Department of Computer Science,
University of Otago

Technical Report OUCS-2011-03

A connectionist model of language acquisition and sentence generation: Technical appendix

Authors:

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

---

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

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

A connectionist model of language acquisition and sentence generation: Technical appendix

Martin Takac, Lubica Benuskova and Alistair Knott
Dept of Computer Science, University of Otago

November 10, 2011

Abstract

In this report we present technical details of a neural network model of sentence generation, including details of the artificial languages it was trained on, its training regime, and of the performance of the trained network.

1 Introduction

This report provides a detailed description of a neural network model of sentence generation, and of its training and evaluation. The network receives a semantic representation (an **episode representation**) as input, and delivers a sequence of words (a **sentence**) as output. The network is trained on a corpus of training items derived from an artificial language, which maps episode representations onto sentences. Each training item consists of an epsiode, along with the sentence which describes it in the training language. The network is then tested on a set of unseen episodes, for which it must produce an appropriate sentence.

The format of episode representations which provide input to the network is described in Section 2. The architecture of the network and its training are described in Sections 3–5. The way sentence generation happens in the complete model after training is described in Section 6. The artificial languages used to train the network are described in Section 7. The performance of the networks trained on these languages is described in Section 8.

2 Semantic input representations

The network's semantic input representations are assumed to represent concrete episodes, of the kind which can be directly apprehended through sensorimotor experience of the world. Experiencing an episode is assumed to involve a canonical sequence of sensorimotor processes; for justification of this idea, see Knott (2012). Experienced episodes are stored in working memory as prepared sensorimotor sequences, which can be internally replayed:

the process of sentence generation involves replaying a stored sensorimotor sequence in a special mode where sensorimotor signals can have linguistic side-effects.

The complete sequence of signals in Knott's account of a rehearsed reach-to-grasp deictic routine is shown in Table 1. The sequence involves four **sensorimotor operations**,

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	Sensorimotor operation	Reafferent SM 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*	

each of which takes place in a particular sensorimotor context (the **initial context**) and results in an updated sensorimotor context (the **new context**), generating a **reafferent sensorimotor signal** as a side-effect.

Note that the sequence involves a mixture of transient and sustained signals. The sustained signals are part of the static representation in working memory which supports the replay operation. The transient signals are the ones which are evoked when the replay actually occurs. Note also that there are two reafferent sensory reprsentations of the agent at different points during the sequence, and also two reafferent representations of the patient. For detailed motivation of all aspects of this model of reaching-to-grasp, see Knott (2012).

All the episode representations which we will use in our network take the form of sequences with the kind of structure shown in Table 1. They all encode 'transitive episodes', involving an agent, a patient and a transitive action.

3 A core network for learning abstract syntactic rules

The complete sentence generation network 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 Section 5.2.

3.1 The episode rehearsal network

First we present the network which replays sensorimotor sequences, which constitutes the main technical innovation in the model. This network 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.

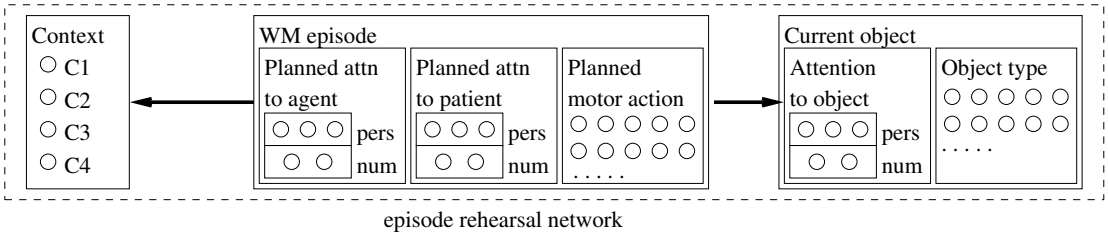

Figure 1: The episode rehearsal system. The working memory (WM) episode comprises planned actions of attention (attn) to an agent and a patient and a planned motor action. Each planned action of attention encodes the person (pers) and number (num) of the attended object using a 1-hot localist scheme where exactly one unit is active at a time. (The pers units represent first, second and third person; the num units represent singular and plural.) The planned motor action uses a similar 1-hot scheme to encode a planned open-class motor action. The current object area holds a transient representation of the currently attended object (which alternates between the agent and the patient). This comprises an attentional action, which represents person and number information as above, and an object type, which uses a 1-hot scheme to encode an open-class object category. The context area holds a representation of the current stage during episode rehearsal (see Table 1), again using a 1-hot scheme.

The episode rehearsal network is shown in Figure 1. 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.) 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 active unit for each 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 2. 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. 2) so that at any time input comes either wholly from the WM episode 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 and 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

