Technical Report OUCS-2011-01

Mapping sensorimotor sequences to word sequences: A connectionist model of language acquisition and sentence generation

Authors:

Martin Takac, Lubica Benuskova, Alistair Knott
Department of Computer Science, University of Otago, New Zealand

Status:

Currently under review in Cognition journal.

Department of Computer Science,
University of Otago, PO Box 56, Dunedin, Otago, New Zealand

http://www.cs.otago.ac.nz/research/techreports.php

Mapping sensorimotor sequences to word sequences: A connectionist model of language acquisition and sentence generation

Martin Takac*, Lubica Benuskova, Alistair Knott

Dept. of Computer Science, University of Otago, PO Box 56, Dunedin 9054, New Zealand

Abstract

In this article we present a neural network model of sentence generation. The main technical novelty in the model is in its semantic representations: the 'messages' which form the input to the network are structured as *sequences*, which are delivered to the network one at a time. Rather than learning to linearise a static semantic representation as a sequence of words, our network rehearses a sequence of semantic signals, and learns to generate words from selected signals. Our use of sequences to encode semantic representations has several benefits, both conceptual and technical. Conceptually, the use of rehearsed sequences of semantic signals connects to work in embodied cognition, which posits that the structure of semantic representations has its origin in the serial structure of sensorimotor processing. It also connects to nativist models of language development: we argue that some of the linguistic universals proposed within Chomskyan models of syntax can be interpreted as reflections of sensorimotor processing. Technically, the use of sequentially structured semantic representations permits a novel answer to the question of how a neural network can learn genuinely abstract syntactic rules (a vexed question in connectionist models of language). Equally importantly, it supports a way of using abstract syntactic rules in combination with rules about surface patterns in language. In summary, sequentially structured semantic representations allow a neural network model which combines elements from nativist, empiricist and embodied theories of language in a novel way.

Keywords: Sentence generation, Language acquisition, Minimalism, Embodied cognition, Neural networks

[*] Corresponding author. Address: Dept. of Computer Science, University of Otago, PO Box 56, Dunedin 9054, New Zealand. Tel: +64 3 4795691, Fax: +64 3 4798529

Email address: takac.martin@gmail.com (Martin Takac)

1. Introduction

Connectionist models of language simulate infants acquiring language: they are exposed to utterances from a particular language, and they learn the patterns in this language. In a connectionist model, mature language is seen as the product of a learning algorithm, with all the messiness and complexity which this entails, rather than as a tidy set of high-level rules. While the learning algorithm itself must be presumed to be innate, connectionist models tend to be empiricist in spirit, assuming that most linguistic knowledge is knowledge gained by learning the rich patterns in the exposure language.

There are several controversial questions which must be answered when building a connectionist model of language. Firstly, what are the patterns in language which the system must learn? There are many kinds of pattern in language. Some patterns are defined in the surface form of utterances. The surface patterns we will focus on in this paper are patterns of whole words. At their most concrete, surface word patterns consist of idioms or fixed expressions (e.g. *let the cat out of the bag*, *Winnie the Pooh*), but they can also take the form of statistical tendencies, of the kind which can be captured by n-gram probabilistic language models (see e.g. Jelinek & Mercer, 1980; Chen & Goodman, 1998). In connectionist linguistics, there are well-known architectures which can learn such probability models. The most familiar of these is the **simple recurrent network** (**SRN**; Elman, 1990): a network which is presented with word sequences as training data, and learns to predict the next word in a sequence using a hidden layer of units containing recurrent connections. Unsurprisingly, SRNs can learn idioms and fixed expressions as well as probability distributions conditioned by grammatical rules (see Takac et al., 2010).

Other patterns in language are more abstract. These are the patterns which have traditionally been modelled as grammatical rules or principles. There is considerable debate amongst connectionists about how these patterns should be modelled. An early connectionist claim was that simple recurrent networks designed to learn common surface patterns develop internal representations which capture abstract patterns in language quite well, at least over short distances (see e.g. Elman, 1990; Christiansen & Chater, 1999; Cernansky et al., 2007). This idea has considerable power, but there is now a consensus that some extension to a simple word-sequencing device is needed to model the discreteness and productivity of syntactic patterns (see e.g. Marcus, Marcus, 2001; Pulvermüller & Asollahi, 2007). The basic problem with simple recurrent networks by themselves is that they are bad at predicting word sequences which they have rarely or never seen, even if these conform well to abstract grammatical rules. In response, many recent connectionist models maintain the idea that grammatical patterns can be modelled as sequences, but further stipulate that these sequences can be of abstract units rather than just of words. There are connectionist models which can learn sequences of semantic roles (Chang, 2002), of abstract word classes (Pulvermüller, F & Knoblauch, 2009). or multi-word phrasal units (Dominey et al., 2003, 2006). All of these more elaborate models require some specialised machinery to supplement simple word-sequencing machinery. To distinguish between these models, several criteria can be used. The most obvious of these are the criteria linguists use to evaluate declarative grammars: a model should have good coverage (it should be able to accurately reproduce some fragment of a natural language) and it should be parsimonious (minimise the amount of machinery which is stipulated). But there is another important criterion: ideally the network model of grammar should make contact with a body of grammatical theory. Linguists have sophisticated models of grammar, many of which have been in development for decades; these should provide a rich source of information for connectionist implementations. If a network's representations can be interpreted using terminology from an existing grammatical paradigm, then the insights gained by linguists working in that paradigm should help guide further development of the network.

A second controversial question for connectionist modellers concerns how much information about the patterns in language is present innately. Since surface patterns involving specific words must unquestionably be learned, this question only relates to the higher-level 'grammatical' patterns in language. As just mentioned, different connectionist models of grammatical paterns use different machinery, and so embody different assumptions about innate knowledge. However, the main debate here pits connectionist linguists against nativist theoretical linguists, mainly those working within the Chomskyan tradition. Connectionist linguists rely on powerful learning architectures to acquire many of the patterns in language. By contrast, the Chomskyan model of learning is rather simple: there is less to learn. Chomskyan models like Government-and-Binding (Chomsky, 1981) and Minimalism (Chomsky, 1995) propose that a child's innately specified 'language acquisition device' imposes fairly strict constraints on the space of learnable languages, so that learning the syntax of a particular language amounts to fixing the values of a relatively small number of parameters. These parameter-based models are attractive, in that they provide elegant accounts of some of the basic syntactic differences between languages. However, they give no account at all of the actual mechanism which learns how to set the values of parameters. Chomskyan theories aim to model knowledge of syntax at a level which abstracts away from processing issues, but it is hard to see how a computational model of parameter-setting can be defined except as part of a model of language processing. In addition, Chomskyan theories have great difficulty representing surface patterns in language—in particular the complex ways in which these patterns interact with productive grammatical patterns. For instance, a complete account of idioms must represent them as structures containing grammatical elements as well as surface-based patterns (see Jackendoff, 2002 for some convincing arguments). Recent 'empiricist' models of language (Pollard & Sag, 1994; Goldberg, 1995; Jackendoff, 2002) see this model of idioms as characteristic of language in general. They assume that the central unit of structure in language is the **construction**: a structure combining grammatical and surface-based elements in almost arbitrary ways. Even if some of our knowledge of grammatical structure must be assumed to be innate, the way that grammatical patterns combine with surface language patterns must of course be learned by exposure to language data. The challenge for connectionist linguists is to model how the brain can represent, and learn, such structures.

In summary, language contains surface patterns and also higher-level grammatical patterns. While there is a consensus that simple recurrent networks are good at learning surface patterns, there is debate among connectionist linguists about how to encode higher-level grammatical generalisations in neural networks. There is also debate between connectionist linguists and nativist linguists about how much linguistic knowledge (of the high-level kind) is learned from exposure to language data. Nativist models assume that grammatical learning is quite heavily constrained by innate knowledge, and consequently these models give parsimonious accounts of the grammatical differences between languages. But they have no account of surface patterns, and the intricate ways these interact with grammatical structures. Connectionist models can give a good account of surface patterns, and offer good scope for an account of their interactions with grammatical patterns. But this account is dependent on finding a workable representation of grammatical patterns.

In this article, we will describe a neural network model which can learn both surface-based ('idiomatic') and abstract ('grammatical') patterns in the exposure language, and which makes a proposal about how these two types of pattern interact. The model is trained on pairs of sentences and their associated meanings, and learns a function which maps meanings onto sentences—in other words it is a model of sentence generation. The network's representation of grammatical patterns is somewhat unusual for a connectionist model, because it draws on some Chomskyan ideas about innate linguistic knowledge and parameter setting. But while it makes contact with a Chomskyan account of syntactic competence, it is still squarely a model of language processing, most of which should be familiar to connectionist linguists. In particular, it makes use of a SRN to learn a probabilistic model of surface language structures, and for various other purposes.

The key innovation in the network is in its representation of sentence meaning. The meaning of a sentence is modelled as a sequence of semantic representations, rather than as a static assembly of active units. The structure of a semantic message is expressed by having representations occupying particular semantic roles (e.g. 'agent', 'patient', 'action') appear at specified serial positions in the sequence. We will begin in Section 2 by introducing this idea in some detail, and providing motivation for it. In Sections 3–5 we present a neural network model of sentence generation, which learns to map semantic representations in the above format onto utterances. The network's use of sequences to represent sentence meanings has two benefits. Firstly, it allows syntactic patterns to be represented and learned using a rather simple mechanism. We describe this mechanism in Section 3. Secondly, it supports an account of the relationship between abstract syntactic patterns and surface patterns in language: the notion of sequences provides a common currency for modelling these two kinds of pattern. In Section 4 we describe some extensions to the model introducing a SRN-like mechanism which can learn surface linguistic patterns. In Section 5 we describe the complete model, which learns a mixture of surface and abstract linguistic patterns. In Section 6 we describe some experiments with the complete model, and discuss their results. The model is able to learn a training language containing both abstract rules and idiomatic surface forms; moreover, its learning progresses through developmental stages which bear some resemblance to stages identified in models of language development in infants. In Sections 7 and 8 we give a general discussion of the model and some conclusions.

2. Semantic representations as sequences

2.1. Connectionist representations of episodes: some existing proposals

A connectionist model of language must employ a scheme for representing sentence meanings. Sentences report events or states—we will use the term **episodes** to cover both cases. So what is needed is a connectionist representation of episodes. Schematically, we can represent an episode as a number of semantic objects, each of which is associated with a particular semantic role. For instance, an episode involving an action can be defined as having an ACTION role, which is occupied by an action type (e.g. CHASE, RUN), together with a number of other roles required by this action type (e.g. CHASE requires an AGENT and a PATIENT while RUN only requires an AGENT).[1] There are many alternative accounts of roles, but as far as connectionist implementations are concerned, the main problem is the same for all of them: a way must be found to associate particular semantic objects with particular roles. For instance, the episode in which John chases Mary must be represented by associating the semantic object JOHN with the AGENT role and the semantic object MARY with the PATIENT role. Simply activating the objects JOHN, MARY and CHASE will not do, because it does not specify who is doing what. To compound the problem, the binding scheme must support the creation of hierarchical representations, because semantic objects can have their own internal structure. For instance, participants in an action may have particular specified properties, so there must be bindings between participants and properties.

Many connectionist schemes have been proposed for binding semantic objects to roles. There are schemes which allow for synchronised activation of assemblies at different phases in a clock cycle, in which role assemblies and filler assemblies are bound together by being activated in the same phase (Shastri & Ajjanagadde, 1993). There are schemes which link all possible roles to all possible fillers, and model particular bindings by enabling particular links; see e.g. Chang (2002), and more elaborate schemes which employ a finite set of specialised binding assemblies to implement these links, which can feature as fillers in their own right, permitting hierarchical structures (van der Velde & de Kamps, 2006). There are schemes using distributed encodings that exploit the sparsity of high-dimensional vector spaces, which also allow a limited degree of hierarchy (see e.g. Plate, 2003). All of these schemes have merits and drawbacks, relating to their expressiveness and neural plausibility. We will not discuss these in any detail. Our main concern is to introduce another scheme,

[1] Elements of semantic representations are given in SMALL CAPS.

which is motivated from a new perspective, and has some novel merits, both technically (as a platform for connectionist sentence processing architectures) and theoretically (as a point of contact with existing models of syntax).

2.2. A new proposal: sensorimotor sequences

Our scheme is motivated from the perspective of 'embodied cognition'. Embodied models of cognition start from the observation that high-level cognitive processes take place in agents situated in a physical world, whose experiences are heavily shaped by their sensory and motor interactions with this world, and draw the conclusion that high-level cognitive representations are likely to reflect properties of the sensorimotor system. This idea has been pursued in several directions; see e.g. Harnad (1990); Brooks (1991); Ballard et al. (1997); Glenberg & Kaschak (2002); Feldman & Narayanan (2004); Barsalou (2008). In each case, the idea is to look to the sensorimotor system for ideas about how to model high-level semantic representations. There are many arguments for this basic idea. One concerns evolutionary processes. High-level 'specifically human' cognitive capacities such as the capacity for language developed relatively late in human evolution, at a time when our sensorimotor system had already reached something like its modern form (as evidenced by the strong similarities between our sensorimotor systems and those of our nearest evolutionary neighbours; see e.g. Tootell et al. (1996); Iacoboni (2006). Evolution works by making small changes to an existing design; see Anderson (2010) for a recent exposition of this principle focussing on brain evolution. Therefore it is possible that elements of the sensorimotor system were reused or adapted in the evolution of language. A related argument is from theoretical parsimony. Sensorimotor routines must somehow deliver high-level semantic representations, because we can talk about what we see and do. The more strongly semantic representations supervene on sensorimotor representations, the easier it is to give an account of how this happens.

2.2.1. Ballard et al.*'s concept of deictic routines*

The particular semantic representation scheme we propose picks up on the observation that at a timescale of around one-third of a second, interactions with the world often take the form of short sequences of sensorimotor operations, with well-defined internal structure. These sequences were first noticed by Ballard et al. (1997), who coined the term **deictic routines** to describe them. The key structuring elements in a deictic routine are 'deployments of attention'—most concretely, saccades (i.e. discrete eye movements). Vision is an active process; we make around three saccades per second. Each saccade evokes transitory representations in the sensorimotor areas of the brain, which can be used to plan the next saccade, or to plan some other motor action, which will have attentional consequences of its own. Hence a deictic routine involves a chain of attentional or motor operations, interleaved with transitory sensory consequences. Ballard *et al.* suggest that deictic routines may feature in higher-level cognitive processing as well as in direct interaction with the world—and may even be an organising principle for such processing.

2.2.2. Knott's model of sensorimotor sequences

The sensorimotor process we have focussed on is that involved in experiencing an elementary transitive action: reaching to grasp a target object. This process has been studied in detail by Knott (in press), who argues that it is structured as a deictic routine. Knott reviews a wide range of experiments investigating the execution and perception of reach-to-grasp actions, and concludes that both these processes involve the same canonical sequence of sensorimotor operations. In the canonical sequence, the first two operations are actions of attention: first, an action of attention to the agent of the action, and then an action of attention to the intended target. Each action of attention delivers a transitory sensory representation which enables the next action to be executed. When the observer attends to the agent, he evokes a sensory representation of the agent, which provides information about the location of the intended target. (If the observer is watching someone else performing the reach action, the observed agent's gaze and posture provide information about the intended target. If the observer is the agent himself, this information is provided by the mechanism which actually selects the target, namely his own motor system which activates the target location in a motor coordinate system centred on his own hand; see e.g. Tipper et al. (1992). In either event, the observer can then execute a saccade to the intended target object, and to evoke a detailed representation of this object. At this point—but not before—the observer can activate a motor programme. (If the observer is watching someone else, this activation is due to a specialised system in the superior temporal sulcus which categorises observed biological motions; see e.g. Jellema et al. (2000); Pelphrey et al. (2005). If the observer himself is the agent, it is again his own motor system which activates the motor programme, by computing the grasp affordances of the observed target; see e.g. Fagg & Arbib (1998); Murata et al. (2000). In either event, the observer is able to activate a motor programme. Once this happens, the mode of sensorimotor processing changes. While the two initial attentional operations are discrete, action monitoring is a temporally extended process, which has the character of a dynamical system (see Jordan & Wolpert, 2000 for action execution and Oztop & Arbib, 2002 for action observation). Finally, at the endpoint of action monitoring, a discrete attentional state is again reached, when the agent has a stable grasp on the target object. The basic sequence of operations, therefore, is: attention to agent, attention to intended target, action monitoring.

In addition to this basic sequence, Knott's model of reach-to-grasp actions has two more components. Firstly, Knott proposes that the agent and the target are each attended to *more than once* in the deictic routine involved in experiencing the action. The agent is initially attended to as a salient object in the world, and classified as an object. Knott argues that in the action-monitoring phase, the agent is reattended to *as an agent*; i.e. as a animate entity with characteristic patterns of motion. Our concepts of objects and agents are intimately connected; we classify or recognise animate objects by their shape, but also by their movement. Knott argues that it is because these two conceptions are axiomatically brought together in sensorimotor action-monitoring routines that we are able to form these cross-modal representations of agents. Knott makes a similar proposal about targets. The target of a reach-to-grasp action is initially attended to as an object in the world, and visually classified as an object. But at the endpoint of the action, when the agent has achieved a stable grasp, the target object is reattended to *as a motor state*—the current state of the agent's hand/arm. A reach-to-grasp action is not only a substantive motor operation, but also an action of attention, which delivers information about the target in the haptic modality. This information arrives towards the end of the action, when the hand is close enough to touch the object, and is most accurate when the agent has achieved a stable grasp on the target. As well as representing (animate) objects as agents, we must be able to represent (manipulable) objects as motor routines, and learn functions which map visually perceived shapes onto the motor programmes needed to interact with them. Again, reach-to-grasp routines are axiomatically involved in learning these functions, and forming the necessary cross-modal representations of target objects. In summary, the fact that the agent and patient each feature twice in a reach-to-grasp deictic routine is an important component of an account of the development of mature cross-modal object concepts.

Secondly, Knott supplements his model of sensorimotor routines with a model of working memory. 'Episode representations' must be more than transient sensorimotor sequences; they must be things that can be stored. Knott proposes that an observer can store recently experienced sensorimotor sequences in working memory, in a format which allows them to be internally rehearsed—a proposal which echoes many similar ideas in embodied cognition (see e.g. Gallese & Goldman, 1998; Jeannerod, 2001; Grèzes & J, 2001; Barsalou et al., 2003; Feldman & Narayanan, 2004). There has been much recent research into the neural basis of prepared sensorimotor sequences. A consensus is that stored sequences are maintained in prefrontal cortex (see e.g. Averbeck et al., 2002; Rhodes et al., 2004; Tanji et al., 2007). An interesting discovery is that the neural assemblies which store a planned sequence of sensorimotor operations include representations of the individual operations involved. These individual representations of the planned actions are active in parallel in prefrontal cortex, both in advance of the sequence being executed, and during its execution, and after it is complete (Averbeck et al., 2002; Averbeck & Lee, 2007). What this means is that during the simulated *replay* of a sequence of sensorimotor operations stored in working memory, there are tonically active representations of all the operations in the sequence in prefrontal cortex, which supplement the sequence of transiently active representations in the sensorimotor system.

