toward it*, showing the agent supports pronoun resolution by context.

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

Discussion and conclusion

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

As is almost certainly the case with people, prepositional phrase attachment ambiguity is resolved by semantics. In the cases where the attachment is known, it performs flawlessly, that is, the system is capable of storing pre-calculated decisions. Moreover, it is capable of handling novel attachment decisions due to the hierarchical nature of the stored semantics and their activation of attachment preference rules. This use of the four-tuple (verb, noun, preposition, noun) has been shown to be

Journal : Large 11571 Dispatch : 9-3-2009 Pages : 14
Article No. : 9080 □ LE □ TYPESET
MS Code : CODY99 ✔ CP ✔ DISK

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

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

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

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

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

These improvements and expansions will run into a simulation boundary. The CABot2 parser has 30,000 neurons and systems with 100,000 fLIF neurons have been simulated in real-time on a standard PC, where real-time means a cycles takes 10 ms to simulate, or 100 cycles take about a second. Expansion beyond 100,000 neurons has radically slowed simulations. These sizes could be improved by improved hardware, distributing the simulator across PCs, or a more efficiently coded simulator, but it is expected that special neural hardware (Khan et al. 2008) will be available within a year or two. This should enable simulations in real-time of a billion neurons.

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

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

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

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

Other linguistic systems, like a lexical system, phonetics, or discourse interpretation, or systems for production of all of these, could be developed and integrated with the parser. A lexical system could be used to resolve lexically ambiguous and polysemous words like *saw*. It is expected that these efforts would be of a similar degree of complexity to parser development but would be made easier by the skills, techniques, and knowledge already gained. Crucially, while these systems would be largely independent according to the tripartite theory, they would function in parallel. Thus the full system would process at roughly the same, simulated, speed.

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

Journal : Large 11571 Dispatch : 9-3-2009 Pages : 14
Article No. : 9080 ☐ LE ☐ TYPESET
MS Code : CODY99 ☑ CP ☑ DISK

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

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

It does appear that the CABot2 parser is a reasonable cognitive model. If so, then this is proof that Smolensky's claim is out of date and that neural models are now capable of being used for sophisticated cognitive modelling. More importantly, these neural cognitive models may be able to address new problems that symbolic and non-neural connectionist systems cannot, such as timing, word coding, and the neural implementation of memory, both short and long-term. The link to neural data may also provide simple solutions to problems that are otherwise difficult to solve. Neural models also can solve the symbol-grounding problem that cause problems for symbolic systems. It therefore seems reasonable to expect that the development of these models will lead to better AI systems.

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

References

Abeles M, Bergman H, Margalit E, Vaddia E (1993) Spatiotemporal firing patterns in the frontal cortex of behaving monkeys. J Neurophysiol 70(4):1629–1638

Amit D (1989) Modelling brain function: the world of attractor neural networks. Cambridge University Press, Cambridge

Anderson J, Lebiere C (1998) The atomic components of thought. Lawrence Erlbaum, Mahwah

Anderson J, Lebiere C (2007) How can the human mind occur in the physical universe. Oxford University Press, Oxford

Belavkin R, Huyck C (2008) The emergence of rules in cell assemblies of flif neurons. In: Proceedings of the eighteenth European conference on artificial intelligence

Bevan M, Wilson C (1999) Mechanisms underlying spontaneous oscillation and rhythmic firing in rat subthalamic neurons. Neuroscience 19:7617–7628

Buonomano D (1999) Distinct functional types of associative long-term potentiation in neocortical and hippocampal pyramidal neurons. Neuroscience 19(16):6748–6754

Carpenter G, Grossberg S (1988) The art of adaptive pattern recognition by a self-organizing neural network. IEEE Comput 21:77–88

Chacron M, Pakdaman K, Longtin A (2003) Interspike interval correlations, memory, adaptation, and refractoriness in a leaky integrate-and-fire model with threshold fatigue. Neural Comput 15:253–278

Chomsky N (1965) Aspects of the theory of syntax. MIT Press, Cambridge

Dayan P, Abbott L (2005) Theoretical neuroscience: computational and mathematical modeling of neural systems. MIT Press, Cambridge

Fan Y, Huyck C (2008) Implementation of finite state automata using flif neurons. In: IEEE systems, man and cybernetics society, pp 74–78

Filmore C (1968) The case for case. In: Back E, Harms R (eds) Universals in linguistic theory. Holt, Rinehart and Winston Inc., New York

Fodor J, Pylyshyn Z (1988) Connectionism and cognitive architecture: a critical analysis. Cognition 28:3–71

Ford M, Bresnan J, Kaplan R (1982) A competence-based theory of syntactic closure. In: Bresnan J (ed) The mental representation of grammatical relations. MIT Press, Cambridge