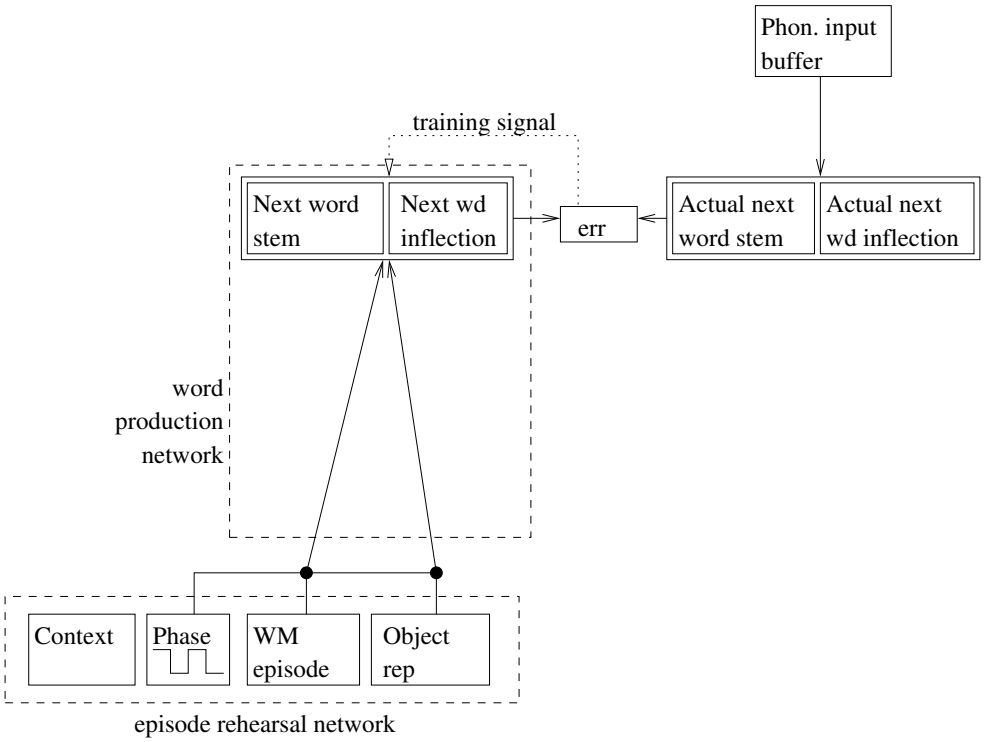

Figure 2: The word production network. This network receives inputs from the episode rehearsal system (shown in Figure 1), and training signals from the phonological (phon.) input buffer. Note that the episode rehearsal system is augmented with a cyclic pattern generator (Phase). During episode rehearsal, 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. The training sentences presented to the system are stored in a phonological input buffer, from where they can be replayed word-by-word. An error term (err) is calculated based on the difference between the next word (wd) predicted by the production network and the actual next word of the training utterance, and this term is used to train the word production network.

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

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 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. (The episodic buffer is implemented in our model by the episode rehearsal system.) 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.[1] 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 and Baddeley, 1990.) However, we argue later in the paper that offline training also has a role in syntactic development.

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

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. As already noted, sentences do not contain wholesale repetition of 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 withhold 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

Table 3: A control policy which produces VSO (verb subject object) word order ('—' and '↓' denote 'withhold' 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*

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 3. 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.[2]

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

[2]As indicated in Figure 3, 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).

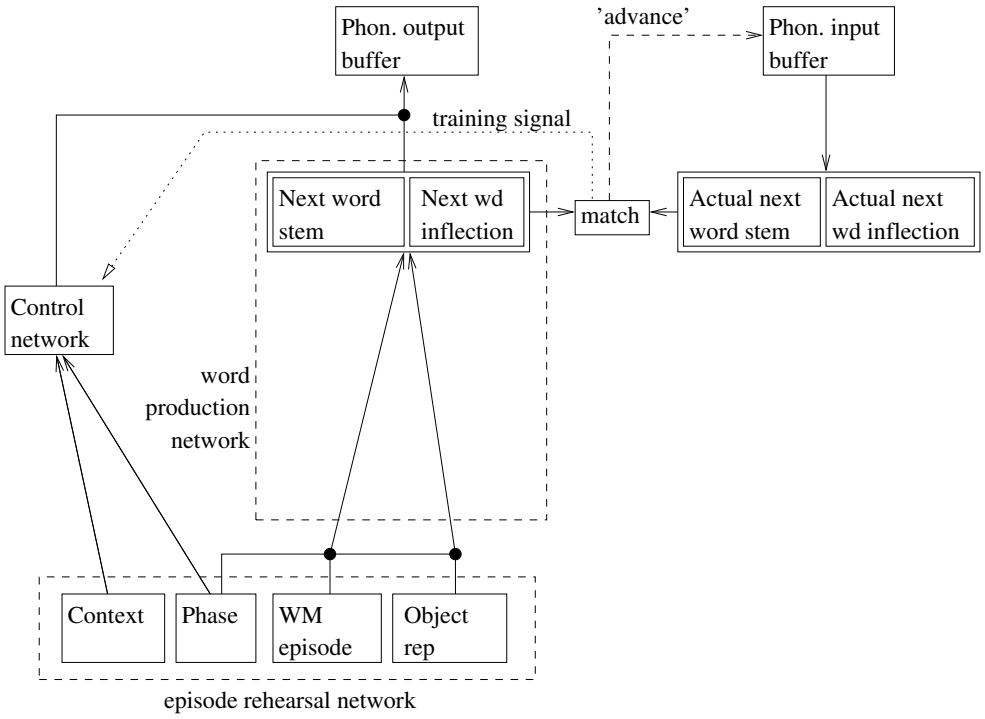

Figure 3: The control network. The control network is a device which learns when to pronounce and when to withold the words evoked by the word production network; thereby learning the basic word order of the particular language. If the output of the word production network matches the next word replayed from the phonological input buffer, the control network is 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 'withhold' 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.

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).

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 'withhold' 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.

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.[3] 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 table 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 'withhold' signal, and the word is retained at the next step. If they are, the control network is trained to give 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,

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

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	—	↓	—	—	↓	—	↓

C3a and C4a, and 'withhold' 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 8.2. 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, advancing 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 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 4, which also shows some of the components of the earlier network which they interact with.