The complete sequence of signals in Knott's account of a rehearsed reach-to-grasp deictic routine is shown in Table 1. Note firstly that there is a mixture of transient and sustained signals. Note also that each step in the routine has the same basic structure. There is an **initial context**, in which a sensory or motor **operation** takes place, which generates a **reafferent signal** (a sensory representation), and establishes a **new context**. The routine is naturally recursive ('tail-recursive', to be precise), because the new context of one operation

Table 1: The time course of signals occurring during the replay of the cup-grabbing deictic routine from working memory

Sustained signals	Transient signals			
	Initial context	Operation	Reafferent signal	New context
$plan_{attend\_agent,attend\_cup,grasp}$	C_1	*attend_agent*	*agent_rep*	C_2
$plan_{attend\_agent,attend\_cup,grasp}$	C_2	*attend_cup*	*cup_rep*	C_3
$plan_{attend\_agent,attend\_cup,grasp}$	C_3	*grasp*	*agent_rep*	C_4
$plan_{attend\_agent,attend\_cup,grasp}$	C_4		*cup_rep*	

constitutes the initial context of the next one. (The recursion 'bottoms out' in Context C_4, which is the stable grasp state.) Note finally that there are two types of repeated signal in the sequence. Planning representations are repeated at every iteration, because they are tonically active. But agent and patient representations are repeated at particular iterations. For detailed motivation of all aspects of this model of reaching-to-grasp, see Knott (in press).

To return to the language-processing network: the 'semantic representations' which we will present to our network will take the form of sequences with the kind of structure shown in Table 1. Our suggestion is that a speaker needs to internally 'replay' a stored episode representation in order to express it verbally: planned sensorimotor sequences by themselves cannot generate the right kind of verbal side-effects. In the rest of this section, we will discuss some of the interesting consequences of thinking of semantic representations in this way.

2.3. Some benefits of using sensorimotor sequences as semantic representations

Modelling semantic representations as sensorimotor sequences has several advantages in a connectionist language-processing architecture. We will summarise the main advantages here.

Firstly, having a connection to a sensorimotor model provides an independent justification for the form of semantic representations. They are not just plucked out of the air: an account can be given of how the representations are delivered by experience. This recalls one of the advantages of embodied models of cognition mentioned earlier: if semantic representations supervene directly on the sensorimotor system, it is relatively easy to describe the interface between the two of them. Semantic representation schemes involving synchrony, binding assemblies, distributed encodings etc must envisage a more complex interface with the sensorimotor system, whose nature is (often tacitly) understood as a separate problem. Our representation scheme emerges naturally out of a model of the sensorimotor system.

Secondly, there are several technical advantages to thinking about semantic representations as sequences. One of these has to do with types of abstraction which the semantic scheme has to support. Any connectionist sentence-processing architecture has to find a representation scheme which allows a lexicon of word meanings to be learned independently of the semantic role which words play, so that words encountered in one role can be understood (and used) in other roles. We would not want a scheme where the word expressing the concept DOG has to be learned separately when it appears as an agent and as a patient. In our sequence-based representation scheme, we only need one medium for representing objects: information about the role an object plays is conveyed by its position in the sensorimotor sequence.[2] Mappings between words and meanings are therefore completely decoupled from information about semantic roles. (In fact, all the representation schemes mentioned earlier achieve this decoupling in one way or another.) Another potential technical advantage of sequentially structured semantic representations is that surface language itself is sequentially structured. Architectures which envisage static semantic representations must solve a linearisation problem: a static message must be mapped to a sequence of words. In fact, there are several reasons to think that sentence generation really does require parallel representations to be linearised: for instance, speech errors which swap or blend words which should appear at different positions in a sentence (see Fromkin, 1973 and much subsequent work). Connectionist models can trade on parallel representations to simulate such errors (well-known models include Dell, 1986; Burgess & Hitch, 1999). We certainly want to make use of parallel representations in our model of sentence generation—indeed, in our model an episode representation in working memory is a set of semantic items active in parallel. However, because these episode representations take the form of planned sequences, there is also a 'natural' way of linearising them which a sentence generation architecture may be able to exploit. One simple idea is that sentence generation just involves playing a sequence of semantic items to an interface which maps semantic items onto words. In fact this simple idea is one of the core mechanisms in our network.

Finally, pursuing the point just made: if we think about semantic representations as planned sensorimotor sequences, and if sentence generation involves generating linguistic side-effects of sensorimotor sequences, then we may be able to attribute some of the *syntactic* properties of the generated sentences to the structure of rehearsed sensorimotor sequences. We will discuss this idea by itself in the following section.

2.4. A syntactic interpretation of sensorimotor sequences

The simple idea just mooted is that if semantic representations are replayable sequences, a sentence generation architecture could take a replayed sensorimotor sequence as input—i.e. receive items from the sequence one by one—and

[2] We assume there are only two basic semantic roles—roughly speaking, the 'proto-agent' and 'proto-patient' roles proposed by Dowty (1991). The proto-agent is the participant attended to first, and the proto-patient is the participant attended to second. However, we will continue to use the terms 'agent' and 'patient'.

generate appropriate linguistic reflexes of these items as they arrive. Of course this idea needs to be refined in several ways. But many of these refinements can be interpreted in relation to models of theoretical syntax. In this section we will consider some of them.

2.4.1. The position of noun phrases

Perhaps most obviously, the agent and patient of our example reach-to-grasp action are each transiently active at two different positions in the sequence. Of course, a transitive sentence only features the agent and patient once each (at least as full noun phrases). So the system which reads out semantic items onto words must learn to suppress one occurrence of each word. The possibility then arises that different languages have different conventions about which occurrence is pronounced. The structure of the sensorimotor sequence involved in grasping a cup must be assumed to be the same for speakers of all languages. But as is well known, different languages have different canonical word orders: SVO, VSO and so on. We can imagine a linguistic interface with some plasticity, which can learn a pattern of suppression able to reproduce the surface form of sentences in the exposure language. Note that learning in this type of system is quite constrained. The structure of the sensorimotor sequence is the same for every language, and does not need to be learned. Acquiring an ordering convention for a particular language is a matter of choosing between a small number of possible alternatives made available by this sequence, rather than of constructing a pattern from scratch. Note also that in this scheme there is some notion of an 'underlying' sentence structure, which is the same for all languages, in which both agent and patient feature in two positions. These ideas are all reminiscent of a Chomskyan model of syntax. To make this point, we will briefly sketch a recent Chomskyan account of transitive sentences, within the Minimalist framework of Chomsky (1995).

In Minimalism, a sentence must be represented at two levels of syntactic structure: a level of surface form called 'phonetic form' (PF), and an underlying form called 'logical form' (LF). PF encodes the surface word order of the sentence, and thus varies considerably from language to language. LF is relatively invariant over translations (at least for simple concrete sentences). It is understood as the level of syntactic representation which 'interfaces with the semantic system', and this fact explains its invariance (or at least part of it). LF is also the level of representation at which supposedly 'universal' syntactic properties of language are manifested. Minimalism describes LF structures by defining an algorithm which generates, or 'derives' these structures. This algorithm is largely the same for each language. The process of deriving an LF structure involves joining together elementary phrase structures associated with lexical items, and also movement of words and phrases within the structure thus created. For instance, the subject and object noun phrases of a transitive sentence (denoting the agent and patient respectively) originate at positions in the VP (the phrase associated with the verb) but they each move to higher positions above the VP during derivation, in an operation called 'DP-movement'.[3] The subject moves to a position in the 'inflection phrase' (IP), and the object to a similar position in the 'object agreement phrase' (AgrP) as illustrated in Figure 1.[4] At some point during these movement operations, the surface form of the sentence (PF) is 'read off' the LF structure. One of the learnable parameters in the language faculty relates to whether subject and object are pronounced 'before' they move or after. This accounts for the different positions that subject and object can take in different languages.

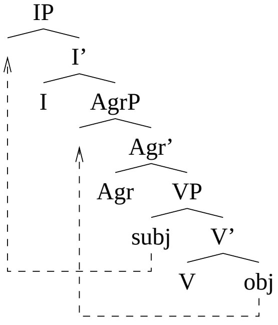

Figure 1: 'DP-movement' during the derivation of a transitive sentence. Subject and object noun phrases (DPs) originate within the verb phrase, and move to higher positions associated with subject (IP) and object (AgrP).

Note that the notion of 'movement' of constituents during derivation of LF structures is emphatically not intended as part of a model of sentence processing: in the Chomskyan account, the algorithm which derives sentences is just part of a declarative specification of the set of syntactically well-formed sentences in a given language. On the other hand, our neural network model is squarely a model of sentence processing. What we want to point out is that there are natural correlates of some of the operations in the LF derivation algorithm in the structure of the sensorimotor account of reaching-to-grasp sketched above. As regards DP-movement, our scheme provides two possible positions for both subject and object in a transitive sentence, and envisages a mechanism which learns which position to use in each case from the exposure language.

2.4.2. Syntactic agreement

DP-movement expresses the idea that there are two natural positions in a sentence for noun phrases realising the arguments of a transitive verb. However, there are other elements of the sentence which can also convey information about these noun phrases. In many languages, information about subject and/or

[3] Noun phrases are often re-analysed as 'determiner phrases' (DPs) in theoretical syntax.

[4] Our label 'IP' is fact a conflation of a subject agreement projection and a 'tense' projection (TP), which we will not distinguish between in this paper.

object noun phrases can also appear in the verb, in the form of morphological inflections. A verb's 'agreement inflections' basically repeat information signalled by its argument NPs. For instance, in English, the inflection *-s* conveys that the verb's subject is singular, and is referred to in the third person. Agreement inflections only convey limited information: most commonly, grammatical person, number and gender. Most of the detailed information about the participants in a reported episode is conveyed by open-class lexical items inside noun phrases (common nouns, adjectives, proper nouns). But a model of syntax still has to explain the existence of agreement inflections. The thing to explain is how the verb can mirror information provided by the subject and object—especially when these can appear quite far away from the verb in a sentence's structure. Whatever syntactic paradigm we adopt, we must explain how the verb has a 'domain of locality' large enough to encompass the NPs which realise its arguments, allowing it to display agreement with these NPs. In some formalisms, the pervasive syntactic influence of the main verb of a sentence is expressed using a mechanism of 'agreement features' which can be transmitted between the nodes of a syntactic tree (see e.g. Pollard & Sag, 1994). In others, it is expressed by having verbs introduce arbitrarily large pieces of syntactic structure (see e.g. Joshi & Schabes, 1997). In Chomskyan paradigms, it is expressed using the notion of verb movement. The basic idea here is that during derivation of an LF structure, the verb moves into syntactic positions where it is close to the subject and object, where it is natural to assume that semantic information can be shared. In Minimalism, for instance, a verb originally appears within its own VP, complete with agreement inflections. But it must move to higher positions in order to 'check' semantic features associated with these inflections. It first moves to AgrP, where it has a local syntactic relationship with the position the object moves to. Then it moves to IP, where it has a similar structural relationship with the position the subject moves to. This type of movement is quite distinct from DP-movement: rather than jumping directly to a higher position, the verb moves iteratively from one head position to the next one up, as shown in Figure 2. The fact that the verb moves into local configurations with subject and object positions provides an explanation of how its inflections can echo in-

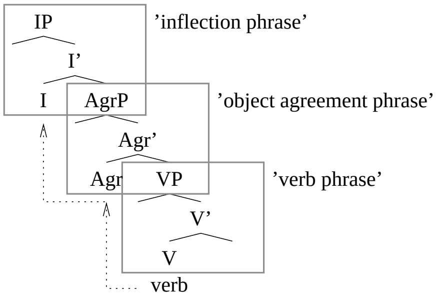

Figure 2: 'Verb movement' during the derivation of a transitive sentence

formation conveyed by the subject or object noun phrase. In addition, it also provides a range of possible locations for the verb, which neatly contribute to an account of the differences between languages. As with subject and object NPs, the verb can be read out at a 'low' position (before movement) or a 'high' position (after movement); this constitutes another parameter for children to set when they are learning their native language.

Our notion of rehearsed sensorimotor sequences allows an interesting account of agreement inflections which in some way chimes with the Minimalist notion of verb movement. Again, since we are implementing a model of sentence processing, we are not modelling any explicit 'movement' of the verb. Our aim is to give a model of sentence processing which has some recognisable analogue of the notion of verb movement.

To begin with, note that in the sensorimotor sequence shown in Table 1 there are several opportunities to pronounce the verb. Recall that experiencing a reach-to-grasp episode involves executing a sequence of three sensorimotor operations, and that we assume a speaker must replay this sequence in order to express it verbally. Recall also that the working memory mechanisms which store sequences of sensorimotor operations feature representations of these operations which remain *tonically active* when a sequence is replayed. The verb in our example transitive sentence reports a sensorimotor operation. Referring to Table 1, note that there are two possible ways we might read out a verb from a replayed sensorimotor sequence. We could read it out from the transient *grasp* motor signal which occurs uniquely in context $C3$. Or we could read it out from the planning representation which enables replay of the sequence, which is active at each iteration of the replay process. On this alternative, an action verb is read out from a representation of a *planned* action, rather than from a representation of an action itself. If we assume this alternative, we have a natural account of why the verb has a syntactic domain which encompasses a whole sentence.

Now consider the agreement inflections on a verb. These report a small amount of the information about the agent (or patient) which is conveyed in detail by the subject (or object) noun phrases. Again it is useful to note that there are several sources of information about the agent and the patient in a replayed sensorimotor sequence. Assume a language where the verb inflection agrees with the subject—so for our cup-grasping example, provides information about the agent. Referring once more to Table 1, we can read out information about the agent from the (replayed) operation of attending to the agent (*attend_agent*), the reafferent sensory representation which results from this operation (*attending_agent*), or the planning representation which supports replay of this attentional operation ($plan_{attend\_agent}$). We have already proposed that full noun phrases are read out from transitory reafferent signals. But we can also envisage that attentional operations have linguistic side-effects. Specifically, we propose that *agreement inflections are read out from planned attentional operations.*

This proposal achieves several things. Firstly, it explains why agreement inflections have the same range of potential syntactic positions as verb stems.

Planned attentional operations are tonically active throughout replay of a sensorimotor sequence, just like planned motor operations, and so can be read out at any position, including at positions which are distant from the positions at which noun phrases are read out. Secondly, it explains why agreement inflections appear *with* verb stems. Sequence plans have separable components, but they are still neural assemblies in their own right; it is natural to assume that they are made available as wholes to a linguistic interface. Finally, it goes some way towards explaining the kind of semantic information which is conveyed in agreement inflections. Attending to oneself is very different from attending to another person, and arguably attending to one's interlocutor is different from attending to a third party. Similarly, attending to a single object is significantly different from attending to a group of objects (see Walles et al., under review for some references and a computational model). Arguably, therefore, agreement inflections pick up on the 'attentional' components of sensorimotor sequences.

It is useful to consider the model just outlined in an evolutionary perspective. Assume that our prelinguistic ancestors experienced the world through deictic routines similar to ours, and used similar mechanisms to store these routines in working memory. (The models of prepared sequences we introduced in Section 2.2.2 come from macaque.) The adaptations needed for language are circuits allow sensorimotor signals to generate overt linguistic side-effects. Our proposal is that evolution happened to find a way of expressing motor programmes using a circuit connecting to the action preparation system in prefrontal cortex, while the method it found for expressing objects involved a circuit connecting to areas of the brain evoking transient object representations, with only coarse-grained inputs coming from planning areas. This proposal in turn links quite well to current research in neurolinguistics and brain imaging. For instance, in a well-known study by Pulvermüller, F et al. (1999) presentation of concrete nouns and verbs was found to preferentially activate different areas: concrete nouns with strong visual associations activated visual cortices, and action verbs activated motor, premotor cortices but also prefrontal cortices. Generating verbs also appears to involve a large region of left anterior prefrontal cortex (see e.g. Perani et al., 1999; Tranel et al., 2001). Interestingly, one area of prefrontal cortex (left Brodmann's area 9) appears selectively involved in producing verb inflections (see Shapiro & Caramazza, 2001).

2.4.3. Hierarchical constituent structure

Another interesting way our model of sensorimotor sequences connects with syntactic theory concerns how syntactic units are defined. Most syntactic paradigms include the proposal that lexical items 'project' their own local syntactic structure—i.e. specify the kinds of syntactic environment they can appear in. On these models, a verb phrase is the structure projected by a verb, a noun phrase is the structure projected by a noun, and so on. Many accounts include a suggestion that the shape of this projected structure is the same for different word types: i.e. that verb phrases, noun phrases etc. are all instances of a more basic structural template, often termed the 'X phrase', or 'XP'. On this view, an XP is the fundamental building block of syntactic structures. Minimalism makes heavy use of this idea; XPs are identified by the grey boxes in Figures 1 and 2. In the Minimalist account, inflections can contribute projections to a sentence as well as open-class lexical items.[5] Even 'syntactic objects with no phonological content' can contribute projections to a sentence—a point which is very controversial for linguists from other backgrounds. In a Minimalist model, the backbone of a clause (at LF) is a right-branching structure of XPs, the first two of which are introduced by inflectional elements, and the third of which is the VP. This analysis appears profligate, postulating a fairly extended hierarchical structure for a relatively simple sentence. In one sense this is true; however, in another sense it is very economical, in that it only posits a single syntactic schema, the XP schema, which is recursively applied.

In our model of sensorimotor sequences there is a very natural interpretation of the X-bar schema. Replayed sensorimotor sequences have several iterations, each of which has the same basic structure: there is an initial context, a (replayed) sensorimotor operation, which triggers a reafferent sensory side-effect, and brings about a new context. There is also a tonically active planning representation which supports the replayed operation. The structure of the X-bar schema is shown in Figure 3(a). In Figure 3(b) we show that there is a natural sensorimotor interpretation of the X-bar schema as a description of a single iteration within a replayed sensorimotor sequence. (Note also that as a corollary of this interpretation, a right-branching chain of X-bar schemas describes a *sequence* of iterations, because the next context of one schema is the initial context of the next one: just the result we want.)

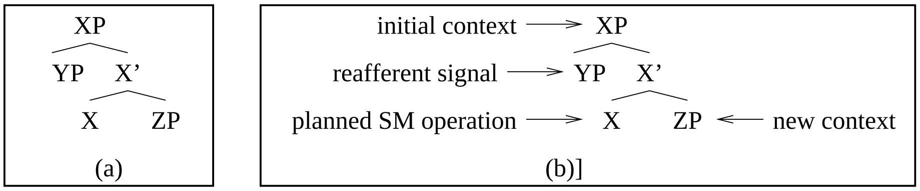

Figure 3: (a) The X-bar schema. (b) A sensorimotor interpretation of the X-bar schema

To summarise: there are several ways in which our notion of sensorimotor sequences makes interesting contact with syntactic theory. It allows an interesting account of the alternative syntactic positions of noun phrases and verbs, and an interesting account of the syntactic properties of verbs which allow them to carry agreement inflections. It also makes an interesting proposal about the basic recursive building blocks of syntactic structures. We have expressed all these ideas with reference to Minimalism, because they correspond particularly well to devices within this paradigm. However, we again emphasise that what we are proposing is a model of sentence processing which reinterprets these devices.

[5] This is another simplification, which is in fact a description of Government-Binding theory (see e.g. Chomsky, 1981) rather than Minimalism. But there is a similar idea in Minimalism.

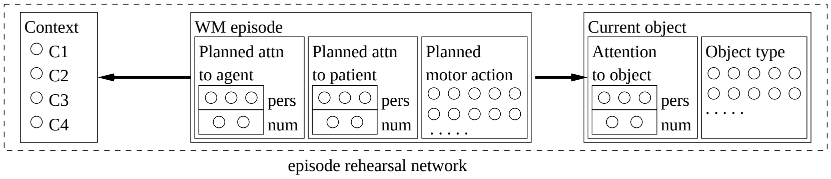

Figure 4: The mature episode rehearsal system

We turn now to the model itself. We will introduce it in several steps. Section 3 describes the core of the model, which captures the syntactic phenomena just discussed. Section 4 describes some additional components of the model which allow it to learn to generate surface patterns in language. Section 5 describes the complete model, which can learn a mixture of syntactic and surface-based linguistic patterns.