Harnad S (1990) The symbol grounding problem. Phys D 42:335–346

Hebb DO (1949) The organization of behavior. Wiley, New York

Hempel C, Hartman K, Wang X, Turrigiano G, Nelson S (2000) Multiple forms of short-term plasticity at excitatory in rat medial prefrontal cortex. J Neurophysiol 83:3031–3041

Henderson J (1994) Connectionist syntactic parsing using temporal variable binding. J Psycholinguist Res 23(5):353–379

Hodgkin A, Huxley A (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol 117:500–544

Huyck C (1999) Modelling cell assemblies. Technical report, Middlesex University

Huyck C (2007) Creating hierarchical categories using cell assemblies. Connect Sci 19(1):1–24

Huyck C (2008) CABot1: a videogame agent implemented in fLIF neurons. In: IEEE systems, man and cybernetics society, pp 115–120

Huyck C, Belavkin R (2006) Counting with neurons: rule application with nets of fatiguing leaky integrate and fire neurons. In: Proceedings of the seventh international conference on cognitive modelling, pp 142–147

Huyck C, Fan Y (2007) Parsing with fLIF neurons. In: IEEE systems, man and cybernetics society, pp 35–40

Jackendoff R (1977) X-bar syntax: a study of phrase structure. MIT Press, Cambridge

Jackendoff R (2002) Foundations of language: brain, meaning, grammar, evolution. Oxford University Press, Oxford

Kaplan S, Sontag M, Chown E (1991) Tracing recurrent activity in cognitive elements(TRACE): a model of temporal dynamics in a cell assembly. Connect Sci 3:179–206

Kempen G, Vosse T (1991) Incremental syntactic tree formation in human sentence processing: a cognitive architecture based on activation decay and simulated annealing. Connect Sci 1: 275–292

Khan M, Lester D, Plana L, Rast A, Painkras J, Furber S (2008) SpiNNaker: mapping neural networks onto a massively-parallel chip multiprocessor. In: Proceeding of 2008 international joint conference on neural networks, pp 2850–2857

Knoblauch A, Markert H, Palm G (2004) An associative model of cortical language and action processing. In: Proceedings of the ninth neural computation and psychology workshop, pp 79–83

Lewis R, Vasishth S (2005) An activation-based model of sentence processing as skilled memory retrieval. Cogn Sci 29(3):375–419

Journal : Large 11571 | Dispatch : 9-3-2009 | Pages : 14
Article No. : 9080 | ☐ LE | ☐ TYPESET
MS Code : CODY99 | ☑ CP | ☑ DISK

MacDonald M, Pearlmutter N, Seidenberg M (1994) Lexical nature of syntactic ambiguity resolution. Psychol Rev 101(4):676–703

McCulloch W, Pitts W (1943) A logical calculus of ideas immanent in nervous activity. Bull Math Biophys 5:115–133

Mikkulainen R (1993) Subsymbolic natural language processing. MIT Press, Cambridge

Miller G (1990) WordNet: an on-line lexical database. Int J Lexicogr 3(4):235−312

Milward O, Cooper R (1994) Incremental interpretation: applications, theory and relationships to dynamic semantics. In: 15th international conference of computational linguistics, pp 748–54

Nadh K, Huyck C (2009) Prepositional phrase attachment ambiguity resolution using semantic hierarchies. In: Proceedings of Artificial Intelligence and Applications (AIA 2009). Innsbruck, Austria

Nakov P, Hearst M (2005) Using the web as an implicit training set: application to structural ambiguity resolution. In: Proceedings of HLT-NAACL

Passmore P, Huyck C (2008) Models of cell assembly decay. In: IEEE systems, man and cybernetics society, pp 251–256

Pollard C, Sag I (1994) Head-driven phrase structure grammar. Center for the Study of Language and Information, Stanford

Pulvermuller F (1999) Words in the brain's language. Behav Brain Sci 22:253–336

Pulvermuller F (2000) Syntactic circuits: how does the brain create serial order in sentences? Brain Lang 71:194–199

Ratnaparkhi A, Reynar J, Roukos S (1994) A maximum entropy model for prepositional phrase attachment. In: Proceedings of the ARPA human language technology workshop, pp 250–252

Rayner K (1998) Eye movements in reading and information processing: 20 years of research. Psychol Bull 124(3):372–422

Roark B, Johnson M (1999) Efficient probabilistic top-down and left-corner parsing. In: 37th meeting of the ACL, pp 421–428

Rolls E (2008) Memory, attention, and decision-making: a unifying computational neuroscience approach. Oxford University Press, 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

Journal : Large 11571 Dispatch : 9-3-2009 Pages : 14
Article No. : 9080 ☐ LE ☐ TYPESET
MS Code : CODY99 ☑ CP ☑ DISK