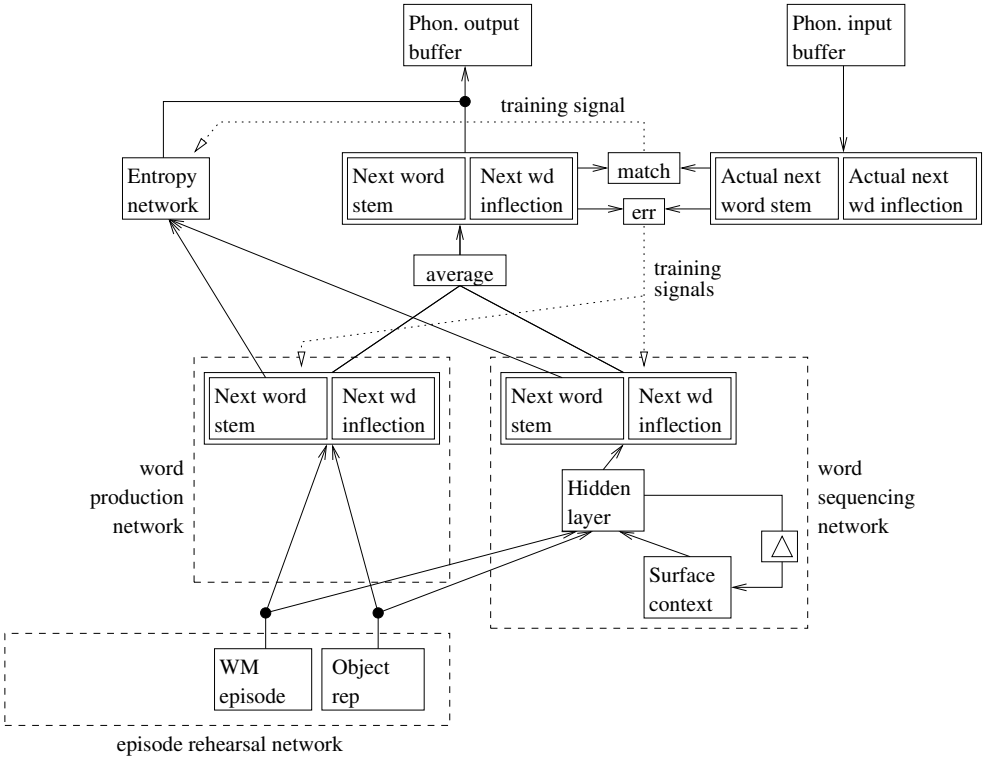

Figure 4: The word sequencing network and its interaction with the word production network. Both networks are trained using the 'actual' next word replayed from the phonological input buffer (err). However, the word sequencing network has one hidden layer with recurrent connections, which enables it to take into account the history of previous inputs. 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). The aggregated output of the two networks is gated by a confidence signal provided by the entropy network. 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.

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 4, 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 using 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. This allows it to produce an 'idiomatic' sequence of words, which collectively express a single semantic input.

The output of the aggregated network depicted in Figure 4 (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.[4]

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 control network to support the generation of sentences containing a mixture of surface and abstract linguistic patterns. However, we also argue that it has an important developmental role, in generating early pre-syntactic multi-word utterances before the control network is fully developed. These roles of the production/sequencing network will be discussed in Section 5.1.

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 the form of this word as output: so as well as learning about sequential patterns of words, it learns a mapping from word meanings to word forms, just like the word-production network. (However, the word-production network still has an important function in its own right, in making predictions about the next word which are based *purely* on its meaning, and not conditioned on the current context.)

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

4.2 The entropy network

The word-production/sequencing network operates in conjunction with another network: the **entropy network** shown at the top left of Figure 4. 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. It learns a simple threshold function, which takes the entropies of the word production and sequencing networks and returns a binary decision: 'pronounce' or 'withhold'.

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 'withhold' 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 Section 5.2.

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

As discussed in Section 4.1, the word sequencing network is able 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 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*.[5] 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 The complete model of word-learning and syntactic development

So far we have 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, as well as the model's training regime.

5.1 Overview: structure and training of the complete network

The structure of the complete model is shown in Figure 5. It combines the word production and sequencing networks described in Section 4 with the episode rehearsal and control networks described in Section 3.

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 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 withhold words.

[5]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*.

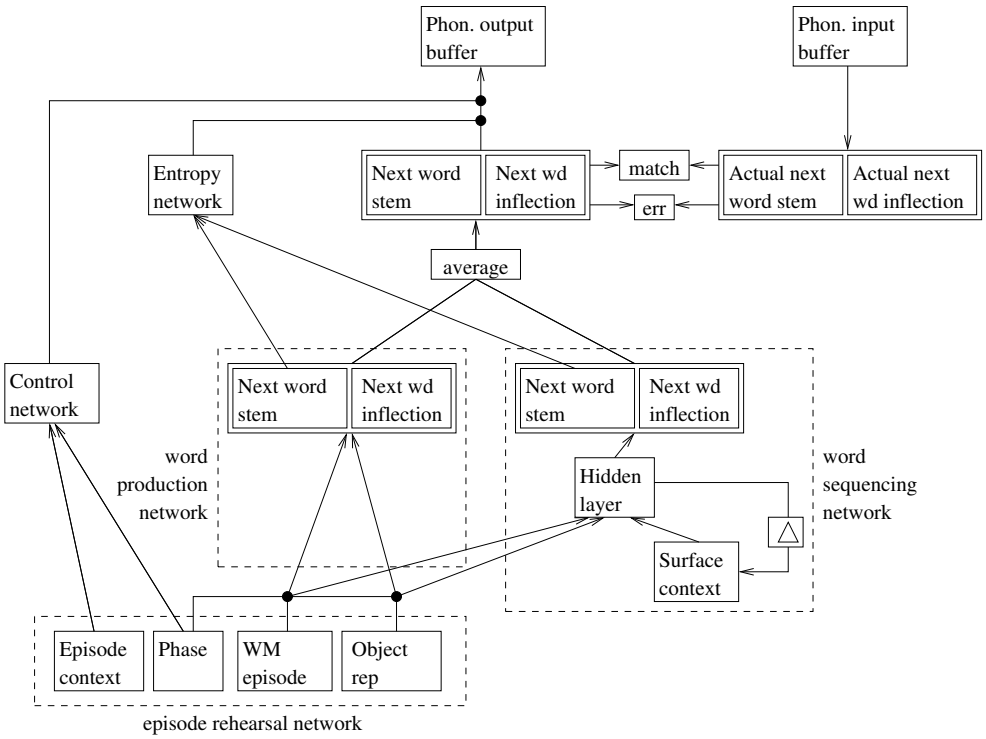

Figure 5: The complete model of language production. In the complete model, generating a sentence involves replaying an episode in the episode-rehearsal system, and at various points during replay, pronouncing one or more words (i.e. dispatching words to the phonological output buffer). A key issue in the combined model is the synchronisation between the episode rehearsal and word sequencing networks. Besides gating overt pronunciation, this coordination is the job of the control network and the entropy network as described in detail in the text.

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 Section 5.2.

This way of training creates a circular dependence: each network's training relies on the others already giving meaningful output. Our intention 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 and episode-rehearsal networks. To be concrete, we assume that learning in the WPSN begins around 12 months, when infants begin learning their first word meanings (Tomasello, 2003), while learning in the control network begins around 18–24 months, when infants first show evidence of using abstract syntactic rules (Hirsh-Pasek and Golinkoff, 1996; Tomasello and Broooks, 1998). The idea that cognitive control strategies mature fairly late in development is well supported (see Novick *et al.*, 2005 for a review), but some simple control strategies start to emerge at around 18 months (Posner and Rothbart, 2000). Our model simulates linguistic development from around 12 months, when the word-learning circuit becomes active. We simulate gradual maturation of the control network by adding a measure of noise to the control signal which is gradually reduced during training.[6] While the control network is delivering random output, it is

[6]Syntactic development would also be affected by the maturation of episodic working memory, but

still possible for the WPSN to learn word-meaning mappings, because there are higher-than-chance correlations between concepts and the words which denote them: this is the principle on which 'cross-situational' models of word learning are founded (see e.g. Siskind, 1996; Yu and Ballard, 2007).