3. A core network for learning abstract syntactic rules

The complete model of language production consists of several functional modules that work together: an **episode rehearsal network**, which replays a working memory episode representation to generate a sequence of sensorimotor signals; a **word production network**, which maps individual sensorimotor signals onto word forms; a **control network**, which determines the points during episode rehearsal when these word forms should be pronounced; and a **word sequencing network** which learns surface regularities in word sequences. In this section, we describe how the first three of these modules work together. For technical details of all networks see Appendix A.

3.1. The episode rehearsal network

First we present the network which replays sensorimotor sequences, which embodies the key novel idea of this article. This system is responsible for generating sequences of the kind shown in Table 1 and provides the (semantic) input for other modules of the language production system.

The developmentally mature version of the episode rehearsal network is shown in Figure 4. The **WM episode** area models a working memory episode representation that takes the form of a prepared sensorimotor sequence tonically active in PFC. Besides a planned motor action, it comprises planned actions of attention to an agent and a patient. The diagram only shows information which can be linguistically expressed.

Planned attentional actions interface rather weakly with the linguistic system: they convey basic information about whether the attended object is the agent himself or his interlocutor or something else, and about whether the attended object is a single entity or a group. (This information ends up being expressed in grammatical person and number inflections on verbs.[6]) For each component of a planned attentional action, we use 1-hot localist coding, i.e. there is exactly one active unit at a time, out of three units specifying (first, second and third) person, and one out of two units for (singular or plural) number of the agent. The same holds for the patient person and number units. The planned motor action interfaces more strongly with language: there is one unit for each open-class motor action.

The **current object** area holds a transient representation of the currently attended object. During the course of episode rehearsal, this area alternately holds representations of the agent and the patient. This area conveys person and number information in the same format as the WM episode (i.e. coarse information about an action of attention), but unlike the WM episode it also conveys fine-grained information about the type of the attended object. Again we use a 1-hot coding scheme to represent type information; i.e. there is a dedicated unit for each possible object concept (regardless of the role it appears in). Together these sources of information will eventually enable the generation of inflected open-class nouns, and of pronouns.

The **context** area holds a representation of the current stage during episode rehearsal. This representation helps to drive the episode rehearsal process. In our simulation there are four possible contexts (see Table 1), each represented by a single localist unit. The thick arrows in the diagram reflect the fact that the sequence of transient representations in the current object and context areas are generated by a WM episode representation.

The episode rehearsal system provides input to the word production and control networks, which we will describe next.

3.2. The word production network

The word production network is shown in Figure 5. It serves as the system's lexicon, in that it learns to generate a (possibly inflected) word in response to an input signal from the episode rehearsal system. The inputs to the network are the WM episode and current object areas of the episode rehearsal network. The output layer holds a set of units that represent all possible words—or more precisely, all possible word stems and all possible inflections, including the null inflection. Word stems and inflections are represented in a localist fashion: i.e. there is one unit for each stem and each inflection. Total activation of all units in the word stem area can be scaled to sum to 1 and treated as a probability distribution; likewise for the inflection area. We envisage that individual word stems and inflections represent premotor articulatory plans, rather than actual utterances.

The input and output layers are fully connected. These connections are gated by inhibitory links from a cyclic pattern generator (depicted as 'Phase' in Fig. 5) so that at any time input comes either wholly from the WM episode

[6] We discuss gender agreement in Section 7.3.

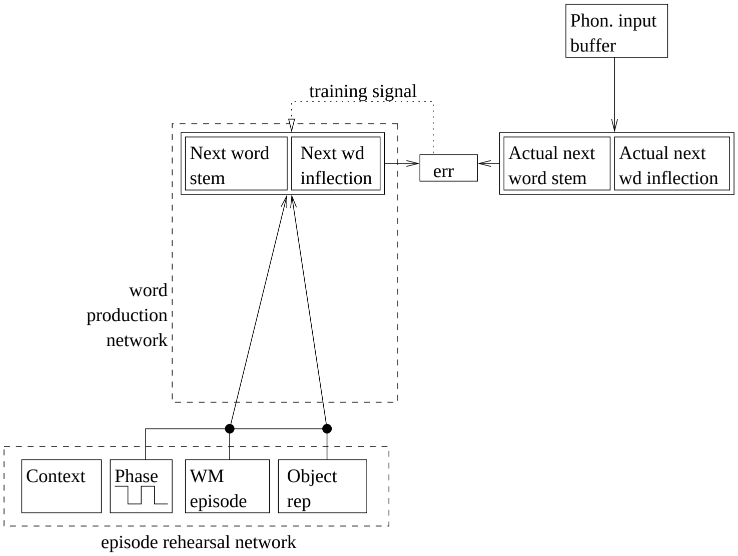

Figure 5: The word production network receives inputs from the episode rehearsal system and training signals from the phonological input buffer. (Note that the episode rehearsal system is augmented with a pattern generator that delivers inputs from the episode rehearsal system in two alternating phases.)

or wholly from the current object. During episode rehearsal, the pattern generator cycles through two phases in each context, providing first an opportunity to read out the tonically active WM episode, and then an opportunity to read out the current object representation, as shown in Table 2. Pattern generators are commonly postulated in models of the prosodic aspects of language production; see for instance Hartley & Houghton (1996) neural network model of syllabic structure. We propose that a pattern generator is also involved in syntactic processing, to produce the regular alternation between heads and specifiers characteristic of X-bar structure.

The word production network is trained on the utterances of mature speakers, paired with episode representations. We are simulating an infant who experiences episodes in the world and who also hears mature speakers talking. We assume the infant is well enough attuned to the pragmatics of communicative actions to pair the utterances of mature speakers somewhat reliably with semantic representations of the episodes they report, using devices such as joint attention and intention recognition (see e.g. Tomasello, 2003), though of course there is a great deal of noise in the mapping between semantic signals and words, especially to begin with.

The mature utterances the system hears are stored in a **phonological input buffer**, from where they can be replayed word-by-word. The episodes the system experiences are stored in the episode rehearsal network. Note that

Table 2: The sequence of inputs to the word production network after modulation by the pattern generator. Object representations and WM episodes alternate.

Context	Phase	WM episode rep	Object rep
C_1	a		MAN
	b	$plan_{attend\_agent/attend\_cup/grasp}$	
C_2	a		CUP
	b	$plan_{attend\_agent/attend\_cup/grasp}$	
C_3	a		MAN
	b	$plan_{attend\_agent/attend\_cup/grasp}$	
C_4	a		CUP

episodes and utterances are stored in quite separate media in working memory. We assume, following Baddeley (2000), a distinction between a phonological input buffer, holding a recently presented sequence of words, and an 'episodic buffer', holding semantic material. Words replayed from the phonological input buffer function as training signals for the word production network. They are represented in exactly the same way as words generated by the word production network.[7] An error term is calculated based on the difference between the 'next word' predicted by the production network and the 'actual next word' of the training utterance, and this term is used to train the production network.

Note that the word production network is trained on a replayed sequence of words, rather than words arriving in real time. Initially, the effect of this 'offline' form of training is to allow several training words to be presented for each sensorimotor signal, which helps to combat the noisiness of the training data. (There is a well-attested relationship between phonological working memory capacity and early vocabulary size; see e.g. Gathercole & Baddeley, 1990.) However, we argue later in the paper that offline training also has a role in syntactic development.

3.3. The control network

Assume that some learning has taken place in the word production network, and that it can reliably map some sensorimotor signals onto words. Now consider what happens when an episode is rehearsed. The word production network receives a sequence of sensorimotor signals—two in each context—and for each signal it will generate a word form. As is clear from Table 2, each sensorimotor signal occurs more than once in this sequence: the planning representations occur once per context, and the transient representations of agent and patient each occur exactly twice. Of course, sentences do not contain wholesale repetition of

[7] We assume that phonological word representations in the input buffer are stored as articulatory plans (see e.g. Browman & Goldstein, 1995) and are therefore directly comparable to words generated by the word-production network.

Table 3: A control policy which produces VSO (verb subject object) word order ('—' and '↓' denote 'withold' and 'pronounce' respectively).

Context/phase	C1a	C1b	C2a	C2b	C3a	C3b	C4a
SM sequence	MAN	GRAB-PLAN	CUP	GRAB-PLAN	MAN	GRAB-PLAN	CUP
control policy	—	↓	—	—	↓	—	↓
output words		*grabs*			*man*		*cup*

words—at least, not to this degree. We therefore envisage a device which learns when to pronounce the word forms evoked by the word production network, and when to withold them. Our suggestion is that different languages have different conventions about which versions of the agent, patient and action signals to pronounce, and that these conventions determine the basic word order of the language. In our model, the device which learns a policy about when to pronounce these repeated sensorimotor signals is called the **control network**. For instance, in a VSO (verb, subject, object) language, the control network must learn to pronounce the action signal at the first opportunity, and the agent and patient signals each at the second opportunity, as shown in Table 3. In Minimalist terms, the episode rehearsal network implements the logical form (LF) of a sentence, and the control network learns to map this logical form onto a surface sequence of words, or phonetic form (PF).

The control network, with its connections to the networks described earlier, is shown in Figure 6. It takes its input from the context and phase areas of the episode rehearsal network. These areas are fully connected to a hidden layer, which is in turn fully connected to one output unit that serves as a gating signal between the output layer of the word production network and the actual phonological output.[8]

In neural terms, we think of the word forms generated by the word production network as premotor articulatory plans, rather than overt motor outputs. This allows for a separate system to decide whether to overtly pronounce any given word form. The idea that one can prepare an action without executing it is well established in models of the motor system; see for instance Fadiga et al. (2002) for evidence specific to articulatory actions. The control network's role is to decide which premotor word forms evoked during a rehearsed episode should be overtly pronounced: in other words, its role is to selectively enable and disable a connection from premotor to motor articulatory cortex. We assume the control network is part of Broca's area, because Broca's area is known to have a general nonlinguistic role in suppressing habitual responses (see Novick et al., 2005 for a review of evidence to this effect).

[8] As indicated in Figure 6, we assume the phonological output system has internal structure of its own. Items to be pronounced sit in a **phonological output buffer** where phonological planning effects at the level of prosody can be modelled. For a review of evidence for a separate phonological output buffer, see e.g. Shallice et al. (2000). In fact our current implementation does not model such effects.

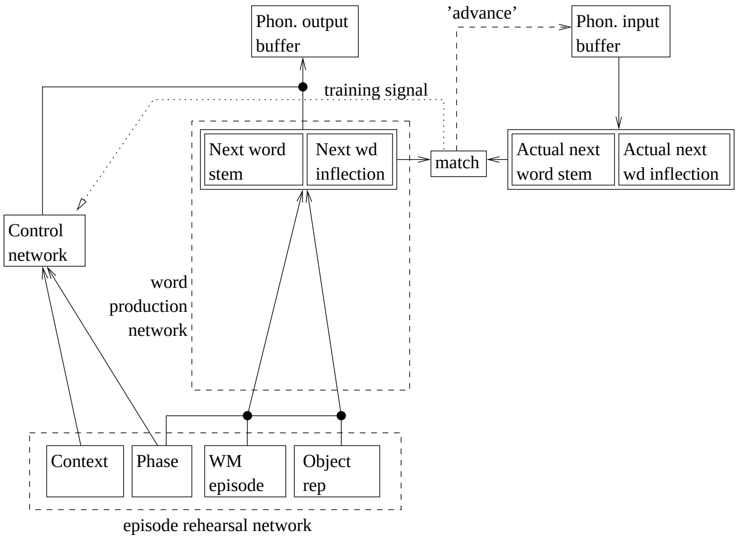

Figure 6: The control network gates the output of the word production network. It is trained on a Boolean match between the word generation network's predicted next word and the actual next word in the phonological input buffer.

Like the word production network, the control network is trained on utterances paired with episode representations. As already noted, the episode is stored as a replayable sensorimotor sequence, and the utterance is stored as a separately replayable sequence of words. The words replayed from the input buffer again function as training signals for the control network, but in a slightly different way. For the control network we use a 'match' operation, which compares the word predicted by the word production network with the 'next word' replayed from the phonological input buffer and returns a Boolean value: either the word matches or it does not. This Boolean value functions as a training signal for the control network, but it also has a procedural role in synchronising the training utterance being replayed with the episode being rehearsed. This is important, because there are many more iterations in episode rehearsal than there are words in the training utterance: we cannot advance to a new word in the training utterance at each iteration.

The 'match' circuit works as follows. If the output of the word production network matches the 'next word' replayed from the phonological input buffer, the control network will be trained to generate a 'pronounce' signal, which allows this word into the phonological output buffer. At the same time, it generates a signal to advance to the next item in the training utterance. If there is no match, on the other hand, the control network is trained to generate a 'withold' signal, which prevents the output of the word production network from being pronounced—and there is no signal to advance to a new word in the training utterance.

Table 4: A typical training item in a VSO exposure language, with the 'match' signal generated in each context/phase. The 'actual next word' field steps through the words of the target utterance one at a time, advancing to a new word when it matches the predicted next word.

Context/phase	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	grabs man cup						
SM signal	MAN	GRAB-PLAN	CUP	GRAB-PLAN	MAN	GRAB-PLAN	CUP
predicted next wd	*man*	*grabs*	*cup*	*grabs*	*man*	*grabs*	*cup*
actual next wd	*grabs*	*grabs*	*man*	*man*	*man*	*cup*	*cup*
'match' signal	no	yes	no	no	yes	no	yes
training signal for ctrl network	—	↓	—	—	↓	—	↓

To illustrate the training mechanism, assume the system is exposed to training items from a VSO language. A training item representing the episode 'a man grabs a cup' is shown in Table 4.[9] The training item consists of a sensorimotor sequence representing this episode, paired with an utterance which reports the episode in a VSO language. During training, the sensorimotor sequence is rehearsed one step at a time. At each step, the figure shows the sensorimotor signal providing input to the word production network, along with the word this network predicts from this signal. It also shows the 'actual next word' in the training utterance, as replayed from the phonological input buffer. At each stage, the 'match' signal reflects whether these two words are the same. If they are not, the control network is trained to give the 'withold' signal, and the word is retained at the next step. If they are, the control network is trained to givee the 'pronounce' signal, and we advance to the next word in the training utterance. With enough training examples of this kind, the control network will learn a policy of generating 'pronounce' at context/phases C1b, C3a and C4a, and 'withold' at all other contexts/phases.

There are two interesting things to note about the control network's training regime. Firstly, note that the control network is *content blind.* It does not receive any information about individual words or word meanings—only about contexts and phases. It learns that in some contexts/phases the output of the word production network should be pronounced, while in others it should be witheld, but it does not know anything about the content of the words it is controlling. In other words, it learns 'structural', content-independent word-ordering rules. We will demonstrate some of these rules in Section 6.2.2. Secondly, note that the control network learns its rules 'offline': it learns to map a *replayed* sensorimotor sequence onto a *replayed* sequence of words, by selectively advancing the sequence of words so it is synchronised with the sensorimotor sequence. The system has precise control over the way the training utterance is presented, ad-

[9] We distinguish words and concepts by font: words are in *italics* and concepts are again given in SMALL CAPS.

vancing to a new word in some context/phases, but not in others. This is only possible because training happens offline.

4. Extensions allowing the learning of surface patterns in language

The model described so far can learn a lexicon and a set of abstract word ordering conventions for a given a target language. In this section, we will describe two additional networks which allow the model to learn surface structures in language. One is a **word sequencing network**—a familiar SRN-style network. The other is a more novel **entropy network**, which controls how the sequencing network operates. These networks are shown in Figure 7, which also shows some of the components of the earlier network which they interact with.

4.1. The word sequencing network

The word sequencing network is a variant of a Simple Recurrent Network (Elman, 1990). In a way it mimics the word production network: as shown in Figure 7, its input layer consists of the WM episode and current object areas of the episode rehearsal system (gated by the phase generator), and its output layer has an identical structure to the output layer of the word production network. Both networks are trained from the 'actual' next word replayed from the phonological input buffer. However, the word sequencing network has one hidden layer with recurrent connections, which enables it to take into account the history of previous inputs. In each step, activities of the hidden layer from the previous step are copied to a context layer, which provides an additional input to the hidden layer at the next time step. We will refer to the context layer as the **surface context**, to distinguish it from the context representation used in the episode rehearsal network. Using this surface context representation, the network can learn to produce different words for a given semantic input depending on the history of preceding inputs (while the word production network would produce the same output word regardless of the context). Moreover, the word sequencing network can produce a *sequence* of different output words for one (unchanging) semantic input, because of its recurrently defined surface context layer.

The output of the aggregated network depicted in Figure 7 (the top 'next word stem/inflection' box) is a simple average of the activities in the output layers of the word production and word sequencing networks. Because each of the output layers represents two probability distributions (see Section 3.2), the aggregated result also represents probability distributions of a predicted word stem and an inflection.[10]

We will refer to the word production and word sequencing networks together as the **word production/sequencing network** or **WPSN**. In a mature system, we argue that the production/sequencing network interacts with the con-

[10] Computing simple linear combinations of probabilistic population codes is biologically plausible (Ma et al., 2006).

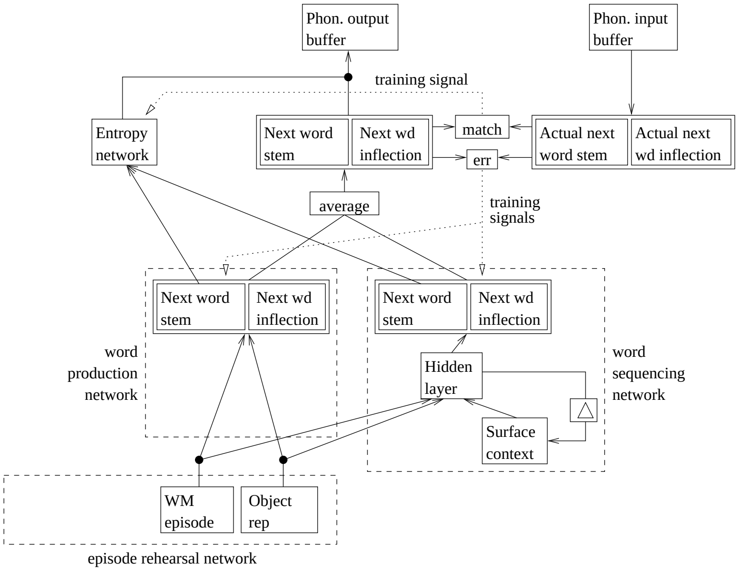

Figure 7: The word sequencing network and its interaction with the word production network. The aggregated output of the two networks is gated by a confidence signal provided by the entropy network.

trol network to support the generation of 'idiomatic' language. However, we also argue that it has an important developmental role, in generating early presyntactic multi-word utterances before the control network comes online. These roles of the production/sequencing network will be discussed in Section 5.2.

Note that the word-sequencing network is not a standard SRN. A standard Elman network takes a 'current word' as input (as well as the current surface context) and predicts the next word. Our network takes a *word meaning* as input (as well as the current surface context) and predicts a word as output. In fact, this meaning is the meaning of the next word—so the network basically learns to map meanings onto word forms as well as learning about sequential structures of words. In fact, the network learns about sequential structures of *word meanings* rather than of actual words. But because it receives structured *sequences* of word meanings, and because word meanings map closely onto words, it is still able to use this learning to predict the next word on the basis of the recently produced words, like any Elman network.

4.2. The entropy network

The word-production/sequencing network operates in conjunction with another network: the **entropy network** at the top left of Figure 7. This network generates the same kind of output as the control network: a gating signal determining whether the word form chosen by the combined word production and word sequencing networks is pronounced or witheld. But its decision has an altogether separate motivation, to do with the *confidence* of the production/sequencing network in its choice of word. The basic function of the entropy network is to make sure a word is not pronounced unless the production/sequencing network is reasonably confident it is the right one. Early during development, the network has the function of preventing the generation of 'nonsense', i.e. random outputs. Later in development, the network has a role in identifying surface linguistic patterns and treating these in a special way, as we will describe in Section 5.

