COSC421: Neural Models of Language

Lecture 12: Neural network models of sentence syntax

Alistair Knott

Dept. of Computer Science, University of Otago

Recap

Lectures 6-7: Language learning networks in the brain

- Phonological networks: auditory cortex $\leftrightarrow$ STS $\leftrightarrow$ IP $\leftrightarrow$ PMC
- Word-meaning networks: temporal cortex, premotor cortex, PFC

Lectures 8-9: Chomsky's Minimalist model of syntax

- No account of sentence processing, surface patterns in language, learning from actual data...

Lecture 10: Empiricist models of syntax

- Syntactic development through item-specific constructions
- Networks which learn patterns in surface language (SRNs)

Lecture 11: A SM interpretation of Minimalist LF

Recap

The SM interpretation of LF:

- The LF of a concrete sentence is a description of the sequence of SM operations involved in experiencing the episode it describes.

Recap

The SM interpretation of LF:

- The LF of a concrete sentence is a description of the sequence of SM operations involved in experiencing the episode it describes. (As replayed from working memory.)

Recap

The SM interpretation of LF:

- The LF of a concrete sentence is a description of the sequence of SM operations involved in experiencing the episode it describes. (As replayed from working memory.)

The Minimalist conception of LF has nothing to do with sentence processing. But the SM conception suggests an interesting account of sentence processing:

Recap

The SM interpretation of LF:

- The LF of a concrete sentence is a description of the sequence of SM operations involved in experiencing the episode it describes. (As replayed from working memory.)

The Minimalist conception of LF has nothing to do with sentence processing. But the SM conception suggests an interesting account of sentence processing:

- When a speaker produces a sentence, s/he is replaying an episode-denoting SM sequence *in a special mode*, where SM signals can evoke phonological side-effects.

Outline of today's lecture

1. Neural network representations of sentence meanings
2. Neural network models of syntactic processing

Outline of today's lecture

1. Neural network representations of sentence meanings

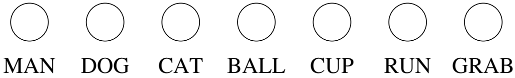
2. Neural network models of syntactic processing

Neural network representations of sentence meanings

How do you represent the meaning of a sentence in a pattern of neural activity?

- E.g. how do you represent our example episode where a man grabs a cup?

Simplest idea: just activate semantic representations MAN, GRAB and CUP simultaneously.

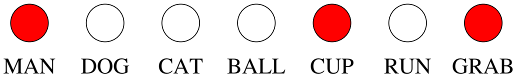


Neural network representations of sentence meanings

How do you represent the meaning of a sentence in a pattern of neural activity?

- E.g. how do you represent our example episode where a man grabs a cup?

Simplest idea: just activate semantic representations MAN, GRAB and CUP simultaneously.

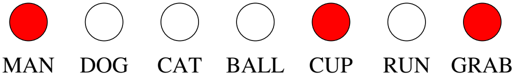


Neural network representations of sentence meanings

How do you represent the meaning of a sentence in a pattern of neural activity?

- E.g. how do you represent our example episode where a man grabs a cup?

Simplest idea: just activate semantic representations MAN, GRAB and CUP simultaneously.



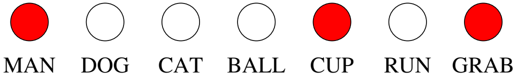
But this doesn't tell us who the agent and patient are.

Neural network representations of sentence meanings

How do you represent the meaning of a sentence in a pattern of neural activity?

- E.g. how do you represent our example episode where a man grabs a cup?

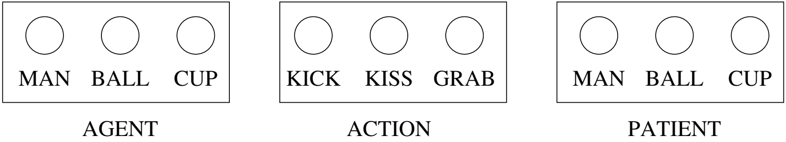
Simplest idea: just activate semantic representations MAN, GRAB and CUP simultaneously.



But this doesn't tell us who the agent and patient are.
The representation scheme needs to **bind** entity reps to thematic roles like AGENT, PATIENT.

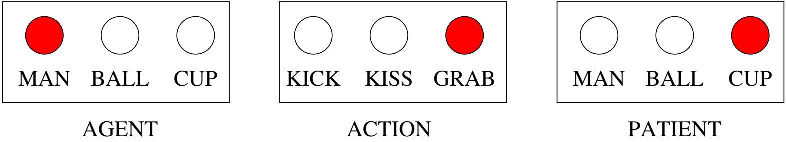
Binding by space

One scheme is to have multiple representations of entities—one for each thematic role.



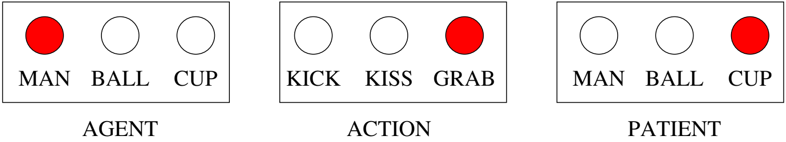
Binding by space

One scheme is to have multiple representations of entities—one for each thematic role.



Binding by space

One scheme is to have multiple representations of entities—one for each thematic role.