We also simulate gradual maturation of the phonological input buffer. This is a separate developmental process: we assume that an infant begins with an immature input buffer, in which a number of recently-heard words are active in parallel, but gradually transitions to a mature mode in which words are replayed from the buffer one by one.[7] 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 Section 5.2.

5.2 Technical description of the complete network

In this section we will describe the modules of the complete network shown in Figure 5 in more detail, as well as the training and sentence generation algorithms.

5.2.1 Modules of the network

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'[8] signal),

for simplicity's sake we assume that the episode rehearsal system is fully mature from the outset in our simulations.

[7]There are many models of the phonological buffer in which words are represented in parallel; see e.g. Burgess and 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.

[8]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

the other possible word inflections[9] (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 and 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), 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

proceeds to the next episode).

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

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 6) 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 6) and is interpreted as 'let through' signal if greater than 0.3.[10] We simulate gradual maturation of the control network in the way that decisions based on the control network's output are interleaved with a certain proportion of random decisions. In each phase, the decision is random with probability $p(epoch)$, where p linearly decreases from 1 in epoch 0 to 0 in epoch 15, i.e. at the start 100% of decisions are random, while after epoch 15 all decisions are based on the real control network output.

5.2.2 Training and sentence generation algorithms

All training algorithms use the same learning rate (0.1) and zero momentum.[11] Connection weights in all networks are initialised with random values between $(-0.5, 0.5)$, the surface

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

[11] 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.

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), which eliminate the influence of the previous episode. (We use multiple passes to allow the SRN's internal dynamics to settle after the episode boundary; see Cernansky *et al.*, 2007 for a similar operation.)

The model is trained for 30 epochs. (Its generation accuracy on the training set typically reaches its maximum point some time before this, but we continue to train until its accuracy on the text set begins to drop, indicative of overfitting; this can be discerned around epoch 30). 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 6 and 5.1). 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 6 and 7.

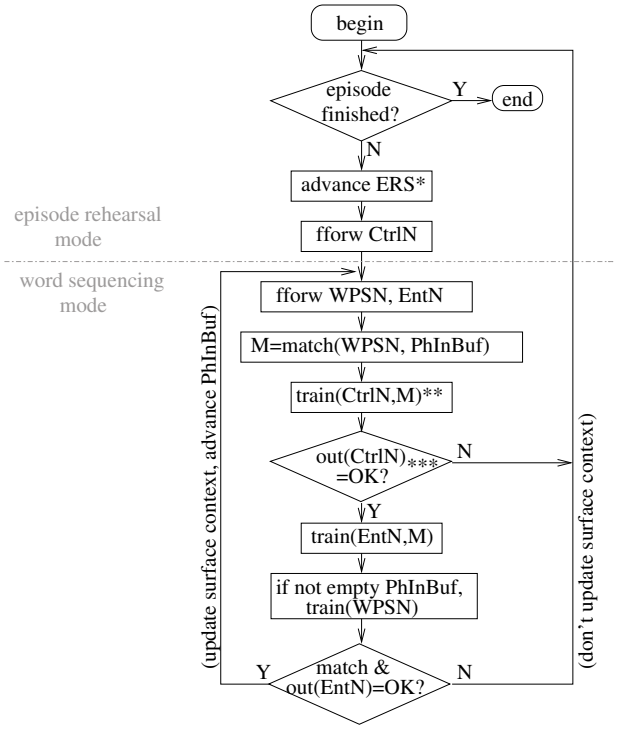

Figure 6: Training algorithm for the complete model. Abbreviations: PhInBuf—the phonological input buffer (other abbreviations are explained in a legend to Fig. 7). The first call of 'advance ERS' (*) puts the network into the C1a context/phase. Control network training (**) is skipped in the first developmental stage (before the control network goes online). Also, the output of the CtrlN (***) is substituted with a random signal in that stage.

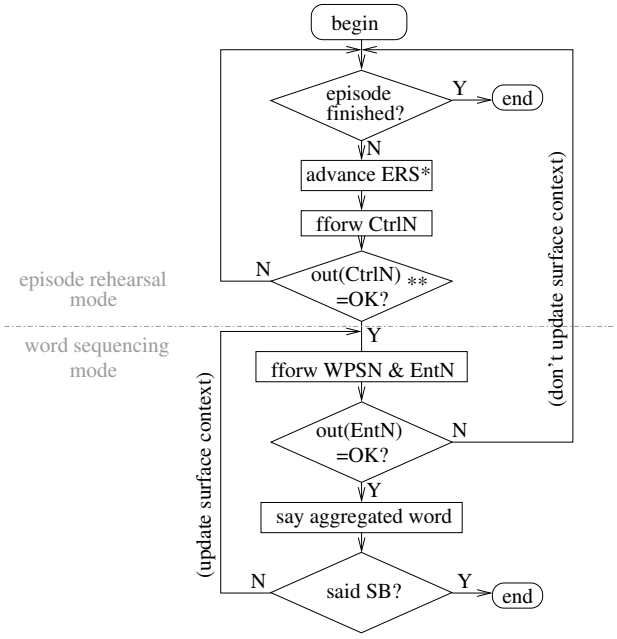

Figure 7: The utterance generation algorithm in the complete model. Abbreviations: ERS—episode rehearsal system, CtrlN—the control network, fforw—feed forward pass, WPSN—the aggregated word production/sequencing network, EntN—the entropy network, SB—sentence boundary signal. The first call of 'advance 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).

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 where a word is to be overtly pronounced.

6 Language generation in the complete network after training

In the full network 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 surface contexts/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 'withhold' a word, to conform to the abstract syntactic word-ordering constraints of the exposure language. There are also occasions when multiple words 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.

When training is complete, 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 network generates a prediction about the next word. If it 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 WPSN attempts to predict another word (from the same sensorimotor signal). Iteration continues in this mode, with a static sensorimotor signal, until the WPSN 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 Section 5.2.