The entropy network receives inputs from both the word production and word sequencing networks. Each of these networks computes a measure of confidence in its own prediction about the next word stem. We use the statistical measure of **entropy**. (As already noted, the output of each network can be understood as a probability distribution over possible word stems. The entropy of a probability distribution is a measure of how evenly probabilities are distributed; entropy is low when just one word is strongly predicted, and high when there are many competing alternatives.) The entropy network's function is basically to decide *how confident* the word production/sequencing networks need to be in order to warrant their predicted word being produced. The entropy network learns a simple threshold function, which takes the entropies of the word production and sequencing networks and returns a binary decision: 'pronounce' or 'withold'.[11]

The entropy network is a feed-forward network (multi-layer perceptron) with one hidden layer. It has two input units, holding the entropy values computed from the word production and word sequencing networks, and one output unit, encoding a 'pronounce' or 'withold' signal. The network learns its threshold function from the same signal as the control network: a binary 'match' between the predicted next word and the actual next word. For details, see Appendix A.

4.3. Interactions between the sequencing and entropy networks: a model of idioms

As discussed in Section 4.1, the main function of the word sequencing network is to learn 'idiomatic' constructions in language: that is, constructions expressing a single semantic signal as an extended pattern of several words. Note that when the sequencing network is producing an idiomatic pattern of words, it can often make confident predictions about several words in a row from just one semantic input. For instance, consider the continuous idiom *Winnie the Pooh*, a sequence of words which collectively express the object concept WINNIETHEPOOH, or WTP for short. Imagine the network has encountered this construction many times during training. It will learn that when it first sees the concept WTP it can confidently predict the word *Winnie*—but after having produced this word and updated its own surface context representation, it can

[11] The entropy network takes two separate entropies rather than a single 'aggregate' entropy in order to allow a special treatment of newly-learned words; for details see Section 6.2.8.

confidently predict the next word (*the*), without any additional semantic inputs. And after another update of its surface context, it can confidently predict the last word of the idiom, *Pooh*.[12] At this point, of course, it can no longer be confident about the next word without receiving an additional semantic input. Like any Elman network predicting the next word on the basis of recently produced words, it will know what *class* of word to expect, but outside idiomatic constructions it cannot select a particular word from this class without knowing its semantics. In our system, idioms are modelled using the concept of entropy—or more specifically, of an entropy threshold on pronunciation. As a first approximation, we define an idiom as a sequence of words which can each be predicted (with enough confidence to be pronounced) by the production/sequencing networks, from a single semantic input, and the recurrent representations produced in the sequencing network's surface context layer. In the next section we will refer to this definition of idioms in our account of how the word sequencing network interacts with the control network.

5. A combined model of word-learning and syntactic development

So far we have conceptually introduced the submodules of our model and outlined their developmental role. In this section we describe the complete model in the form we implemented it. First we describe how the model works after all its components have been fully trained, then we focus on training.

5.1. Language generation in the combined model

The complete model is shown in Figure 8. It combines the word production and sequencing networks described in Section 4 with the episode rehearsal and control networks described in Section 3.

In the complete model, generating a sentence involves replaying an episode in the episode-rehearsal system, and at various points during replay, pronouncing one or more words (i.e. dispatching words to the phonological output buffer). A key issue in the combined model is the synchronisation between the episode rehearsal and word sequencing networks. Both these networks are iterative in nature: the episode rehearsal network iterates through a sequence of sensorimotor signals, and the word sequencing network iterates over a sequence of words. Sometimes these iterations should be synchronised, so that each new sensorimotor signal results in a pronounced word. But sometimes they are out of synch. There are occasions when a sensorimotor signal should occur without any words being pronounced: these are the contexts in which the control network has learned to 'withold' a word, to conform to the word-ordering constraints of the exposure language. There are also occasions when multiple words

[12] Note that the sequencing network is not predicting the words which follow *Winnie* from the *word* '*Winnie*', but from the word representation 'WTP'. Our sequencing network can basically learn a lexicon of idiomatic expressions, as well as the meanings of individual words. An ordinary Elman network making predictions about the next word would have difficulty deciding between *Winnie the Pooh* and *Winnie Mandela*.

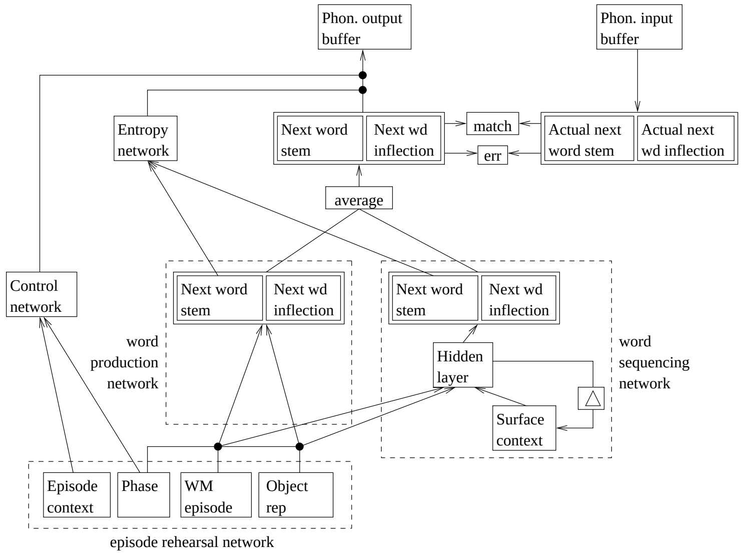

Figure 8: The complete model of language production. Besides gating overt pronunciation, the control network and the entropy network coordinate mode switching between episode rehearsal and surface word sequencing.

should be produced for a single sensorimotor signal: this is the case for idiomatic constructions, as just discussed in Section 4.2. In our combined network, the control and entropy networks jointly manage the synchronisation of the episode rehearsal and word sequencing networks.

The combined network alternates between two modes of iteration. In one mode, the episode rehearsal system iterates through a sequence of sensorimotor signals until it reaches an episode context and a phase at which the control network allows a word to be overtly pronounced. Then it switches to the other mode, in which the word production/sequencing networks generate predictions about the next word. If they can confidently predict the next word (a decision based on the output of the entropy network), the word is pronounced, the sequencing network updates its surface context layer and the networks attempt to predict another word (from the same sensorimotor signal). Iteration continues in this mode, with a static sensorimotor signal, until the word production/sequencing networks can no longer confidently predict the next word. Then the model switches back into the episode rehearsal mode. The algorithm is described in detail in Appendix A.

To illustrate, say that the system is given as input an episode in which Winnie the Pooh grabs a cup. Assume this time that the system has been trained to produce an SVO language (like English). The generation process involves replaying the sensorimotor sequence encoding this episode, and generating a sequence of words. The representations computed during the first seven iter-

Table 5: Processing involved in generating *Winnie the Pooh grabs (the) cup*. (We assume the system trained on an SVO language, including the idiom 'Winnie the Pooh'.) Only the first three context/phases are shown.

Context/phase	C1a				C1b		C2a
Surface context	c1	c2	c3	c4		c5	
SM signal	WTP	WTP	WTP	WTP	GRAB-PLAN	GRAB-PLAN	CUP
predicted next wd	*winnie*	*the*	*pooh*	*?*	*grabs*	*?*	*cup*
confident?	yes	yes	yes	no	yes	no	yes
control network	↓	↓	↓		↓		↓
output word	*winnie*	*the*	*pooh*		*grabs*		*cup*

ations of this process are shown in Table 5. Notice that iterations can now be over surface contexts (denoted c1, c2 etc) as well as over episode-rehearsal context/phases (C1a, C1b etc). The first sensorimotor signal is WTP (short for WinnieThePooh). From this signal, and the 'start-of-sentence' surface context (c1), the word production and sequencing networks can confidently predict the word *winnie*. Since the control network has learned to pronounce words in context/phase C1a, this word is produced. Before we iterate to a new context/phase, we now update to a new surface context (c2) to reflect the newly produced word, and the word production/sequencing networks predict another word from the same sensorimotor signal WTP. In the updated context, they now predict *the*, again with high confidence, so this word is also produced, and we update the surface context to c3. In this context, the word production/sequencing networks predict *pooh* with high confidence, and we update to surface context c4. In c4, the word production/sequencing networks can no longer confidently predict the next word. At this point, control reverts to the episode-rehearsal network, which updates to context/phase C1b, in which the sensorimotor signal GRAB-PLAN is activated. The control network for an SVO language allows words to be pronounced in this phase, and the production/sequencing networks confidently predict the word *grabs*. (This word is consistent with the semantic signal, and is also commonly attested following the phrase *Winnie the Pooh*.) Now we update to a new surface context and attempt to predict another word, but we cannot: *grabs* is not part of an idiom. So we advance to the context/phase C1b, activating the sensorimotor signal CUP. The SVO control network has learned to pronounce words in this context/phase, and the production/sequencing networks confidently predict *cup*, so the last word in the sentence is pronounced. As this example demonstrates, during sentence generation the network alternates between updates in the episode rehearsal system and updates to the surface context. Portions of the sentence generated when the surface context is updating by itself reflect idiomatic surface structures in the exposure language. Portions generated after an update in the episode rehearsal system reflect content-independent word-order rules.

5.2. Development and bootstrapping of the model

The combined model consists of several networks that need to be trained. Training happens in parallel, in a coordinated way that ensures learning can bootstrap. There are two basic systems which must bootstrap one another: one is the control network and one is the word-production/sequencing network (WPSN).

On one hand, there must be some learning in the WPSN before the control network can start to learn. Learning in the control network is governed by whether the next word predicted by the WPSN 'matches' the actual next word in the training utterance. Until the WPSN is reliably mapping some sensorimotor signals to words, the 'match' signal provides no information about when to pronounce and when to withold words.

On the other hand, it also makes sense for learning in the WPSN to be dependent on learning in the control network. The word production and sequencing networks are trained to reproduce the 'actual next word' in the phonological input buffer. But this training should only happen in a context/phase where the control network is going to *pronounce* a word. Training the WPSN in other contexts simply adds noise to the training data, mapping semantic signals onto the wrong words in the training utterance. We therefore specify that the word production and word sequencing networks are only trained to reproduce the next word in the training utterance in contexts/phases for which the control network generates the 'pronounce' signal. Similar reasoning applies to the entropy network. The 'match' signal used to train the entropy network is only meaningful in contexts/phases where the control network thinks a word should be pronounced. In other contexts, we *expect* a mismatch, regardless of how much learning has taken place in the production/sequencing networks. In summary, the control network is trained in all contexts/phases, while the word production, word sequencing and entropy networks are only trained in phases permitted by the control network.

During training, the model alternates between the same two modes as during generation. In the first mode, episode rehearsal advances (and the control network is trained) until a context/phase is reached in which the control network gives the 'pronounce' signal. Then the network switches into the word sequencing mode. As long as the WPSN predicts the next word with sufficient confidence and it matches the actual word in the phonological input buffer, the WPSN keeps predicting (based on a changing surface context), being trained, and advancing the phonological input buffer. If the prediction does not match or has a low confidence, the actual word stays in the phonological input buffer, the surface context is not copied and the model switches back to the episode rehearsal mode. Details of the training algorithm are given in Appendix A.

This way of training creates a circular dependence: each network's training relies on the others already giving meaningful output. Our ambition is to model the development of language production from scratch. So at least one of the systems involved must have some ability to do some learning on its own.

Our crucial assumption is that learning in the WPSN (and the entropy network) starts earlier than in the control network. To be concrete: we assume that the ability to store and rehearse episode-denoting sensorimotor sequences takes some time to mature. Since learning in the control network relies on controlled replay of working memory episodes, it can only begin when episodes can be properly replayed. On the other hand, learning in the WPSN and entropy networks does not require controlled replay of working memory episodes. Even if utterances and episodes are replayed without any synchronisation, we expect higher-than-chance correlations between concepts and the words which denote them: the kind of correlations which are exploited in 'cross-situational' word learning mechanisms (see e.g. Siskind, 1996; Yu & Ballard, 2007). In fact, we even expect above-chance correlations between concepts and words as they arrive online, in real time. So there is a good deal of scope for learning in the WPSN before a controllable episode-rehearsal ability matures.

To model the onset of a mature episode-rehearsal ability, we divide the system's training into two discrete stages: one with no episode rehearsal mechanism and one with a fully developed one. In the first stage, the regular episode-rehearsal system is replaced by a system which selects signals from the given sensorimotor sequence essentially at random.[13] During this first stage we also model development of the phonological input buffer. This is a separate developmental process: we assume that an infant begins in a 'cross-situational learning mode', in which all the words in the input buffer are active in parallel, but gradually transitions to a mature mode in which words are replayed from the buffer one by one.[14] In this scheme, the initial role of the phonological buffer is simply to increase the amount of data available to a cross-situational learning algorithm, as mentioned in Section 3.2. But when it has matured it has a more precise role in delivering words from the training utterance one by one. This maturation basically models a transition from a stage where the system learns (and generates) single words, to a stage where it learns (and generates) word sequences. Details are again given in Appendix A.

During the first developmental stage, training data for the WPSN are extremely noisy. If the control network incorrectly witholds a word in the context/phase when it should be pronounced, this word will probably remain in the phonological buffer and be associated with all subsequent sensorimotor signals randomly chosen from the episode (including those irrelevant to the word). Nevertheless, through cross-situational learning, correct concept-word mappings will start to be learned and the WPSN's error will decrease.

[13] In practice, we implement this by disabling the control network rather than the episode-rehearsal system. We advance through the stages of episode rehearsal as normal, but we substitute the output of the control network with a random signal (and also disable training in the control network). Since the WPSN is only trained on a word when the control network allows it to be pronounced, the basic effect is to create training items from signals from the sensorimotor sequence selected at random.

[14] There are many models of the phonological buffer in which words are represented in parallel; see e.g. Burgess & Hitch (1999). In these models, inhibitory connections between words result in the most active word temporarily suppressing the others, and then habituating or inhibiting itself to make way for the next most active word. We assume that it takes time for these inhibitory connections to develop.

In our model, the ability to replay utterances in the phonological input buffer matures some time before the ability to replay sensorimotor sequences in the episodic buffer. This creates a sequence of three developmental stages. In the first stage, before the phonological input buffer has matured, the system can generate individual words, but cannot learn word sequences, because it receives no sequential training signals. In the next stage, it can learn something about word sequences, but since it cannot accurately replay episodes, it cannot use the control network, and therefore cannot learn content-independent syntactic rules. In the final stage, once the control network has come online, it can learn both word sequences and abstract syntactic rules. Of course, vocabulary learning takes place during all three stages. Very roughly, we see the first stage as modelling infants between 10 and 18 months, the second stage as modelling infants from 16 to 30 months, and the third stage as modelling infants after 24 months.[15]

While we simulate gradual maturation process for the phonological input buffer, we decided to model maturation of the episode-rehearsal system at a single point during training, so that its effects could be clearly identified. Of course in a more realistic simulation it would mature more gradually. But we nonetheless want to suggest that the emergence of mature syntax involves a qualitatively new mechanism.

6. Simulations

The model we have just described has been implemented and tested on several artificial languages. The main hypothesis we wanted to test by simulation was that the model can learn to produce syntactically and morphologically correct and semantically adequate sentences of the target language for given meanings (episodes). We specially focused on the interplay between abstract syntactic knowledge (namely word ordering conventions) and surface regularities in the form of idiomatic expressions. But in addition, we wanted the developmental profile of learning in our model to correspond to some broadly defined stages of syntactic development in infants. We wanted the system to begin in a 'single word' stage, then move through a stage of producing short proto-syntactic utterances (corresponding e.g. to the 'pivot schemas' of Braine (1976) or the 'item-based constructions' of Tomasello (2003), before finally producing full-fledged sentences.

6.1. Training data

The model was trained on episodic representations paired with sentences from an artificial target language containing a mixture of idioms and syntactically regular sentences. Although the sentences varied in their degree of idiomaticity, they were syntactically homogeneous in that they all were transitive

[15] These age spans overlap to take into account individual differences between children, as well as the continuous character of language development.

(i.e. containing three semantic roles AGENT, PATIENT, ACTION—"who did what to whom"). The reason for this is that we have a detailed sensorimotor model of simple transitive actions (that of Knott, in press). We will introduce other syntactic constructions in due course.

A basic language we used for most of our experiments was an invented language with the SVO word order, English vocabulary, English-like inflections on nouns signalling number, and 'rich' inflections on verbs signalling the person and number of their subjects. The inflections were represented schematically as a suffix on word stems, e.g. *mummy-sg* (a singular noun inflection), *see-3sg* (a third-person singular verb inflection). Some words had irregular morphology; we modelled these as words with null inflections (e.g. *mice*). We also included an English-like system of pronouns, distinguishing person, number and nominative/accusative case.

The core of our 105-word vocabulary consisted of words commonly used by 16-30 month-old toddlers according to the Child Development Inventory (CDI, Fenson et al., 1994). The grammar of our language allowed for regular transitive sentences and also for two types of idiom (possible inflections not shown):

- continuous NP idioms (*teddy bear*, *Winnie the Pooh*, *play dough*, *ice cream*, *french fries*),
- discontinuous VP idioms (*kiss NP good bye*, *give NP a hug*, *give NP five*).

Note that these idioms do not all have the same degree of idiomaticity. For instance *give NP a hug* is not fully idiomatic; it contains a noun phrase (NP) 'slot' whose filler can have arbitrary (accusative) NP structure. Rather they exemplify the spectrum of possible idioms. However, there is some evidence that even phrases not considered idiomatic in adult language could be learned by children first as surface patterns or item-based constructions (Tomasello, 2003). Similarly, Pine & Lieven (1997) claim that although children use determiners with different noun types, there is no evidence for them possessing an adult-like syntactic category of determiners, which rather evolves gradually by broadening the range of lexically specific frames in which different determiners appear. Therefore, we omitted determiners (*a* and *the*) from our language, except for cases where they were part of an idiom, as in *give NP a hug* or *Winnie the Pooh.*

The language also featured semantic dependencies, in that all subjects were animate,[16] some verbs could only be followed by animate objects, others only by inanimate objects. It also contained synonyms and lexical ambiguities (the word *give* could be a part of either *give NP a hug* or *give NP five*, the word *hug* could be a regular verb as in *I hug-1sg you* or a part of the idiom *give NP a hug* and the word *kiss* could be either a regular verb as in *grandpa-sg kiss-3sg grandma-sg* or a part of an idiom with a different meaning as in *grandpa-sg kiss-3sg grandma-sg good bye*).

To allow for all mentioned phenomena, we made some extensions to the core CDI-based vocabulary. Out of the idioms used in our language, CDI explicitly

[16] We conceived teddy bears as animate too.

contains *teddy bear*, *play dough*, *ice cream*, *french fries* and *give me five*. It also contains single words *give*, *hug*, *kiss*, *good*, *bye* that we used in discontinuous idioms. We also added the word *rabbit* to feature as a synonym of *bunny*, and the idiom *Winnie the Pooh*.

Utterances of the target language were generated from a context-free grammar specifying syntactic constructions and words that could appear in specific positions (see Appendix B for details). The rules for inflections were as follows:

- All proper names were singular.[17]

- Person and number of the verb agreed with those of the subject. (We did not include tense inflections.)

- Nouns with irregular plural forms (e.g. *mice*), personal pronouns (*I*, *you*, *he*, *she*, *it*, *we*, *they*, *me*, *him*, *her*, *us*, *them*) and words appearing as fixed parts of idioms (e.g. *winnie*) all had null inflections.