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 iterations 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

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*

WINNIETHEPOOH). From this signal, and the 'start-of-sentence' surface context (c1), the WPSN can confidently predict the word *winnie*. Since the control network has learned to pronounce words in context/phase C1a, this word is produced. Before a new context/phase is established, the WPSN first updates to a new surface context (c2) to reflect the newly produced word, and the WPSN predicts another word from the same sensorimotor signal WTP. In the updated context, it now predicts *the*, again with high confidence, so this word is also produced, and the WPSN update the surface context to c3. In this context, the WPSN network predicts *pooh* with high confidence, and updates to surface context c4. In c4, the WPSN 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 WPSN confidently predicts the word *grabs*. (This word is consistent with the semantic signal, and is also commonly attested following the surface word sequence *Winnie the Pooh*.) Now the WPSN updates to a new surface context and attempts to predict another word, but it cannot: *grabs* is not part of an idiom. So the episode-rehearsal system advances 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 WPSN confidently predicts *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.

In our model, the phonological input buffer's ability to replay utterances accurately matures some time before the control network delivers reliable output. This creates a sequence of three developmental stages. In the first stage (roughly before epoch 3–5), 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 (from epoch 3–5 till 10–15), it can learn something about word sequences, but since the control network is still unreliable, it cannot yet learn abstract syntactic rules. In the final stage (from epoch 15), once the control network has matured, 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.[12]

7 Training languages

The model we have just described has been implemented and tested on several artificial languages. One of these languages, with SVO word order, is described in detail in Section 7.1. The other languages are described in Section 7.2.

7.1 Structure of the training languages

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 (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, 2012). 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*),

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

- 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 and 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, 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 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 (described below). The rules for inflections were as follows:

- All proper names were singular. (Proper names include *mummy*, *daddy*, *grandpa* and *grandma*.)
- The person/number of the verb agreed with that 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).[13] To test the generalisation ability of the model, we only trained it on a small subset of all possible sentences (approx. 3%).

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 6. 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 then 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.*

7.2 Other languages with different word-ordering conventions

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 variant artificial 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.

8 Results

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

[13] A description of how the 'degree of idiomaticity' of utterances is determined is given in Section 8.4.1.

Table 6: 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

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 the control network's output was mixed with a gradually decreasing component of noise which accounted for 100% of decisions at the start of the training (epoch 0) and decreased gradually to 0% at epoch 15 (for details see Section 5.2). 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.

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.

8.1 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 8.[14]

As the figure shows, after an initial peak, active vocabulary size rises steadily and asymptotes around epoch 11. By the end of this epoch, the model is correctly producing all the 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:[15] 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.

[14]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 8.

[15]The entropy network is trained by the 'match' signal which compares the WPSN output and the current content of the phonological input buffer. As this content changes from parallel representation of all words to one word at a time, the match criterion becomes more strict.

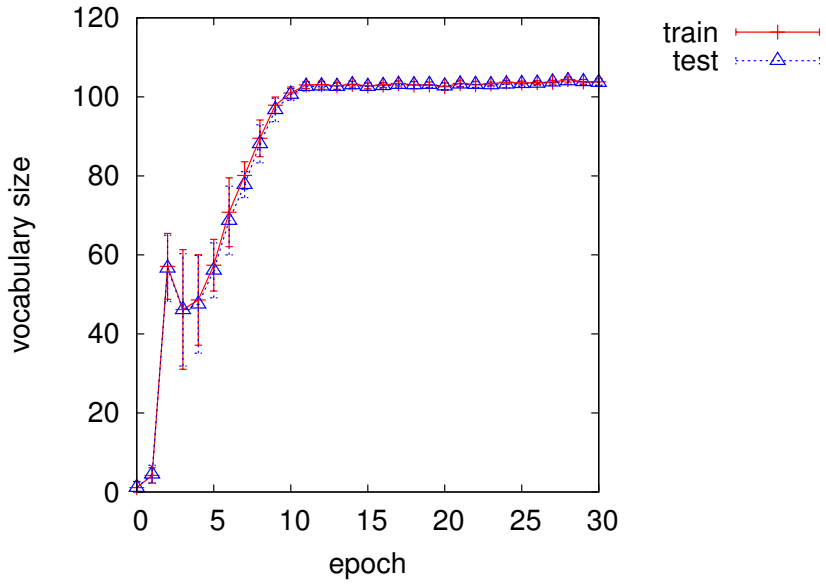

Figure 8: Active vocabulary size after each training epoch. We define the active vocabulary size of the model in a given epoch as the number of word types which were produced correctly at least once during that epoch. The results are averaged over the 10 SVO model subjects.

8.2 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 for each language. The results are shown in Figure 9. For each training language, the same inhibition pattern was learned by each of the 10 model subjects: the learned inhibition patterns for the different languages are shown more concisely in Table 7. 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.

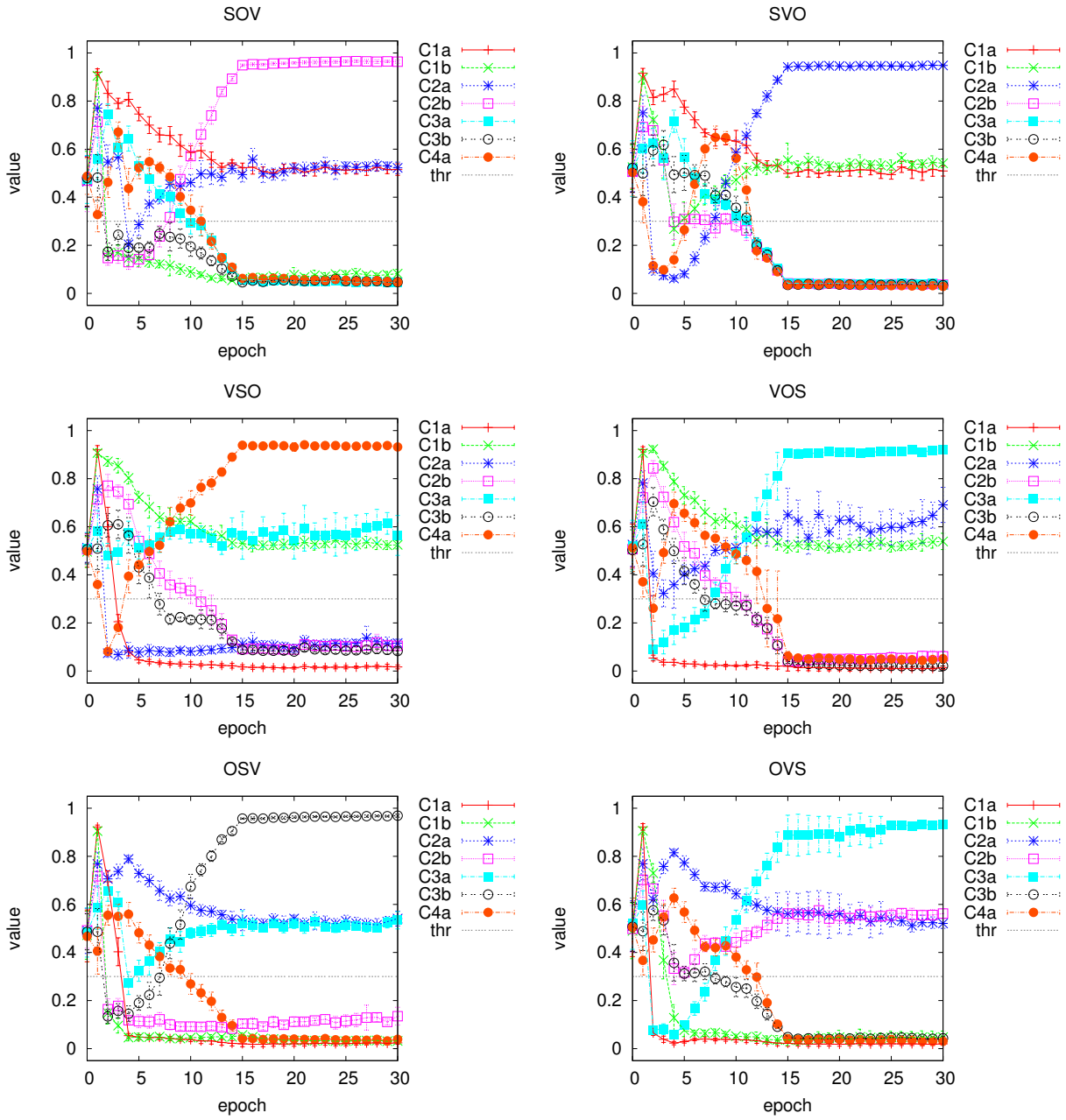

Figure 9: Control network output values for each episode context recorded during training, for model subjects trained on each language (averaged over 10 model subjects in each case). For each training language, the network learns to return above-threshold values for contexts where semantic signals should be pronounced and below-threshold values for those where they should not. (The threshold is set at 0.3.) A summary of the learned inhibition patterns for each language is given in Table 7.

Table 7: 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 'withhold' 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	—	—	↓	↓	↓	—	—

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 8.4, when we evaluate the system's ability to generate full sentences.

8.3 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 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.[16] The graph in Figure 10 shows the proportion of the words generated in each epoch

[16]Using an incorrect pronoun (e.g. *you* instead of *they*) also counts as morphologically incorrect on this

which were morphologically incorrect, averaged over 10 SVO model subjects.

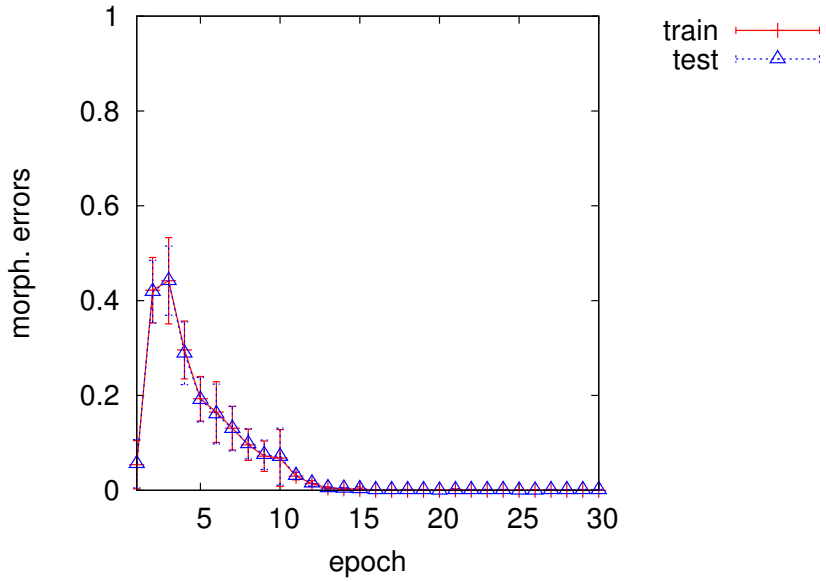

Figure 10: Proportion of words pronounced with morphological errors in sentences generated for meanings from training and test sets after each training epoch. Results are averaged over 10 SVO model subjects.

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 that of the control network. The output of the control network is still dominated by noise at epoch 5. The network clearly learns a good deal about morphology without help from the control network. But its performance is clearly aided by learning in the control network.

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

8.4 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

metric, because the mistake relates solely to person/number information.

utterances it produces, evaluated either in relation to its training set of meaning-utterance pairs, or to an independent test set.

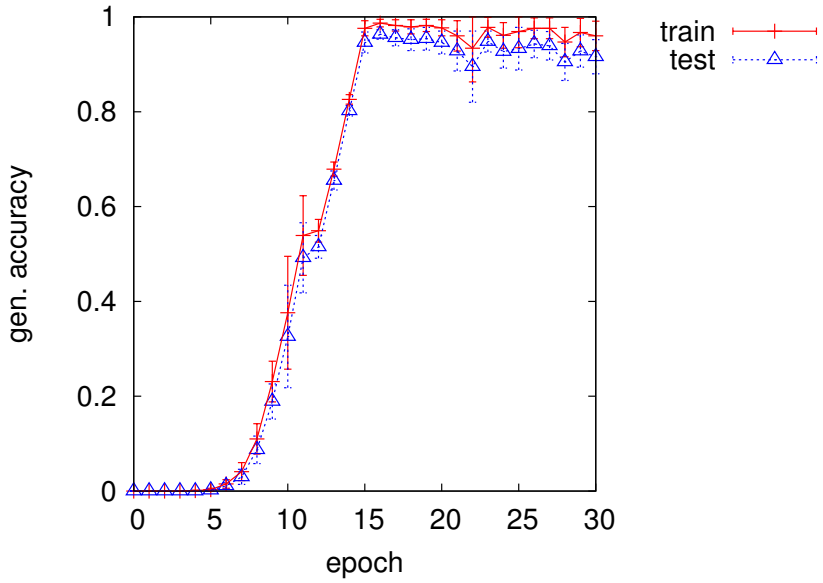

Figure 11: The generation accuracy—relative number of correct sentences generated after each training epoch for meanings from the training set and an independent test set, averaged over 10 model subjects trained on samples of the SVO language.