Each target utterance was paired with an episode representation: a role frame associating agent, patient and action roles with sensorimotor signals. During training/generation, the role frame description was used to generate a sequence of sensorimotor signals in the episode rehearsal system (Table 2), while the target utterance was replayed from the phonological input buffer. For example, the sentence *We like-1pl mummy-sg* was paired with the role frame description

AG:PRON/1/PL, ACT:LIKE, PAT:MUMMY/3/SG

while the sentence *Winnie the Pooh-sg kiss-3sg Helen-sg good bye* was paired with

AG:WINNIETHEPOOH/3/SG, ACT:FAREWELL, PAT:HELEN/3/SG

Note that while in non-idiomatic sentences there is a one-to-one correspondence between words and concepts, multi-word idiomatic phrases are still represented by single concepts.

All personal pronouns were represented by a single concept PRON combined with an appropriate person and number.

The grammar could generate 127088 possible sentences, out of which approximately 20% contained idioms (13% continuous NP idioms and 6.4% discontinuous VP idioms).[18] To test the generalisation ability of the model, we only trained it on a small subset of all possible sentences (approx. 3%).

The basic language we have just described has SVO word order. To test the hypothesis that our model can acquire any possible word order, we created five

[17] We treated *mummy*, *daddy*, *grandpa* and *grandma* as proper names too, assuming they denote a particular person for an infant (i.e. his/her mummy, daddy, etc).

[18] A description of how the 'degree of idiomaticity' of utterances is determined is given in Section 6.2.5.

variant languages with SOV, VSO, VOS, OSV and OVS order. These languages were created by changing the word-ordering rules in the SVO grammar, but retaining the same English vocabulary and morphological rules.

In all our experiments we used 10 simulated 'model subjects'. Each subject was an instance of our model with network connections initialised to different random initial weights, and exposed to a its own training set containing 4000 stochastically generated sentences of the target language, and a test set containing another 4000 sentences of the target language (not present in the training set). In this way we modelled 10 individuals each with their own personal history of exposure to the same target language. All the model subjects were trained on their training sets for 30 epochs. The phonological input buffer was set to mature at around epoch 5, and learning in the control network learning was turned on at epoch 15. After each epoch, the weights were temporarily frozen and the models were tested for their ability to correctly generate sentences for meanings paired with the sentences in their test sets. The results were averaged over the 10 model subjects.

6.2. Results and discussion

When charting the linguistic development of a child, several separate metrics must be used, relating to vocabulary size, acquisition of surface language patterns, and acquisition of fully mature syntactic and morphological rules. In this section we evaluate the learning of our system using an array of metrics of these kinds.

6.2.1. Acquisition of open-class vocabulary

We begin by presenting some basic results about the model's learning of individual words. There are different ways this can be assessed. Most obviously we could simply inspect the word-production network in isolation, and measure the number of word meanings which are correctly mapped onto word forms. But it is more realistic to measure vocabulary by inspecting the model's output utterances. (This corresponds to the measure of 'active vocabulary' used in studies of child language.) We defined the **active vocabulary size** of the model in a given epoch as the number of word types which were produced correctly at least once during that epoch. A word was deemed 'correct' if it matched at least one of the semantic signals in the input episode (ignoring inflections). Active vocabulary development for the 10 SVO model subjects is charted in Figure 9.[19]

As the figure shows, after an initial peak, active vocabulary size rises steadily until there is a sudden jump at epoch 15, the epoch when the control network comes online. By the end of this epoch, the model is correctly producing all the

[19] We could also define vocabulary size as the number of word types which were *always* correct when produced, or at least correct most of the time. In fact, because our model only produces a word when it is confident about its correctness, it hardly ever produced incorrect words, so this definition produces a graph very similar to that in Figure 9.

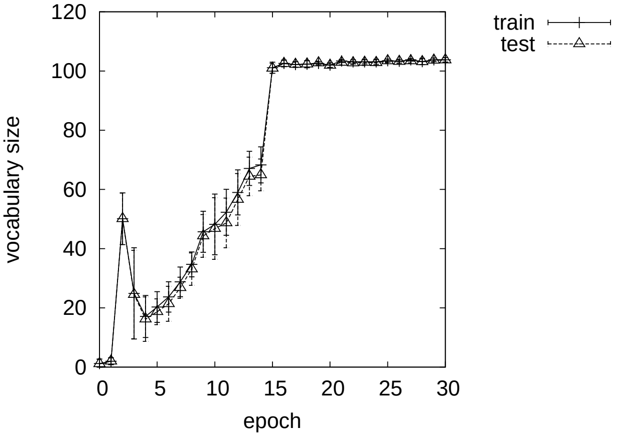

Figure 9: Active vocabulary size after each training epoch. The results are averaged over the 10 SVO model subjects.

words it can represent. The initial peak is an interesting effect, which has its origins in the way the phonological input buffer matures between epochs 0 and 5. As this happens, the entropy network temporarily becomes more conservative about producing words: the drop in vocabulary size between epochs 2 and 4 is actually due to a drop in the number of produced words rather than to any regression in the word-production/sequencing networks. We will discuss this effect in more detail in Section 7.4. The jump in vocabulary at epoch 15 is mainly due to the influence of the word sequencing network on word production. Recall that before epoch 15, the control network activates random contexts/phases. This means that the network is often called upon to generate words in syntactically inappropriate contexts, which results in low confidence in the sequencing network.

6.2.2. Acquisition of word-ordering conventions

A key novel element of our model is the control network, which learns the word-ordering conventions of the training language. We predict that this network will be able to learn any of the possible word orders. To test this prediction, we trained the model on the five variant languages featuring SOV, VSO, VOS, OSV, OVS word orders as well as on the original SVO language.

Acquisition of word-ordering conventions requires the control network to learn what contexts/phases should be inhibited. To verify that the models have really learned the conventions for all the word-orders, we inspected output values of the control network for all contexts/phases during sentence generation on the test set. Learning is very fast: 1-2 epochs after the control network begins training its output values for contexts stabilise, and do not change much thereafter. Figure 10 shows an example of learning for the VSO language. For each training language, the same inhibition pattern was learned by each model

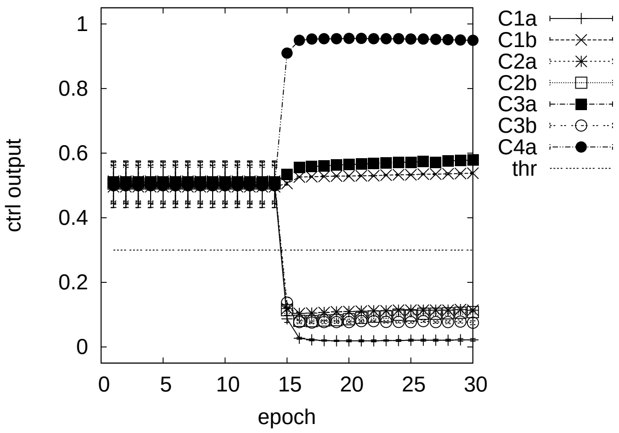

Figure 10: Control network output values for the model trained on the VSO language (averaged over 10 model subjects). After epoch 15, the network returns above-threshold activities (thr=0.3) for contexts C1b, C3a, C4a. For other word-order types, trends are similar, just with different above-threshold contexts (see Table 6).

subject: the learned inhibition patterns for the different languages are shown in Table 6. In each case, the inhibition pattern led to the right word-ordering convention; in other words, the control network learns a correct policy in 100% of cases for each possible language.

Note that for some word orders there are multiple possible inhibition patterns which give a correct result; for instance for SVO word order we could inhibit C2b–C4a, or C1a–C2b. Our match-based training algorithm results in a 'greedy' strategy, where words are pronounced on the first permissible occasion; see Section 7.2 for further discussion.

Since the control network only takes input from the context and phase representations of the episode rehearsal system, we also predict that the word-ordering conventions it learns will generalise well to episodes not encountered during training. We will test this prediction in Section 6.2.4, when we evaluate the system's ability to generate full sentences.

6.2.3. Acquisition of morphological agreement rules

Our model was also designed to learn morphological agreement rules. As discussed in Section 3.1, these rules exploit structure in the 'WM episode' and 'current object' areas of the episode-rehearsal system. The WM episode area, which delivers the semantics of inflected verbs, conveys fine-grained information about a planned motor action to the linguistic system, but also coarser-grained information about planned attentional actions to the agent and patient. The current object area, which delivers the semantics of inflected nouns, conveys information about an object, but also about the attentional action which delivered this information. In our account, grammatical person and number features

Table 6: Inhibition patterns learned by the control network for each word-order language type. '⇃' means an above-threshold activity for a given context (the 'pronounce' signal), '—' means an under-threshold activity (the 'withold' signal).

	SM signals in contexts						
	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Lang. type	AG	ACT	PAT	ACT	AG	ACT	PAT
SVO	⇃	⇃	⇃	—	—	—	—
SOV	⇃	—	⇃	⇃	—	—	—
VSO	—	⇃	—	—	⇃	—	⇃
VOS	—	⇃	⇃	—	⇃	—	—
OSV	—	—	⇃	—	⇃	⇃	—
OVS	—	—	⇃	⇃	⇃	—	—

express coarse-grained information about attentional actions. The fact that this information is present in WM episodes as well as in the current object area is what allows agreement between verbs and argument nouns. In our model, an 'agreement rule' in a given language is really just a policy about how much of this multiply presented information should be explicitly conveyed by nouns and verbs. This is what the word production/sequencing network must learn from the training language.

We define a word generated during by the system to be **morphologically incorrect** if it incorrectly expresses person/number information. For a word with regular inflections, this will mean an incorrect inflection; for an irregular word it will mean an incorrect word stem.[20] The graph in Figure 11 shows the proportion of the words generated in each epoch which were morphologically incorrect, averaged over 10 SVO model subjects.

The basic finding is that the model is successfully able to learn the morphology of the training language. This involves learning about subject-verb agreement rules, irregular plural nouns (e.g. *leaves* as the plural of *leaf*), and about the semantics of pronouns. However, it is also interesting to look at performance in relation to when the control network comes online, at epoch 15. The network clearly learns a good deal about morphology without help from the control network. But its performance is given a distinct boost by the control network, which essentially eliminates all of the remaining errors within a single epoch.

Note that while the model architecture has a potential for representing overregularisations, e.g. *leaf-pl* (*leafs*) or *tooth-pl* (*tooths*), we hardly observed any of these.

[20] Using an incorrect pronoun (e.g. *you* instead of *they*) also counts as morphologically incorrect on this metric, because the mistake relates solely to person/number information.

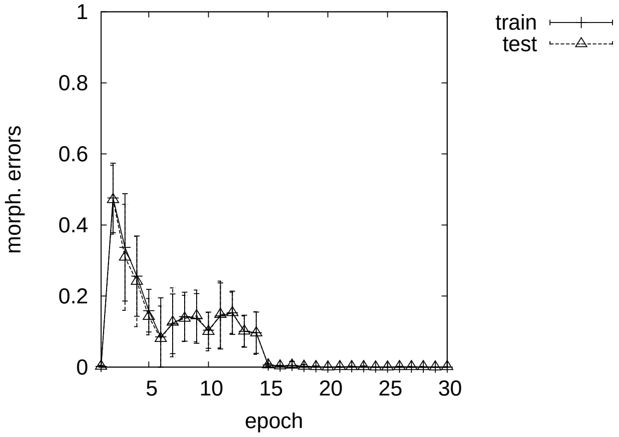

Figure 11: Proportion of words pronounced with morphological errors in sentences generated for meanings from training and test sets after each training epoch. Results are averaged over 10 SVO model subjects.

6.2.4. Overall accuracy of generation

Can our model achieve mature linguistic performance, i.e. can it be trained to generate fully correct sentences in the training language? We consider an utterance generated for a given meaning to be **correct** if all the roles (agent, patient, action) are expressed with semantically appropriate words, the sentence is syntactically correct (i.e. it complies with the transcription rules) and all the words have correct morphology (inflections). And we define the **generation accuracy** of a model being trained as the proportion of correct utterances it produces, evaluated either in relation to its training set of meaning-utterance pairs, or to an independent test set.

Figure 12 shows the generation accuracy of 10 model subjects trained on the SVO language. As is very obvious, the control network, which comes online in epoch 15, has a dramatic impact on generation accuracy: before it comes online, the model produces almost no correct sentences, but afterwards its accuracy improves to close to 100% within a few epochs. Before epoch 15, even though the model is learning vocabulary, morphology and surface regularities, it is not able to produce *fully* correct sentences. We should stress that during this time the model is not 'silent': it is producing a range of single-word and multi-word utterances, which often convey a good deal of the message to be expressed, as should be clear from Sections 6.2.1 and 6.2.3. (We will analyse these pre-syntactic utterances in more detail in Section 6.2.6.) So development is not as discontinuous as the generation accuracy measure seems to indicate. Nonetheless, prior to epoch 15, utterances are hardly ever *fully* correct.

After the jump at epoch 15, the learning curves tend to saturate, but they do not increase strictly monotonically; instead they show small fluctuations.

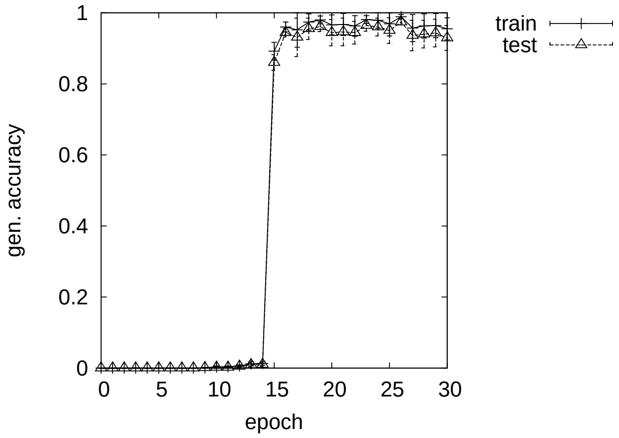

Figure 12: The generation accuracy—relative number of correct sentences generated after each training epoch for meanings from the training set and an independent test set, averaged over 10 model subjects trained on samples of the SVO language.

This is due to the complex interactions between multiple subnetworks, mainly caused by ongoing adaptations in the entropy network. To prevent overtraining, we considered each model subject as **fully trained** in the epoch in which it achieved the best generation accuracy on the independent test set.

On average, a model subject became fully trained after 24.2 training epochs (SD=3.12), and at this point achieved 99.4% (SD=0.1%) generation accuracy on the training set and 98.3% (SD=0.5%) on the test set. Given that each model subject was only trained on 3% of the training language, this suggests the network has very good generalisation abilities. This is largely due to the control network, whose word-ordering rules make no reference to the semantics of particular words.

6.2.5. Acquisition of surface regularities (idioms)

The target language contained a mixture of syntactic patterns (produced by abstract constituent-ordering rules) and surface linguistic patterns (expressing idiomatic constructions). We were interested whether the model learned both types of pattern equally well. We divided generated sentences into several groups: **regular sentences**—those that do not contain any idioms; **continuous NP idioms**—sentences with an idiomatic noun phrase in at least one of the agent/patient roles, and *not* containing a discontinuous verb idiom; and **discontinuous VP idioms**—sentences containing an idiomatic verb phrase (regardless of the presence or absence of continuous idioms in the sentence).

We measured the generation accuracy of fully trained model subjects for each group separately; the results are shown in Figure 13. The model performs well for all sentence types. Its good performance on discontinuous idioms is especially significant, given that these constructions only feature in about

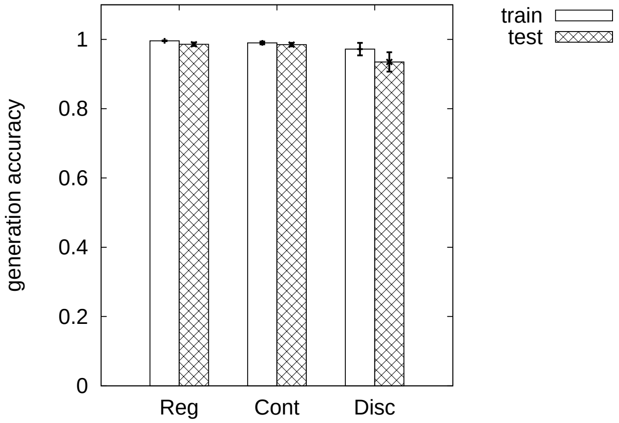

Figure 13: Generation accuracy of fully trained model subjects by idiomaticity type of generated sentences. Reg—regular (non-idiomatic) sentences, Cont—sentences with continuous NP idioms, Disc—sentences with discontinuous VP idioms.

6.4% of the training sentences.[21] The average generation accuracy on previously unseen episodes (test set) was 98.6% (SD=0.5%) for regular sentences, 98.5% (SD=0.5%) for continuous idiomatic sentences and 93.5% (SD=2.8%) for discontinuous idiomatic sentences.

It is interesting to examine how the model is able to generate discontinuous idioms. For instance, consider *Daddy-sg kiss-3sg me good bye.* 'Kiss X good bye' is a pattern of words collectively expressing the semantic signal FAREWELL/3/SG, but production of this pattern must be interrupted by production of the word *me*—which realises its own semantic signal, PRON/1/SG. Say the model has already produced the word *Daddy-sg*, and the episode-rehearsal network has just presented FAREWELL/3/SG for the first time. The WPSN will confidently predict the first word of the idiom, *kiss-3sg*. But when the surface context is updated, it is unable to make a confident prediction, since it needs information about the patient at this point. So we update the episode-rehearsal network until the context/phase which presents the patient signal, PRON/1/SG. At this point the WPSN can confidently predict *me*. The important thing is that we now update the surface context, to give the WPSN an opportunity to generate an idiomatic continuation. And in fact the WPSN can confidently predict *good* and then *bye*, the remainder of the discontinuous idiom. This is mainly due to learning in the sequencing network. Recall that this network

[21] In a training epoch the model is exposed to around 250 (SD=15.9) discontinuous idiomatic sentences, compared to 3235 (SD=28.1) non-idiomatic and 515 (SD=17.9) continuous idiomatic sentences (averaged over 10 SVO model subjects). The figures for the test sets are similar.

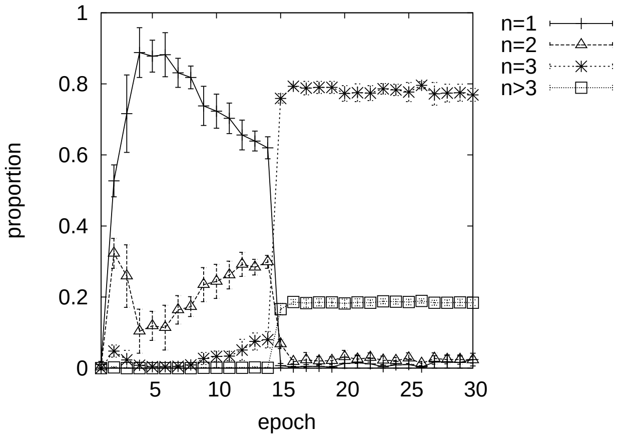

Figure 14: Proportion of sentences of the length n among sentences generated for an independent test set after each training epoch. Results are averaged over 10 SVO model subjects.

receives a sequence of word meanings as inputs, and learns to represent the relevant recently-presented word meanings in its context layer. So even though its current semantic input is some arbitrary object representation, its context layer still holds a record of the semantics of the partially produced idiom. Moreover, it knows that *. . . me good bye* is a common word sequence. This knowledge is sufficient for it to be confident about predicting *good* and then *bye*.

6.2.6. Early syntactic development

Before the model learns to correctly generate sentences with full-fledged syntax, it produces a range of single-word and fragmentary multi-word utterances. In this section we will look at these. The discussion will focus on development of word-ordering rules. For some comments about morphological development, see Section 7.3.