Figure 11 shows the generation accuracy of 10 model subjects trained on the SVO language. As is very obvious, the control network, which accounts for more than 50% of decisions since epoch 8, has a dramatic impact on generation accuracy: before it comes online, the model produces almost no correct sentences, and by the time it is fully trained, accuracy has improved to close to 100%. Before the control network has learned abstract constituent-ordering rules, the model is not able to produce *fully* correct sentences. We should stress that during this time the model is not 'silent': it is learning vocabulary and surface word regularities, and it produces a range of single-word and multi-word utterances, which often convey a good deal of the message to be expressed. We will analyse these pre-syntactic utterances in more detail in Section 8.4.2.

One interesting feature of the model's learning is that accuracy does not increase monotonically: it undergoes small fluctuations. These are mainly caused by adaptations in the entropy network, which has to modify its function as the other networks learn new material. 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 18.8 training epochs (SD=1.47), and at this point achieved 99.4% (SD=0.2%) generation accuracy on the training set and 97.7% (SD=0.4%) 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.

To further test our complete model's ability to generalise away from the training sentences, we created a new set of test sentences for each model SVO subject, each comprising 100 sentences which were not only unseen, but contained unseen pairings of actions and patients, and accordingly, unseen transitions between verbs and objects. 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*. We let the 10 fully trained SVO model subjects described in this section generate sentences for meanings in these new test sets. Their average generation accuracy was 84.1% (SD=3.8%). While this is somewhat lower than their accuracy on 'standard' unseen sentences, it still provides evidence that the rules our network learns generalise quite successfully away from token words.

8.4.1 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 12. The model performs well for all sentence types. Its high accuracy on discontinuous idioms is especially significant, given that these constructions only feature in about 6.4% of the training sentences.[17] The average generation accuracy on previously unseen episodes (test set) was 98.1% (SD=0.4%) for regular sentences, 97.2% (SD=0.7%) for continuous idiomatic sentences and 93.1% (SD=2.9%) 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

[17] 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.

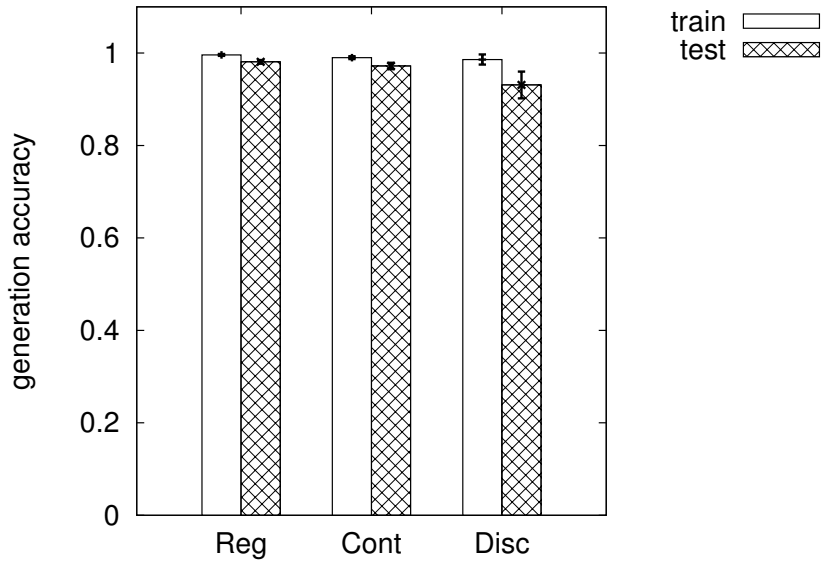

Figure 12: Generation accuracy of fully trained model subjects by idiomaticity type of generated sentences. Reg—regular (non-idiomatic) sentences, Cont—sentences with continuous NP idioms, Disc—sentences with discontinuous VP idioms.

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 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*.

8.4.2 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.

Figure 13 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 starts to significantly participate in decision making, one-word and two-

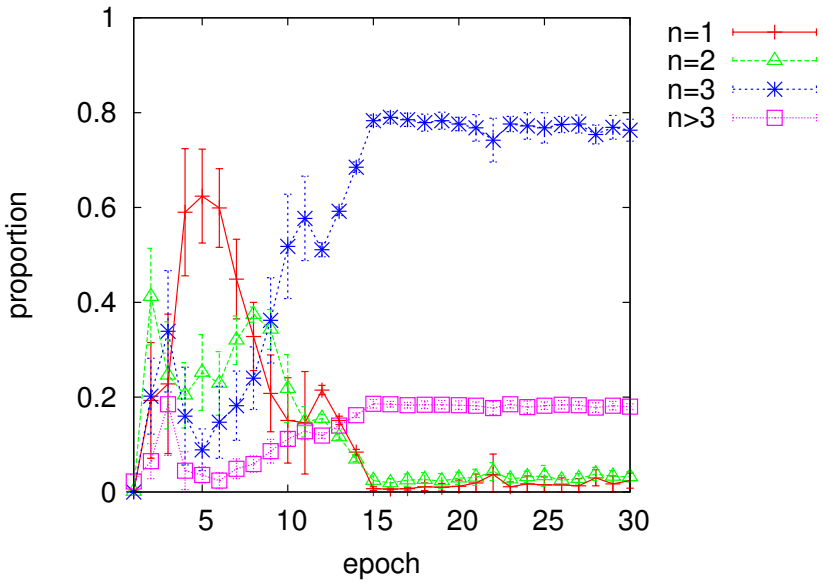

Figure 13: Proportion of sentences of the length n among sentences generated for an independent test set after each training epoch. Results are averaged over 10 SVO model subjects.

word utterances predominate. As the control network starts to take over after epoch 7, the model produces more utterances containing three or more words (and these utterances become more likely to be correct, as already shown in Figure 11).

In early development, 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*). And sometimes they result from the (incorrect) repetition of a word.

Interestingly, 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. As already noted, children's early uses of grammatical constructions are typically item-based (see e.g. Pine and Lieven, 1997; Lieven *et al.*, 1997; Tomasello, 2003). For instance, a child's first utterances productively combining a transitive verb with an object noun typically involve particular verbs, rather than all the transitive verbs in the child's vocabulary (Lieven *et al.*, 1997). Something similar occurs in our system. We analysed the set of verb-noun constructions generated by the system at each epoch of training in the SVO language.[18] We investigated whether there was a bias towards particular verbs

[18]A verb-noun construction was defined as a pair of consecutive words correctly expressing either the

in verb-noun constructions by finding the verb which appeared most frequently in these constructions and comparing the frequency of the constructions featuring this verb with the average frequency of constructions featuring other verbs. Results are shown in Figure 14. As the figure shows, at around epochs 5–6 the most frequent verb combines with nouns

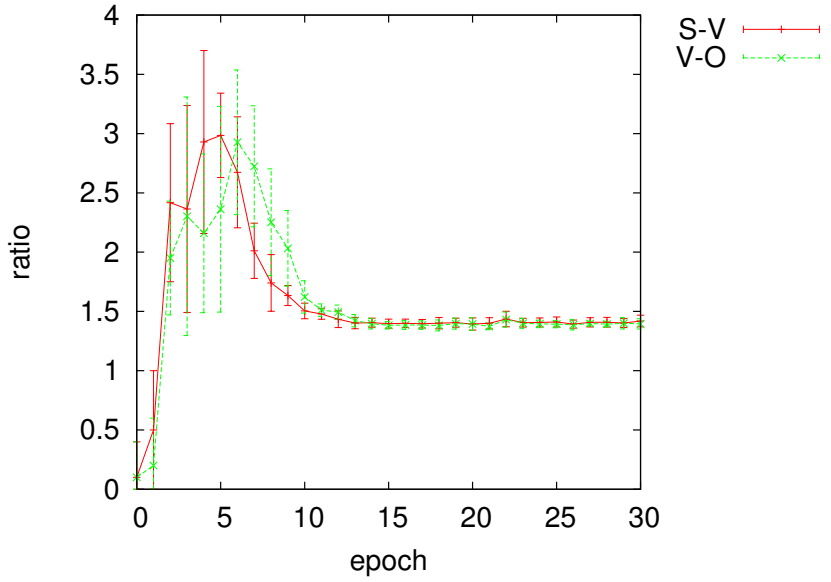

Figure 14: Ratio of the frequency of verb-noun constructions featuring the most common verb to the average frequency of verb-noun constructions featuring other verbs, recorded on the test set for the SVO language, at each epoch of training. (Results averaged over all 10 model subjects.)

around 3 times more often than an average verb. This ratio decreases to around 1.5 after epoch 10, and it stays at this lower level for the remainder of training. The asymptote of 1.5 probably reflects sampling biases towards particular concept combinations in the training set. The fact that the ratio is much larger earlier in training is a clear indication of the item-specific character of the network's early verb-noun constructions.

The system's item-specific constructions early in training reflect the influence of the word-sequencing network on generation before the control network is fully mature. When the control network is mature, it supports the generation of arbitrary verb-noun combinations, even entirely new combinations of action and patient, as shown in Section 8.4. But before it is mature, the task of generating these combinations falls partly to the word-sequencing network—which learns patterns which make reference to particular words. As the control network matures, it takes over responsibility for generating these patterns, and the sequencing network retains responsibility only for idiomatic constructions.

agent and action, or the action and the patient of the episode to be communicated, in the order required by the SVO language: examples would be *teddy tickle* or *tickle teddy*. (Morphology was ignored, and we excluded episodes where the same object was the agent and the patient from the analysis.)

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% 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.*[19] 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.

References

Baddeley, A. (2000). The episodic buffer: A new component of working memory? *Trends in Cognitive Sciences*, **4**(11), 417–423.

Browman, C. and Goldstein, L. (1995). Dynamics and articulatory phonology. In R. Port and T. van Gelder, editors, *Mind as Motion: Explorations in the Dynamics of Cognition*, pages 175–193. MIT Press, Cambridge, MA.

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

Cernansky, M., Makula, M., and 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**(2), 236–244.

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

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

Fenson, L., Dale, P., Reznick, J. S., Thal, D., Bates, E., Hartung, J., Pethick, S., and Reilly, J. (1994). Variability in early communicative development. *Monographs of the Society for Research in Child Development*, **59**(5), i–185.

[19]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.

Gathercole, S. and 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.

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

Hirsh-Pasek, K. and Golinkoff, R. (1996). *The Origins of Grammar: Evidence from Early Language Comprehension*. MIT Press, Cambidge, MA.

Knott, A. (2012). *Sensorimotor cognition and natural language syntax*. MIT Press, Cambridge, MA. Currently available at www.cs.otago.ac.nz/staffpriv/alik/publications.html.

Lieven, E., Pine, J., and Baldwin, G. (1997). Lexically-based learning and early grammatical development. *Journal of Child Language*, **24**, 187–219.

Ma, W. J., Beck, J. M., Latham, P. E., and Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nat Neurosci*, **9**(11), 1432–8.

Novick, J., Trueswell, J., and Thomson-Schill, S. (2005). Cognitive control and parsing: Reexamining the role of Broca's area in sentence comprehension. *Cognitive, Affective and Behavioural Neuroscience*, **5**(3), 263–281.

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

Posner, M. and Rothbart, M. (2000). Developing mechanisms of self-regulation. *Development and Psychopathology*, **12**, 427–441.

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

Shallice, T., Rumiati, R., and Zadini, A. (2000). The selective impairment of the phonological output buffer. *Cognitive Neuropsychology*, **17**(6), 517–546.

Siskind, J. (1996). A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, **61**(1–2), 39–91.

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

Tomasello, M. and Brooks, P. (1998). Young children's earliest transitive and intransitive constructions. *Cognitive Linguistics*, **9**, 379–395.

Werbos, P. J. (2002). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, **78**(10), 1550–1560.

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

Yu, C. and Ballard, D. H. (2007). A unified model of early word learning: Integrating statistical and social cues. *Neurocomput.*, **70**(13-15), 2149–2165.