Figure 14 shows the proportions of utterances of length 1, 2, 3 and over 3 words generated for meanings in the test set after each training epoch. We can see that before the control network is turned on at epoch 15, one-word and two-word utterances predominate. Single-word utterances can reflect any aspect of the meaning to be expressed: agent, patient or action. Two-word utterances often reflect two components of meaning; in these cases they mostly have the form S V or O V (where S, V and O realise the agent, action and patient respectively). Sometimes they are idiomatic expressions reflecting the agent or patient (e.g. *teddy bear*). Sometimes they reflect an agent or patient and an action, but contain a fragment of an idiom (e.g. *teddy tickle* or *mummy* *give*[22]. And sometimes they result from the (incorrect) repetition of a word.[23]

The system's earliest semantically productive multi-word utterances appear to reflect 'item-specific' rules for word combination—i.e. rules which are tied to particular individual words. Children's early uses of grammatical constructions are typically item-based (see e.g. Tomasello, 2003). For instance, when a child first uses a particular transitive verb in combination with an object noun, the range of nouns which appear with it is often very restricted: we may see *kiss mummy* and *cuddle teddy* but not *kiss teddy*, even if there are 'kiss teddy' episodes to report. Empiricist linguists argue that mature grammatical rules are learned through a gradual process of generalisation away from concrete utterances, and see item-based constructions as reflecting early steps in this abstraction process. Our model's first multi-word utterances are distinctly item-based. For instance, after epoch 7, one of our SVO model subjects regularly produced *I draw* and *we help*, but not *we draw* or *I help*, even though the test set provided opportunities to express both these phrases. (A description of all the utterances produced by this subject at this epoch is given in Appendix C.) The model's item-based constructions reflect a mixture of learning in the word-production and word-sequencing networks. The words in an utterance must individually reflect semantic signals, but collectively they must conform to known sequential patterns in the training utterances.

Once the control network comes online, the model can learn rules about the structure of utterances which do not make reference to individual words. Note that the system's utterances must still conform to word-sequencing constraints; the control network's main role is to select sequences of semantic signals which *align* with surface word sequences. As shown in Figure 14, after epoch 15 the network rapidly learns to generate utterances of three or more words, to the virtual exclusion of shorter utterances. These longer utterances are by and large correct, both syntactically and semantically. Utterances with more than three words are those which contain idioms; e.g. *Winnie the Pooh-sg like-3sg ice cream-sg*. Note that the network learns to produce idiomatic sentences just as fast as non-idiomatic ones, even though there are fewer of them.

One final interesting feature of the 'mature' stage of language production after epoch 15 is that our model sometimes uses idiomatic forms to express productive combinations of concepts. For instance, the SVO model subjects sometimes generated a verb in context/phase C1b, as normal, but then continued to generate the object in this same context/phase, before the semantic signal for the object had actually been delivered. This happened in around 4%

[22] The verb *give* only appears in our language as a part of discontinuous idioms *give NP five* and *give NP a hug*

[23] Repeated words often occur because learning of word meanings in the word production network outpaces learning of surface patterns in the word sequencing network. For instance, the peak in the number of two-word utterances before epoch 4 is due to repetitions such as *you you*, which occur because the sequencing network does not generate a low enough confidence for the sequence *you you* to override the word production network's continued confident prediction of *you*.

of utterances. We surmise that the behaviour is due to sampling biases in the training set, which caused certain concepts to be combined in predictable ways. For instance, it may be that in a certain training set, if the agent was MUMMY and the action was EAT, the patient was always PIZZA. In such a case, our system could generate *pizza* as an idiomatic continuation of *Mummy eats.*[24] Something similar may happen in humans when expressing messages whose components combine with particularly high frequency (e.g. *How are you doing?*, *Get out of here!* etc). In fact, it has often been suggested that idioms have their origin in this kind of over-representation of particular concept combinations. It is interesting to see our network learning to express such combinations using surface word patterns.

6.2.7. Syntactic bootstrapping

A common proposal in models of language development is that once children start to learn the syntax of their exposure language, they become more efficient at learning individual words. The idea is commonly termed **syntactic bootstrapping** (see e.g. Landau & Gleitman, 1985). To illustrate, say a child growing up in an English environment hears the utterance *Gick glebs snuck.* Knowledge of basic English morphosyntax places strong constraints on the possible referents of each word: for instance, *gleb* probably denotes an action.

Our model demonstrates simple syntactic bootstrapping of this kind. Imagine the model is given a training utterance featuring a novel word. If this happens before the control network comes online, it associates the word with one particular sensorimotor signal from the associated episode (or perhaps with multiple signals, if the phonological input buffer is still immature). In either event the meaning of the new word can be learned over multiple exposures, because of weak cross-situational associations between signals and words, but this form of learning is very slow. However, if the model has already learned to appropriately synchronise replaying of a word sequence with episode rehearsal, the new word will only be a training target for the semantic signal which it actually denotes. This should make learning much faster.

To verify that our model can learn a novel word faster after having acquired syntactic knowledge, we designed a modification of our basic training regime. In the modified regime, we train the model initialised with random weights for a certain number of epochs on a sample of 3980 SVO sentences (paired with meanings) that do *not* contain a particular word (*dog*), then we add to the training set another 20 sentences—10 with *dog* in the subject position and 10 with *dog* in the object position, and continue training. After each epoch of training, the model is tested for sentence generation on 100 different sentence meanings—50 with *dog* in the subject position and 50 with *dog* in the object position.

The variable parameter was the epoch in which sentences with *dog* were

[24] Note that in context/phase C2a, when the semantic signal PIZZA is actually delivered, the network is silent, because the sequencing network is reluctant to produce two pizzas in a row.

introduced into the training set. In an 'early word learning' condition, we introduced the *dog* sentences at epoch 5, well before the control network comes online, and in a 'late word learning' condition, we introduced them at epoch 20, after it has fully learned the SVO word ordering convention. We were interested in how many training epochs are necessary to learn to use the new word. To separate word production performance from ability to correctly generate whole sentences, we recorded the relative number of times the model correctly predicted the word *dog* for the semantic input DOG in each epoch after *dog*-sentences were introduced, and also the relative number of times it was actually overtly pronounced for this signal. We considered the novel word successfully acquired when it was correctly predicted/produced in at least 75% of cases.

We created 10 model subjects with different initial weights and different training samples in the 'early learning' condition and 10 model subjects in the 'late word learning' condition, paired by training samples. The average number of training epochs necessary for correct prediction of the novel word was 8.7 (SD=2.06) for the early learning group and 2.6 (SD=0.52) for the late learning group. For correct pronunciation the figures for early and late learning groups were 16.7 (SD=1.70) and 9 (SD=1.76) respectively. Both these differences are statistically significant ($t_{18} = 9.931; p < 0.0001$ for pronunciation, $t'_{10.13} = 9.093; p < 0.0001$ for prediction).

Note that while our model shows some analogue of syntactic bootstrapping, we are not trying to model 'fast mapping', the learning of a new word in a single exposure (Carey & Bartlett, 1978). In our simulation it takes at least 2-3 epochs (40-60 exposures) to learn to predict a new word reliably. This is basically a result of our use of a back-propagation training regime.

6.2.8. Systematicity in generalisation

As already noted, even though the control network learns word-order rules which are independent of surface words, the word-sequencing network learns rules which retain reference to surface words. It is interesting to ask what happens when the conventions learned by these two systems conflict. The rules learned by our control network generalise easily to unseen utterances, but an Elman-style network has difficulty producing words in utterances which differ from those encountered during training. How does our combined model deal with such utterances?

Of course, a 'new utterance' can differ from training utterances to differing degrees. A useful distinction was proposed by Hadley (1994), between a new utterance which contains a word sequence not encountered during training, and a new utterance which uses a word in a syntactic position in which it was never encountered during training. Assume a toy training set containing the utterances *Dog bites man* and *Dog eats food.* The test utterance *Dog eats man* is new in the first sense, because the sequence *eats man* is novel. A network which can generalise to this type of novel utterance is said to show **weak systematicity**. The test utterance *Dog bites dog* is new in the second sense, because the word *dog* only ever appeared as a subject in the training utterances. A network which can generalise to this type of novel utterance is said to show **strong systematicity**: it can be said to have learned an abstract rule, defined with reference to syntactic categories rather than words. Of course, our control network by itself passes the strong systematicity test with flying colours: the rules it learns are abstract in just this way. But if we want our model to handle idiomatic constructions in mature language, we need to learn surface patterns as well. What we need is a model which shows systematicity *even though it also has a capacity to learn surface patterns.*

We will begin by discussing weak systematicity. To examine our complete model's ability to produce unseen word sequences, we created 10 new test sets called WS sets (1 for each model subject trained on the SVO language), each comprising 100 items featuring unseen action-patient combinations, and correspondingly, unseen word sequences. We generated these by altering one of the selectional restrictions in the standard grammar, so that actions which normally require animate patients were given inanimate ones. This resulted in sentences like *Helen-sg tickle-3sg banana-sg* and *We hug-1pl pizza-sg.* Not only have these sentences never been seen during training (which also holds for our standard test sets), but each of them contains a *transition* that has never been in any training sentence. We let the 10 fully trained SVO model subjects described in Section 6.2.4 generate sentences for meanings in these new WS test sets. For fully trained models, the average generation accuracy was 90.0% (SD=3.2%), compared with 98.3% on standard test sets (see Section 6.2.4). Our network clearly achieves weak systematicity.

We now consider strong systematicity. In this case, there is a much stronger conflict between the predictions of the content-independent control network and the word-sequencing network. To illustrate, consider the above scenario, where we give our model a training set in which a given noun (e.g. *dog*) only ever appears in subject position, and then ask it to generate a sentence where this word appears in object position. In the phase of episode rehearsal when the object noun should be produced, there will be a strong conflict between the predictions of the word-production network and those of the word-sequencing network. The former network will predict the word *dog* from the sensorimotor signal DOG which appears at this phase. The latter will predict a distribution of words which it has seen in object position, but this will not include *dog.* What we want is a way of giving precedence to the word-production network's predictions in a case like this.

In fact, our model does show some strong systematicity: its design allows it to override the sequencing network's inability to produce words in novel contexts, in some special circumstances. Our design makes the assumption that the network will only be called upon to generate a word in an unseen syntactic position *if the word is newly learned.* We assume that words are evenly distributed over the syntactic positions in which they can appear. (At least for subjects and objects, this is not implausible.) If this is the case, the only time when it will be necessary to produce a word in a new syntactic position is when the word is new. Now note that our network has a way of measuring how new a word is: this can be read directly from the entropy of the word-production network

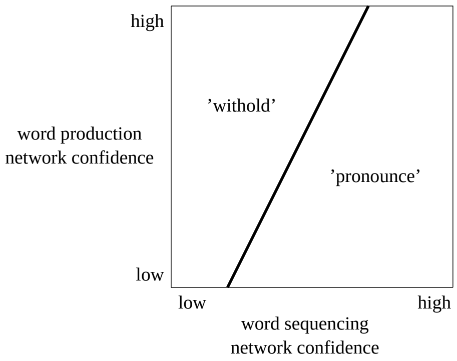

Figure 15: The input space of the entropy network in one of the model subjects after 15 epochs of training, with an indication of the learned confidence threshold.

when it is given the semantic signal associated with the new word as an input.[25] If a word has only been encountered a few times, the word-production network will be relatively unconfident in its prediction of this word from the associated semantic signal. (It will probably still make the right prediction, because this signal will not have been paired with any *other* words—but its confidence will nonetheless be lower than normal.) Recall from Section 4.2 that the entropy network takes the entropies of the word-production and word-sequencing networks separately as inputs, and learns a function mapping these entropies onto a pronounce/withold decision. There is scope for it to learn to ignore low confidence in the sequencing network if the confidence of the word-prediction network is also low. And in fact it does often learn exactly this. To illustrate, consider Figure 15, which approximates the function learned by the entropy network for one of our model subjects after 15 epochs of training. The two axes define the range of possible inputs it can receive: the confidence of the word-sequencing network is on the x axis, and that of the word-production network is on the y axis. The thick line separates inputs which lead to a 'withold' decision from those which lead to a 'pronounce' decision. As can be seen, a predicted word can be pronounced even if the word-sequencing network is relatively unconfident about it (i.e. in an unusual surface context)—but only if the word-production network is also relatively unconfident (i.e. if the word is newly learned). Hence we conjecture that at least in some phases of development, the model is able to produce words in genuinely new syntactic positions, but only if the words are newly learned.[26]

[25] In fact, this is the only reason we have to include the word-production network. If we do not need to identify new words, the word-sequencing network can learn mappings from word meanings to word forms as well as sequential patterns of words, and we can do without a specialised word-production network.

[26] We observed the characteristic slope shown in Figure 15 in all our model subjects, but just in some stages of development (usually between 10-15th epoch).

Of course our model implements a rather simple idea about how to reconcile the need to model surface language forms with the need to produce novel sentence structures. But we believe it is helpful to isolate the network component which learns word meanings from the component which learns word sequences, and to compute the entropies of these components separately.

7. General discussion

7.1. Related connectionist models

Our model of sentence processing combines a basic Elman network with a model of sequential attentional phenomena. There are several other connectionist models of sentence processing which augment a SRN to handle attentional sequences; in this section we will discuss two of these, and make some comparisons.

Mayberry et al. (2009) present an Elman-style recurrent network which performs sentence interpretation. It is designed to model the role of visual attention in online sentence interpretation—a topic which has been explored in several experiments (see Tanenhaus et al., 1995 and subsequent work). The network maps an input utterance presented one word at a time onto a static episode representation. As in our experiments, episodes are all transitive actions featuring an agent, action and patient. The innovation in the network is that it receives inputs from a simulated visual system as well as from a word sequence, and it generates an attentional signal at each iteration, which selectively gates these visually derived inputs. The visual input provides two episodes, only one of which is described by the input utterance, and the attentional signal can generate a bias towards one or other episode. After training, the network is able to accurately map word sequences onto interpretations, but more importantly its simulated attentional signals reproduce the anticipatory saccades made by human subjects during online processing of sentences (see Knoeferle & Crocker, 2006 for the human data). Beyond the fact that our model is of sentence generation rather than interpretation, the main difference between Mayberry *et al.*'s model and ours concerns the role of attentional operations. Mayberry *et al.* are concerned to model the attentional operations which happen during online language processing. Our attentional sequences model the attentional operations involved in the process of perceiving episodes—in its own right, a purely 'perceptual' rather than linguistic task. In the experimental task Mayberry *et al.*'s model simulates, episode apprehension and utterance interpretation happen in parallel, and interact in rich ways. The experimental task we are simulating is more like that studied by Griffin & Bock (2000)), where subjects first observe an event with no linguistic task, and then afterwards describe it. Griffin and Bock's study showed that the patterns of eye movements during these two tasks are quite different. Our episode-denoting sensorimotor sequences primarily model the former task, rather than the latter.[27] We view sensorimotor sequences primarily as language-independent semantic representations of episodes. Of course, we are also assuming that *rehearsed* sensorimotor sequences have a role in online sentence generation. We know from Griffin and Bock's study that during utterance generation, overt eye movements are closely tied to the surface structure of utterances: speakers tend to saccade to an object just before generating a noun phrase referring to it. Modelling these overt eye movements would require an extension to our network. In fact there is a fairly simple extension which would generate suitable overt eye movements: we could just specify that whenever the control network allows a word to be pronounced, it also executes an overt eye movement (if one is possible), to help amplify the semantic signal from which the word is produced. This would result in eye movements which anticipate the surface noun phrases in an utterance, whatever language it is in. So there is some scope for extending our system to model how the eye movements involved in event apprehension interact with those involved in online sentence processing, at least for sentence generation.

A more directly relevant connectionist model is that of Chang (2002). This model is a model of sentence generation, like ours. It uses a recurrent Elman-type network, augmented with an additional episode-denoting semantic input, to produce an utterance one word at a time. Like our network, it decouples the task of vocabulary learning from that of learning syntax, so that it can generate words in new syntactic positions. As in our network, this effect is achieved by presenting items from the semantic episode representation one at a time. At each iteration, the word production network maps the currently activated semantic item to a word, and this word is pronounced. There are two key differences between our network and Chang's. One concerns the mechanism which generates a sequence of semantic items for the word-production network, in accordance with syntactic word-ordering rules. In our system, semantic representations are inherently sequential, reflecting the sequential structure of the sensorimotor system, and moreover each component of an episode-denoting sequence is repeated; the language network simply has to learn which version of each component it should pronounce. In Chang's system, episode representations are static, and the syntax network has to learn to deliver the components of an episode sequentially to the word-production network. It does this by learning word-independent rules about the ordering of *semantic roles* in the exposure language. In Chang's network, an episode representation explicitly binds semantic role symbols such as AGENT and PATIENT to semantic objects such as DOG and CAT, so that activating a semantic role selectively activates the associated semantic object. The recurrent network in Chang's system operates on semantic role tokens, rather than on words, and learns to activate semantic roles

[27] In fact, Griffin and Bock found no systematic sequential structure of saccades in the former task. However, we argue that their use of static pictures as stimuli makes their results hard to interpret: it is hard to distinguish saccades executed as part of 'scene interpretation' from those involved in 'episode perception'. In an experiment using video stimuli, we found clear sequential structure in the perception of reach-to-grasp episodes; see Webb et al. (2010).

in the right order. The mechanisms which translate between words and semantic roles are quite complex, but they do result in content-independent ordering principles; Chang's network, like ours, achieves some degree of strong systematicity. In our network, semantic representations do not have to include explicit semantic role tokens; semantic roles are implicitly associated with positions in sensorimotor sequences. This allows a somewhat simpler mechanism for learning abstract word-ordering rules. But as regards abstract word-ordering rules, the main difference between our model and Chang's is that in ours, these rules concern how to 'read out' language-independent sequential semantic structures, while in Chang's they are abstract constructions learned from exposure data. The other difference between our model and Chang's is that our model is able to learn surface language structures as well as abstract syntactic rules. Our Elman-style network does the usual job of learning word sequences; in Chang's case it learns abstract sequences, and there is no mechanism for learning surface structures. But since Chang's model uses the medium of sequences for expressing abstract syntactic rules, it could be extended to deal with surface forms using the same sequence-synchronisation mechanisms as ours. Finally, we should note that Chang's model has much wider syntactic coverage than ours. See Section 7.5 for more discussion about coverage.

7.2. The control network and the LF-PF interface

In a Minimalist interpretation of our model, the job of the control network is to learn a mapping from LF to PF. However, it is worth noting that for some languages, the mapping which the control network learns is not the one standardly proposed in Minimalist accounts. SVO languages are a case in point. In these languages, the subject is standardly pronounced 'high', in IP, while the object is pronounced 'low', in the complement of VP; the verb is high in Romance languages and low in English. In our model, the control network learns to pronounce words at the first available opportunity; for a SVO training language it pronounces the subject and verb high, in IP, and the object high, in AgrP. This is simply an artefact of the narrow range of sentence types in our training language, together with the control network's 'greedy' training algorithm. If we included additional syntactic phenomena in the language, the control network would have to learn policies more in line with standard syntactic analyses in order to minimise its overall error. For instance, if we included adverbs and negation, appearing at positions in between AgrP and VP, as assumed in Pollock (1989) and subsequent work, then the control network would have to learn to pronounce verbs 'low' in English, where they occur after these elements. We plan to expand the system's coverage, but this is quite an incremental task, because each new 'sensorimotor sequence' must be justified in its own right. Again, see Section 7.5 for more discussion.

7.3. Morphology

Even though our model learns 'mature' morphological agreement rules, we have not made an effort to model the developmental stages through which these rules are acquired. However, there are a number of extensions to the system which can be envisaged. Many accounts of morphological development adopt a broadly empiricist perspective, assuming that morphological paradigms are acquired by progressive generalisation away from surface forms found in the exposure language (see e.g. Bittner et al., 2003). Our network, which allows for both surface-based and abstract morphological patterns to be learned, could provide a good framework for modelling this transition. One useful extension would be to introduce an entropy system for inflections. At present, entropy is only calculated for predicted word stems; if a word is pronounced, we simply pick the inflection which is most strongly predicted, regardless of how many competing alternatives there are. A separate entropy system for inflections would allow the model to generate bare a word stem when it cannot confidently predict an inflection, reproducing the fact that children's earliest words are frequently uninflected. Our network may also provide an interesting platform for modelling the relationship between morphological and syntactic development. It is often noted that mature syntactic forms begin to appear at around the same time as productive morphology (see e.g. Tomasello, 2003; Bittner et al., 2003). In our model, abstract morphological and syntactic rules both have their origins in the structure of rehearsed episodes, so the maturing of the episode-rehearsal system can potentially account for the emergence of productive morphological rules as well as abstract syntactic structures.

On a separate note, recall that in our network we model person and number agreement, but not gender. In our model, as discussed in Section 3.1, verb inflections reflect planned attentional actions in prefrontal cortex: we assume that these planned attentional actions only interface weakly with the linguistic system, so that only some of their attributes can be conveyed. We argued that person and number inflections signal salient attributes of attentional actions. But how does this suggestion extend to gender? Gender inflections signal quite arbitrary attributes of objects, which are harder to think of as attentional in origin. However, it should be borne in mind that attentional operations can involve the top-down activation of semantic representations as well as actual perceptual processes. For instance, when we search for an object, we must activate a representation of the searched-for object, to be matched against representations of the objects we perceive. Several studies have located these top-down semantic representations in prefrontal cortex (see e.g. Hasegawa et al., 2000). If we assume that top-down prefrontal semantic representations form part of attentional actions, but maintain the idea that the linguistic system can only read coarse semantic features from these representations, we should be able to extend our current model of the agreement system to cover gender features. Designing these extensions is a topic for further work.

7.4. The role of entropy in a model of language acquisition

One of the distinctive components of our model is the use of entropy. Our system learns to measure its own confidence in its ability to predict words, and if it is not confident enough, no words are produced. We introduced the entropy network for a specific purpose, to manage transfers of control between the word-sequencing and episode-rehearsal networks. But the notion of entropy may also have some interesting contributions to make to an account of language development, or development more generally.

Two points have already been noted in passing. Firstly, note that learning in the word-sequencing and production networks does not immediately produce overt behavioural changes. A word can be *reliably* predicted by these networks some time before it is predicted with enough confidence to be pronounced. It is interesting to ask whether something similar happens during actual language development. There are several developmental models which characterise children as 'conservative' word users; see in particular MacWhinney (1984) and subsequent work. The entropy network causes our model to generate utterances conservatively. In fact, the notion of conservativeness goes hand-in-hand with the notion of item-based constructions: as discussed by MacWhinney (2005), the production of item-based constructions is a natural result of a conservative learning strategy, which generalises only so far as the data permits. In our network, the generation of item-based constructions is in large measure a result of the entropy network. The constructions which are produced are those that slip over the confidence threshold, while those more productive combinations which are not produced are those that fall below this threshold. The system's gradual increase in confidence in particular combinations leads to the gradual broadening of item-based constructions.

A more novel effect of the entropy network is in producing 'nonmonotonic' effects in the learning of our model. For instance, as discussed in Section 6.2.1, the size of the model's active vocabulary sometimes decreases from one epoch to another. This happens because of the way in which the entropy network learns its confidence threshold. If an incorrectly predicted word is pronounced, the threshold is incrementally raised. When it is being raised, we can see a drop in the number of words being produced. This reflects adjustments in the entropy threshold, rather than any drop in the performance of the word-production/sequencing network. There are similar fluctuations in the system's overall accuracy in utterance generation, as mentioned in Section 6.2.4. We are not aware of any experimental evidence of nonmonotonic effects in language development, but such effects are certainly predicted by our model.

In fact, there is another factor which contributes to the large drop in vocabulary size discussed in Section 6.2.1. The drop coincides with a period of development in the phonological input buffer. Recall from Section 5.2 that the input buffer begins by presenting words from an incoming utterance in parallel to the production/sequencing network as training signals, but gradually shifts to a mode where these words are presented one at a time, and updated with an explicit 'advance' signal. This shift models a change in the task performed by the system: in the parallel mode it just needs to produce contextually relevant words, but in the serial mode it must produce words in an appropriate order. Since the system evaluates its own predictions by matching them against the phonological input buffer, its evaluation necessarily drops when the buffer starts to deliver words individually; this in turn causes the entropy network to learn a more stringent confidence threshold, which results in a drop in the number of words produced. It would be interesting to investigate whether something similar happens in children as they transition from producing single words to producing words in utterances. We suggest that if children's utterance generation algorithm uses a confidence metric, they might be expected to show these nonmonotonic effects.

7.5. Further work

There are still many questions raised by the current model. For one thing, we have not discussed any neural correlates of the different components of our network. Are there any neural regions with a role in language processing which might correspond to the episode-rehearsal, control and entropy networks? For another thing, since our model deals with syntactic knowledge deployed in the task of utterance generation, we should also say something about utterance interpretation; does it rely on the same mechanisms, and if so, how? We will not consider these questions here, but they are both discussed in Knott (in press).

Another pressing question concerns whether the network can be extended to deal with other syntactic constructions. We have only shown it works for transitive clauses. We plan to extend it to cover a range of other clause types. We also plan to cover nested syntactic structures—most obviously, full noun phrases, but also embedded clauses (in relative clauses and clausal complements). When we designed the network, we of course gave some thought to how it can be extended in these directions. To cover different clause constructions, the control network must be able to deliver a gating pattern specific to a particular clause type (transitive, intransitive, ditransitive and so on). We propose that clause type can be read off a WM episode, and that if the control network receives input from WM episodes as well as from context/phase, it will be able to learn policies specific to different clause types. (And these policies should still be content-independent, making no reference to individual words.)

As for embedded clauses—of course this is a thorny issue for any neural network. In fact we believe our use of sequences to encode semantic representations may allow an interesting new treatment of embedded constructions. One of the difficulties in modelling embedded clauses using a neural network is in representing their semantics. Many neural network models use a semantic representation which is designed to hold single propositions. (For instance, both Mayberry et al. (2009) and Chang (2002) model propositions as tuples mapping semantic roles like AGENT and ACTION onto values like DOG and CHASE.) An embedded clause contains more than one proposition, and therefore more than one set of role-value pairs: clearly these cannot be activated simultaneously in the 'proposition' medium without losing information about which roles belong together. But if they were activated *in sequence* within this medium, there would be no loss of information. In our model, semantic representations are inherently sequential: even a single proposition is encoded as a structured sequence of semantic signals. A natural way to extend this scheme to cover embedded clauses is to allow that semantic representations can involve sequences of *whole propositions* as well as of semantic signals denoting the components of propositions. To be concrete, we might envisage extending the network described in the current paper to allow that the semantic representation of a sentence can comprise a sequence of whole WM episodes, which is each individually rehearsed while it is active. This idea has been worked out to some extent for clausal complements in a paper by Caza & Knott (in preparation). We are currently exploring how the idea might work for relative clauses. The basic suggestion is that we may not need to represent all the component propositions in a nested message simultaneously; the appropriate relationships between the different propositions can perhaps be well modelled through the side-effects that individual propositions have while they are active. For example, Knott and Caza propose that 'Mother says that Mary runs' is represented by a sequence of two propositions, 'Mother says' and 'Mary runs', with the proviso that activating any proposition about 'saying' triggers a special cognitive mode where semantic signals are activated by words rather than by the world.

In general, given our basic hypothesis that LF structures reflect sequences of sensorimotor operations (or perhaps of cognitive operations more generally), any extension of coverage to a new construction has to be justified from two perspectives: on the syntactic side we must justify a particular LF analysis of this construction, but in addition to this we must also evidence for a corresponding sequence of sensorimotor/cognitive processes, drawing on separate evidence from psychology and neuroscience. Of course there is no guarantee that a corresponding sequence will be found—in which case our hypothesis will be falsified. But we believe the hypothesis defines an interesting research programme.

8. Summary and Conclusions

In this paper we have presented a neural network model of sentence generation which incorporates ideas from both nativist and empiricist models of language development. From the nativist tradition we take the idea that learning syntax involves learning to map a rich language-independent logical form onto a surface sequence of words, by setting discretely valued parameters. Our network's semantic representations of episodes correspond closely to logical form structures in Chomsky's Minimalist model: they contain analogues of right-branching X-bar structures, which make available multiple positions for the subject and object, as well as for the inflected verb. The control network learns whether to pronounce the subject, object and verb at their 'high' or 'low' LF positions: effectively, it learns discrete parameter values mapping logical forms onto word sequences. From the empiricist tradition we take the idea that learning syntax involves learning surface patterns in an exposure language: that early in development syntactic generalisations retain reference to individual words, and that a model of mature language must make reference to idiomatic surface forms as well as abstract syntactic generalisations. From empiricist linguists we also take the idea that studying language development should involve building computational simulations of language learners, which are exposed to complex and noisy training utterances. 'Chomskyan' and 'empiricist' models of language development are often seen as alternatives to one another, but we suggest that the above ideas can be quite successfully combined in a connectionist model.

The main innovation allowing an integration of nativist and empiricist ideas about language modelling is the use of sequences to encode semantic representations. This is a novel idea from both perspectives. From the perspective of Minimalism, it is innovative to interpret LF structures as (rehearsed) sequences of semantic signals. (In fact, Kayne (1994) suggests that the right-branching form of LF structures seen in Minimalism may have a temporal origin. Our interpretation can be seen as picking up on this suggestion.) From the perspective of connectionist models of language, it is innovative to represent episodes as canonically structured sequences of semantic signals. Representing episodes in this format provides a simple way of linking object representations to particular semantic roles. It also helps to express complex interactions between surface and abstract patterns in language in a format which is tractable for a language processing network. If both surface and abstract patterns are patterns in temporal sequences, then the interactions between them can be captured by devices which synchronise sequences. This is what happens in our combined network. Note that sequentially structured episode representations can be justified both in terms of their computational role in a neural network model of language, and as representations which allow the network's operations to be interpreted in Minimalist terms. This may not be a coincidence.

Note that our network architecture can also be understood as a model of the interface between language and the sensorimotor system. Again, sequentially structured episode representations are at the heart of this account. Our proposal is that experiencing an episode in the world requires a canonical sequence of sensorimotor operations—and that we represent episodes in working memory by storing these sequences. In our model, the 'episode rehearsal' system predates language, and works in the same way for speakers of any language: we assume that those aspects of language which are universal, and which are captured in the Minimalist account of LF, are in fact reflections of the sensorimotor system. From a Minimalist perspective, this means we can offer an interesting new account of the neural underpinnings of linguistic universals. Our suggestion in this paper is that the language-independent structure of LF does not reflect the operation of a modular language acquisition device, but rather various properties of the sensorimotor system and of sensorimotor working memory. From a connectionist perspective, our sensorimotor model of sentence semantics means we can give an unusually detailed account of 'where the semantic representations in our model come from'. The structure of episode representations—at least, concrete transitive ones—is motivated in detail in the model of sensorimotor processing and working memory given in Knott (in press).

In summary, the hypothesis that episodes are represented by rehearsed sensorimotor sequences may have the potential to draw together a number of different theoretical perspectives on language and its neural implementation. We hope that readers of this journal find this an appealing prospect.

Acknowledgement

This research was supported by a New Zealand BuildIT postdoctoral fellowship grant for Martin Takac.

Appendix A. Technical description of the composite network

In this section we will describe the modules of the complete network shown in Figure 8 in more detail, as well as the training and sentence generation algorithms.

The **episode rehearsal system** is a layer of input neurons with 1-hot localist coding in each of the four parts: the **episode context** (4 neurons coding contexts $C1,\ldots,C4$), the **phase** (2 neurons coding phases a,b), the **WM episode** (3+2 neurons for person (1,2,3) and number (Sg,Pl) of the agent, 3+2 for person and number of the patient, 34 neurons coding possible motor actions), and the **current object** (3+2 neurons for person and number, 46 neurons coding possible objects).

The **word production network** consists of one layer of linear perceptrons taking input from all the units in the WM episode and the current object parts of the episode rehearsal system (95 neurons). The connections are gated by the phase generator in the way that input from the WM episode part is blocked and that from the current object is let through in the phase a (and vice versa in the phase b). The output neurons are grouped in two blocks: one representing the next word stem (localist coding—106 units for all possible word stems, including one unit representing a conventional 'utterance-boundary'[28] signal), the other possible word inflections[29] (9 units, one of them representing null inflection). Activities of linear neurons in each of the blocks are combined using the softmax function

$$p_i = \frac{\exp(o_i)}{\sum_j \exp(o_j)} \ ,$$

where o_i, o_j are activities of the linear output neurons, j ranges over all neurons in the block, and p_i is the resulting activity of the i-th neuron. Hence, combined activities in each block sum to 1 and so can be treated as probability distributions. The word production network is trained using the delta rule (Widrow & Hoff, 1960) for error minimisation.

The **word sequencing network** is a recurrent neural network with one hidden layer of 100 units with a sigmoidal activation function. It is connected to the same input as the word production network (including the gating). The recurrent connections are mediated through a surface context layer (100 units),

[28] The 'utterance-boundary' or 'period' signal is the last element of the sequence of word in each training utterance. It allows the trained network to explicitly predict the end of the sentence, which is utilised for early stopping in sentence generation (after having generated the 'utterance boundary', the network proceeds to the next episode).

[29] The possible inflections were *-sg*, *-pl* (for nouns), *-1sg*, *-2sg*, *-3sg*, *-1pl*, *-2pl*, *-3pl* (for verbs), and *null*.

which carries a copy of activities of the hidden layer from the previous time step. The output layer of 106+9 linear neurons has exactly the same structure as that of the word production network. The network is trained using the back propagation through time (BPTT) algorithm (Werbos, 2002) with a time window of size 3.

The layer aggregating outputs from the word production and word sequencing networks also has 106+9 neurons, and the activity of each aggregated unit is computed as a simple average of the activities of corresponding units in the two output layers.

The **phonological input buffer** holds a sequence of words (an utterance that the infant heard), which are activated one by one and serve as a source of training signal for other subnetworks. The currently active (actual) word is accessible in a layer of units of the same structure as the output layers of the word production/sequencing networks (and their aggregated output), i.e. 106 units representing a word stem and 9 units representing an inflection.

We assume that the sequencing ability of the phonological input buffer is not mature from the very beginning, but matures gradually. In early stages, the activity in the actual next word layer is a noisy blend of all words in the sequence:

$$\vec{v}_i = \sum_{j=1}^{|U|} g_p(i,j)\vec{u}_j \ ,$$

where $\vec{u}_j$ is the j-th word in the sequence (represented as a vector of 1-hot localist code of the word stem concatenated with the code of the word inflection), $|U|$ is the length of the word sequence, and $\vec{v}_i$ is the i-th representation activated in the actual next word layer.

$$g_p(i,j) = \exp(-p(i-j)^2)$$

is a Gaussian neighbourhood function with parameter p regulating the width of the Gaussian. The most strongly present representation in $\vec{v}_i$ is $\vec{u}_i$, then $\vec{u}_{i-1}$ and $\vec{u}_{i+1}$ etc, decreasing with the distance between time points i and j. The parameter p is initially zero (a Gaussian with infinite width) and all words are represented in the actual word layer with the same strength, which models a cross-situational concept of associating a current sensorimotor signal with *all* words heard within some time span. The p increases with time and provides a smooth transition from ‘associate with all words’ training mode to ‘associate with the current item in the sequence’.

The **entropy network** is a feed-forward network with two input units, one hidden layer of three neurons with hyperbolic tangent activation function and one sigmoidal output neuron. The input units represent the **entropy** in word stem parts of the word production and word sequencing networks computed as

$$H = -\sum_{i=1}^{b} p_i \log_b p_i \ ,$$

where the logarithm base $b = 106$ is the size of the word-stem part, p_i are output activities of units in the word stem block after application of the softmax combination function.

The network is trained on a match between the aggregated next word stem and an actual next word stem representation in the phonological input buffer, using simple back propagation (Rumelhart et al., 1986). The training signal is 1, if the phonological output buffer is not empty and the cosine between vectors representing the two word stems is bigger than 0.5, otherwise it is 0. The output of the entropy network has a gating and mode-switching function (see Section 5.1) and is interpreted as a 'let through' signal if greater than 0.5.

The **control network** is a feed-forward network with one hidden layer of three neurons with hyperbolic tangent activation function and one sigmoidal output neuron. The network takes its input from episode context (4 units) and phase (2 units) parts of the episode rehearsal system. It is trained on the same match signal as the entropy network, using back propagation. Like in the entropy network, the output neuron activity has a gating and mode-switching function (see Section 5.1) and is interpreted as 'let through' signal if greater than 0.3.[30]

All training algorithms use the same learning rate (0.1) and zero momentum.[31] Connection weights in all networks are initialised with random values between $(-0.5, 0.5)$, the surface context layer of the word sequencing network is initialised with random values from $(0, 1)$. In addition to that, before each new episode rehearsal, the word sequencing network makes five 'dummy' passes on inactive episodic input (all zeros) to reset the hidden layer history (i.e. set the 'start-of-sentence' surface context) and eliminate the influence of the previous episode. The model is trained for 30 epochs, the control network comes online since epoch 15. The annealing/maturation parameter p of the phonological input buffer rises linearly from 0 in the first epoch to 8 in the final epoch of training.

Recall that the complete network alternates between two different modes when processing training sentences and when generating test sentences (see Sections 5.1 and 5.2). In one mode, there are iterations in the episode rehearsal system, and in the other mode, there are iterations in the word-sequencing network. Flow charts for the training and generation algorithms showing the two modes are given in Figures A.16 and A.17. Note that in the training algorithm, there is a forward pass through the WPSN in each phase of episode rehearsal, generating a 'predicted word' to compare with the current word in the training utterance and create the Boolean match signal which trains the control network. In the generation algorithm, the control network is already trained, so there is only a forward pass through the WPSN in contexts/phases

[30] The threshold is lower than 0.5 to boost learning in early phases, when the WPSN does not yield good predictions yet.

[31] The parameter values have been determined experimentally. Generally, the model is not very sensitive to learning rates. We have experimented with several sizes of hidden layers and have chosen such that yield the best performance at the lowest possible computational cost.

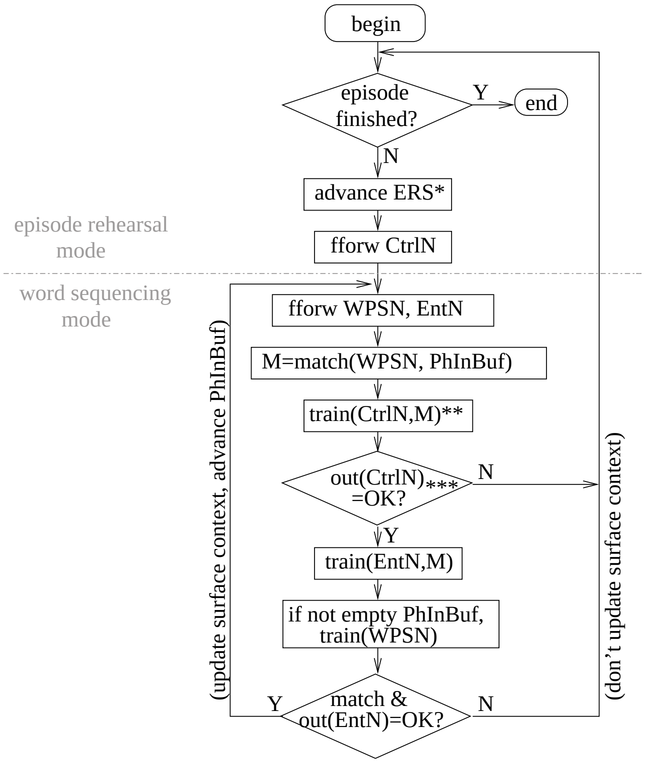

Figure A.16: Training algorithm for the complete model. Abbreviations: PhInBuf—the phonological input buffer (other abbreviations are explained in a legend to Fig. A.17). The first call of 'advance ERS' (*) puts the network into the C1a context/phase. Control network training (**) is skipped in the first developmental stage (before the control network goes online). Also, the output of the CtrlN (***) is substituted with a random signal in that stage.

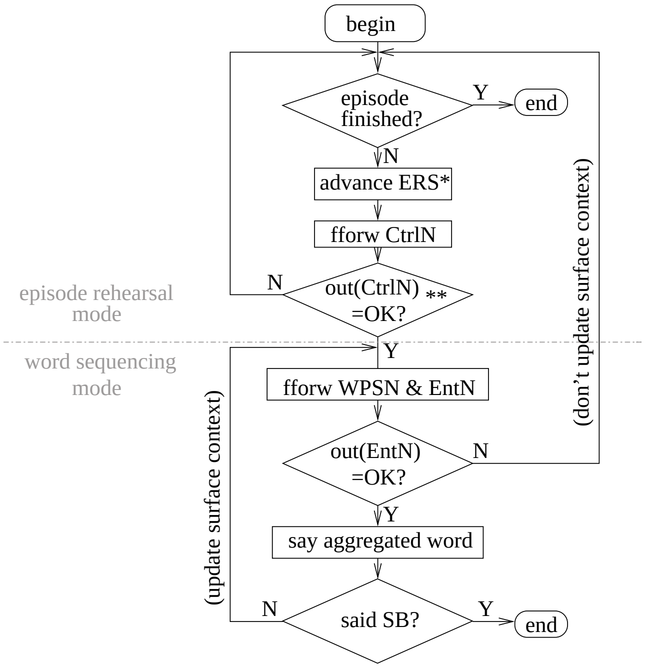

Figure A.17: The utterance generation algorithm in the complete model. Abbreviations: ERS—episode rehearsal system, CtrlN—the control network, fforw—feed forward pass, WPSN—the aggregated word production/sequencing network, EntN—the entropy network, SB—sentence boundary signal. The first call of 'advance ERS' (*) puts the network into the C1a context/phase, subsequent calls iterate through C1b, C2a, C2b, C3a, C3b, C4a. The output of the CtrlN (**) is substituted with a random signal in the first developmental stage (before the control network goes online).

Table B.7: Transcription rules for the syntax of the language used in our simulations. All non-terminals are written in all capital letters (TRANSITIVE is the initial non-terminal), all terminals contain small letters. Period (.) is a terminal and stands for the sentence boundary (SB).

TRANSITIVE	→	SUBJ VERB_GEN OBJ_GEN . | SUBJ VERB_INANIM OBJ_INANIM . | SUBJ VERB_ANIM OBJ_ANIM . | SUBJ kiss OBJ_ANIM good bye . | SUBJ give OBJ_ANIM five . | SUBJ give OBJ_ANIM a hug .
SUBJ	→	ANIM_NP | S_PRONOUN
OBJ_GEN	→	OBJ_ANIM | INANIM_NP
OBJ_ANIM	→	ANIM_NP | O_PRONOUN
ANIM_NP	→	mummy | daddy | Samko | Mia | Helen | grandma | grandpa | nanny | Winnie the Pooh | man | men | woman | women | mouse | mice | fish | goose | geese | dog | kitty | duck | bunny | rabbit | cow | pig | bug | puppy | bee | monkey | teddy bear
INANIM_NP	→	ball | book | balloon | toy | doll | block | crayon | pen | play dough | ice cream | cookie | banana | apple | cheese | cracker | bread | pizza | leaf | leaves | tooth | teeth | french fries
S_PRONOUN	→	I | you | he | she | it | we | they
O_PRONOUN	→	me | you | him | her | it | us | them
VERB_GEN	→	see | love | hold | bite | wash | hit | push | like | draw | hide | kick | carry | watch | find | wipe | touch | share | pull | lick | pick
VERB_ANIM	→	kiss | tickle | hug | help | feed | chase
VERB_INANIM	→	break | throw | buy | drop

where a word is to be overtly pronounced.

Appendix B. The target SVO language

Utterances in training and test sets for our SVO model subjects were stochastically generated by the rules of a context-free grammar shown in Table B.7. The rules were assigned different probabilities (not shown in the table) to ensure balanced generation and a sufficient number of idiomatic sentences. Morphological inflections were than added, respecting subject-verb agreement and irregular plurals.

Examples of sentences composed of single words, continuous idioms, and discontinuous idioms (with morphological inflections added) are given below. Note that a discontinuous VP idiom can be interleaved with a continuous NP one. *Mummy-sg love-3sg me. I like-1sg ice cream-sg. Helen-sg tickle-3sg Winnie* *the Pooh-sg. Grandpa-sg give-3sg grandma-sg a hug. Daddy-sg kiss-3sg teddy bear-sg good bye.*

Appendix C. Generated multi-word expressions

Here we list 240 utterances generated after 7 epochs of training in one of the model subjects trained on the SVO language. We use the following concise notation: words in curly brackets {} mean that any of them can be substituted into an utterance, e.g. "X {Y, Z}" is a shortcut for "X Y", "X Z" (potentially appearing multiple times). A morphological inflection in parentheses means that in some cases the model failed to generate it (i.e. generated null inflection instead).

1-word utterances (197) "{*I*, *we*, *you*, *women*}"

2-word utterances (41) "*I* {*bite-v1sg*, *draw-v1sg*, *hit(-v1sg)*, *kick-v1sg*, *kiss-v1sg*, *pick-n3sg*, *push*, *see*, *wash-v1sg*}",
"*we* {*bite*, *help*, *hold-v1pl*, *like*, *pull-v1pl*, *wash-v1pl*}",
"*you* {*bite*, *buy-v2sg*, *give*, *hit(-v2sg)*, *hold(-v2sg)*, *like*, *pick-n3sg*, *push(-v2sg)*, *watch*, *you*}",
"*women* {*kick-v3pl*, *push-v3pl*, *wash*}

3-word utterances (2) "*women watch-v3pl women*",
"*you you you*"

References

Anderson, M. (2010). Neural re-use as a fundamental organizational principle of the brain. *Behavioral and Brain Sciences*, *33*, 245–313.

Averbeck, B., Chafee, M., Crowe, D., & Georgopoulos, A. (2002). Parallel processing of serial movements in prefrontal cortex. *Proceedings of the National Academy of Sciences of the United States of America*, *99*, 13172–13177.

Averbeck, B., & Lee, D. (2007). Prefrontal correlates of memory for sequences. *Journal of Neuroscience*, *27*, 2204–2211.

Baddeley, A. (2000). The episodic buffer: a new component of working memory? *Trends in Cognitive Sciences*, *4*, 417–423.

Ballard, D., Hayhoe, M., Pook, P., & Rao, R. (1997). Deictic codes for the embodiment of cognition. *Behavioral and Brain Sciences*, *20*, 723–767.

Barsalou, L. (2008). Grounded cognition. *Annual Review of Psychology*, *59*, 617–645.

Barsalou, L., Simmons, W., Barbey, A., & Wilson, C. (2003). Grounding conceptual knowledge in modality-specific systems. *Trends in Cognitive Sciences*, *7*, 84–91.

Bittner, D., Dressler, W., & Kilani-Schoch, M. (2003). *Development of verb inflection in first language acquisition: a cross-linguistic perspective*. Berlin: Mouton de Gruyter.

Braine, M. (1976). Children's first word combinations. *Monographs of the society for research in child development*, *41*, 1–104.

Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence*, *47*, 139–159.

Browman, C., & Goldstein, L. (1995). Dynamics and articulatory phonology. In R. Port, & T. van Gelder (Eds.), *Mind as motion: Explorations in the dynamics of cognition* (pp. 175–193). Cambridge, MA: MIT Press.

Burgess, N., & Hitch, G. (1999). Memory for serial order: a network model of the phonological loop and its timing. *Psychological Review*, *106*, 551–581.

Carey, S., & Bartlett, E. (1978). Acquiring a single new word. In *Proceedings of the Stanford Child Language Conference 15* (pp. 17–29).

Caza, G., & Knott, A. (in preparation). Pragmatic bootstrapping: a neural network model of vocabulary acquisition. Manuscript.

Cernansky, M., Makula, M., & Benuskova, L. (2007). Organization of the state space of a simple recurrent neural network before and after training on recursive linguistic structures. *Neural Networks*, *20*, 236–244.

Chang, F. (2002). Symbolically speaking: a connectionist model of sentence production. *Cognitive Science*, *26*, 609–651.

Chen, S., & Goodman, J. (1998). *An empirical study of smoothing techniques for language modeling*. Technical Report TR-10-98 Harvard University Cambridge, MA.

Chomsky, N. (1981). *Lectures on government and binding*. Dordrecht: Foris.

Chomsky, N. (1995). *The Minimalist program*. Cambridge, MA: MIT Press.

Christiansen, M., & Chater, N. (1999). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, *23*, 157–205.

Dell, G. (1986). A spreading-activation theory of retrieval in sentence production. *Psychological Review*, *93*, 283–321.

Dominey, P., Hoen, M., Blanc, J. M., & Lelekov-Boissard, T. (2003). Neurological basis of language and sequential cognition: evidence from simulation, aphasia and ERP studies. *Brain and Language*, *86*, 207–225.

Dominey, P., Hoen, M., & Inui, T. (2006). A neurolinguistic model of grammatical construction processing. *Journal of Cognitive Neuroscience*, *18*, 2088–2107.

Dowty, D. (1991). Thematic proto-roles and argument selection. *Language*, *67*, 547–619.

Elman, J. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Fadiga, L., Craighero, L., Buccino, G., & Rizzolatti, G. (2002). Speech listening specifically modulates the excitability of tongue muscles: a TMS study. *European Journal of Neuroscience*, *15*, 399–402.

Fagg, A., & Arbib, M. (1998). Modeling parietal-premotor interactions in primate control of grasping. *Neural Networks*, *11*, 1277–1303.

Feldman, J., & Narayanan, S. (2004). Embodiment in a neural theory of language. *Brain and Language*, *89*, 385–392.

Fenson, L., Dale, P., Reznick, J. S., Thal, D., Bates, E., Hartung, J., Pethick, S., & Reilly, J. (1994). Variability in early communicative development. *Monographs of the Society for Research in Child Development*, *59*, i–185.

Fromkin, V. (1973). The non-anomalous nature of anomalous utterances. In V. Fromkin (Ed.), *Speech errors as linguistic evidence* (pp. 215–242). The Hague: Mouton.

Gallese, V., & Goldman, A. (1998). Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, *2*, 493–501.

Gathercole, S., & Baddeley, A. (1990). The role of phonological memory in vocabulary acquisition: A study of young children learning new names. *British Journal of Psychology*, *81*, 439–454.

Glenberg, A., & Kaschak, P. (2002). Grounding language in action. *Psychonomic Bulletin and Review*, *9*, 558–565.

Goldberg, A. (Ed.) (1995). *Constructions. A Construction Grammar approach to argument structure*. Chicago: University of Chicago Press.

Grèzes, J., & J, D. (2001). Functional anatomy of execution, mental simulation, observation and verb generation of actions: a meta-analysis. *Human Brain Mapping*, *12*, 1–19.

Griffin, Z., & Bock, K. (2000). What the eyes say about speaking. *Psychological science*, *11*, 274–279.

Hadley, R. (1994). Systematicity in connectionist language learning. *Mind and Language*, *9*, 247–272.

Harnad, S. (1990). The symbol grounding problem. *Physica D*, *42*, 335–346.

Hartley, T., & Houghton, G. (1996). A linguistically constrained model of short-term memory for nonwords. *Journal of Memory and Language*, *35*, 1–31.

Hasegawa, R., Matsumoto, M., & Mikami, A. (2000). Search target selection in monkey prefrontal cortex. *Journal of Neurophysiology*, *84*, 1692–1696.

Iacoboni, M. (2006). Visuo-motor integration and control in the human posterior parietal cortex: Evidence from TMS and fMRI. *Neuropsychologia*, *44*, 2691–2699.

Jackendoff, R. (2002). *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford: Oxford University Press.

Jeannerod, M. (2001). Neural simulation of action: a unifying mechanism for motor cognition. *NeuroImage*, *14*, S103–S109.

Jelinek, F., & Mercer, R. (1980). Interpolated estimation of Markov source parameters from sparse data. In E. Gelsema, & L. Kanal (Eds.), *Pattern recognition in practice* (pp. 381–397). Amsterdam: North-Holland.

Jellema, T., Baker, C., Wicker, B., & Perrett, D. (2000). Neural representation for the perception of the intentionality of actions. *Brain and Cognition*, *44*, 280–302.

Jordan, M., & Wolpert, D. (2000). Computational motor control. In M. Gazzaniga (Ed.), *The new cognitive neurosciences* (pp. 71–118). MIT Press.

Joshi, A., & Schabes, Y. (1997). Tree-adjoining grammars. In G. Rosenberg, & A. Salomaa (Eds.), *Handbook of formal languages* (pp. 69–123). Berlin: Springer.

Kayne, R. (1994). *The antisymmetry of syntax*. Cambridge, MA: MIT Press.

Knoeferle, P., & Crocker, M. (2006). The coordinated interplay of scene, utterance, and world knowledge: Evidence from eye-tracking. *Cognitive Science*, *30*, 481–529.

Knott, A. (in press). *Sensorimotor cognition and natural language syntax*. Cambridge, MA: MIT Press. Currently available at http://www.cs.otago.ac.nz/staffpriv/alik/publications.html.

Landau, B., & Gleitman, L. (1985). *Language and experience: Evidence from the blind child*. Cambridge, MA: Harvard University Press.

Ma, W. J., Beck, J. M., Latham, P. E., & Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nat Neurosci*, *9*, 1432–8.

MacWhinney, B. (1984). Where do categories come from? In C. Sophian (Ed.), *Origins of cognitive skills* (pp. 407–418). Hillsdale, NJ: Lawrence Erlbaum Associates.

MacWhinney, B. (2005). Item-based constructions and the logical problem. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition* (pp. 53–68).

Marcus, G. (2001). *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. Cambridge, MA: MIT Press.

Mayberry, M., Crocker, M., & Knoeferle, P. (2009). Learning to attend: a connectionist model of situated language comprehension. *Cognitive Science*, *33*, 449–496.

Murata, A., Gallese, V., Luppino, G., Kaseda, M., & Sakata, H. (2000). Selectivity for the shape, size, and orientation of objects for grasping in neurons of monkey parietal area aip. *Journal of Neurophysiology*, *83*, 2580–2601.

Novick, J., Trueswell, J., & Thomson-Schill, S. (2005). Cognitive control and parsing: Reexamining the role of Broca's area in sentence comprehension. *Cognitive, Affective and Behavioural Neuroscience*, *5*, 263–281.

Oztop, E., & Arbib, M. (2002). Schema design and implementation of the grasp-related mirror neuron system. *Biological Cybernetics*, *87*, 116–140.

Pelphrey, K., Morris, J., Michelich, C., Allison, T., & McCarthy, G. (2005). Functional anatomy of biological motion perception in posterior temporal cortex: An fmri study of eye, mouth and hand movements. *Cerebral Cortex*, *15*, 1866–1876.

Perani, D., Cappa, S., Schnur, T., Tettamanti, M., Collina, S., Rosa, M., & Fazio, F. (1999). The neural correlates of verb and noun processing - a pet study. *Brain*, *122*, 2337–2344.

Pine, J., & Lieven, E. (1997). Slot and frame patterns in the development of the determiner category. *Applied Psycholinguistics*, *18*, 123–138.

Plate, T. (2003). *Holographic reduced representations. CSLI Lecture Notes Number 150*. Stanford, CA: CSLI Publications.

Pollard, C., & Sag, I. (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Pollock, J.-Y. (1989). Verb movement, universal grammar and the structure of IP. *Linguistic Inquiry*, *20*, 365–424.

Pulvermüller, F., & Asollahi, R. (2007). Grammar or serial order?: Discrete combinatorial brain mechanisms reflected by the syntactic mismatch negativity. *Journal of Cognitive Neuroscience*, *19*, 971–980.

Pulvermüller, F, & Knoblauch, A. (2009). Discrete combinatorial circuits emerging in neural networks: A mechanism for rules of grammar in the human brain. *Neural networks*, *22*, 161–172.

Pulvermüller, F, Lutzenberger, W., & Preissl, H. (1999). Nouns and verbs in the intact brain: Evidence from event-related potentials and high-frequency cortical responses. *Cerebral Cortex*, *9*, 497–506.

Rhodes, B., Bullock, D., Verwey, W., Averbeck, B., & Page, M. (2004). Learning and production of movement sequences: behavioral, neurophysiological, and modeling perspectives. *Human Movement Science*, *23*, 699–746.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations* (pp. 318–362). Cambridge, MA, USA: MIT Press.

Shallice, T., Rumiati, R., & Zadini, A. (2000). The selective impairment of the phonological output buffer. *Cognitive Neuropsychology*, *17*, 517–546.

Shapiro, K., & Caramazza, A. (2001). The representation of grammatical categories in the brain. *Trends in Cognitive Sciences*, *7*, 201–206.

Shastri, L., & Ajjanagadde, V. (1993). From simple associations to systematic reasoning. *Behavioral and Brain Sciences*, *16*, 417–494.

Siskind, J. (1996). A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, *61*, 39–91.

Takac, M., Knott, A., & Benuskova, L. (2010). *Generation of idioms in a simple recurrent network architecture*. Technical Report OUCS-2005-02 Department of Computer Science, University of Otago Dunedin, New Zealand.

Tanenhaus, M., Spivey-Knowlton, M., Eberhard, K., & Sedivy, J. (1995). Integration of visual and linguistic information in spoken language comprehension. *Science*, *268*, 1632–1634.

Tanji, J., Shima, K., & Mushiake, H. (2007). Concept-based behavioral planning and the lateral prefrontal cortex. *Trends in Cognitive Sciences*, *11*, 528–534.

Tipper, S., Lortie, C., & Baylis, G. (1992). Selective reaching: Evidence for action-centred attention. *Journal of Experimental Psychology: Human Perception and Performance*, *18*, 891–905.

Tomasello, M. (2003). *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press.

Tootell, R., Dale, A., Sereno, M., & Malach, R. (1996). New images from human visual cortex. *Trends in Neurosciences*, *19*, 481–489.

Tranel, D., Adolphs, R., Damasio, H., & Damasio, A. (2001). A neural basis for the retrieval of words for actions. *Cognitive Neuropsychology*, *18*, 655–674.

van der Velde, F., & de Kamps, M. (2006). Neural blackboard architectures of combinatorial structures in cognition. *Behavioral and Brain Sciences*, *29*, 37–108.

Walles, S., Robins, A., & Knott, A. (under review). A model of serial visual attention and group classification. Manuscript.

Webb, A., Knott, A., & MacAskill, M. (2010). Eye movements during transitive action observation have sequential structure. *Acta Psychologica*, *133*, 51–56.

Werbos, P. J. (2002). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, *78*, 1550–1560.

Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. *1960 IRE WESCON Convention Record*, *4*, 96–104.

Yu, C., & Ballard, D. H. (2007). A unified model of early word learning: Integrating statistical and social cues. *Neurocomput.*, *70*, 2149–2165.