But:

- This scheme is combinatorially expensive.
- More importantly: it doesn't allow *generalisation* from one role to another.
  (There's no relationship between MAN-AGENT and MAN-PATIENT.)

Systematicity

After seeing a word in one role during training, a sentence-processing network should immediately be able to use it in another role.

Training:	*man grab cup* *man kick ball* *man kick cup*

Systematicity

After seeing a word in one role during training, a sentence-processing network should immediately be able to use it in another role.

Training:	*man grab cup* *man kick ball* *man kick cup*
Testing:	

Systematicity

After seeing a word in one role during training, a sentence-processing network should immediately be able to use it in another role.

Training:	*man grab cup* *man kick ball* *man kick cup*
Testing:	*man kick* **man**

Systematicity

After seeing a word in one role during training, a sentence-processing network should immediately be able to use it in another role.

Training:	*man grab cup* *man kick ball* *man kick cup*
Testing:	*man kick* **man**

- It should learn the meaning of words independently of the roles they play in a sentence.

Systematicity

After seeing a word in one role during training, a sentence-processing network should immediately be able to use it in another role.

Training:	*man grab cup* *man kick ball* *man kick cup*
Testing:	*man kick* ***man***

- It should learn the meaning of words independently of the roles they play in a sentence.
- It should learn rules about the positions of AGENT, PATIENT in a sentence which are independent of particular words.

Systematicity

Symbolic grammars have perfect systematicity.

Here's a rule about the positions of AGENT, PATIENT and ACTION which abstracts away from words:

$$\boxed{\mathrm{S}(\textsc{ag}{=}\mathrm{X1},\ \textsc{act}{=}\mathrm{X2},\ \textsc{pat}{=}\mathrm{X3}) \rightarrow \mathrm{N}(\mathrm{X1}),\ \mathrm{VT}(\mathrm{X2}),\ \mathrm{N}(\mathrm{X3})}$$

(The arguments carry semantic representations.)

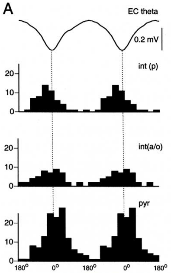
Here are representations of words which abstract away from the roles they can play:

N(MAN)	→	*man*
N(CUP)	→	*cup*
VT(GRAB)	→	*grab*

Binding by synchrony

Background for the **binding-by-synchrony** scheme:

- At a macro-level, the brain's activity is *cyclic*: different periodic signals can be detected at different times, but a strong one is the **theta rhythm** (e.g. in the hippocampus).
- The firing of individual neurons is often time-locked to a particular phase in one of these cycles. Here are some examples:

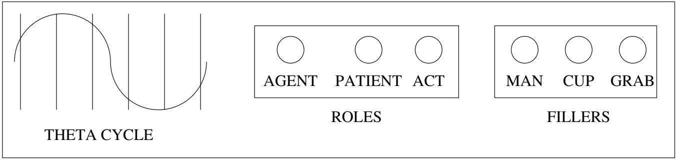


Buszaki (2002)

Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

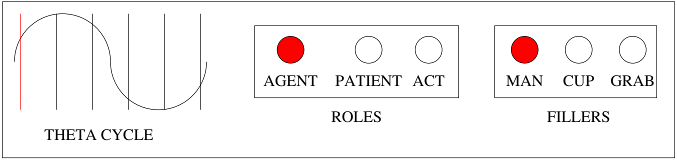
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

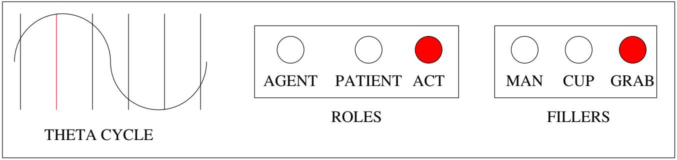
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

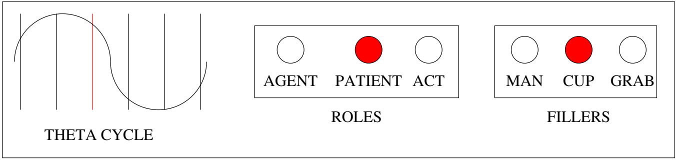
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

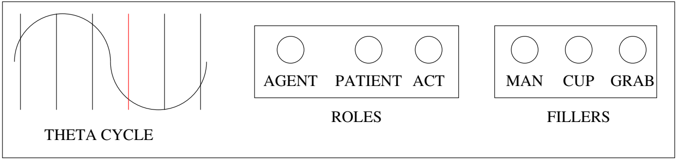
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

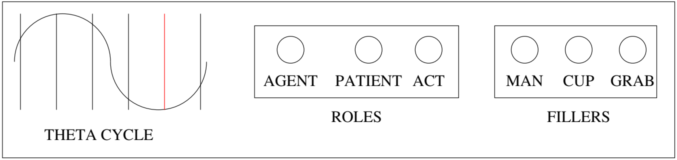
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

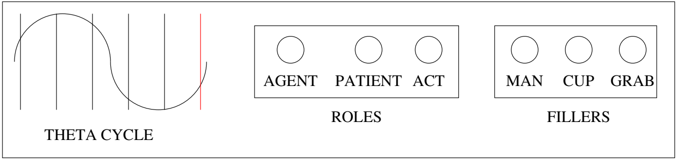
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

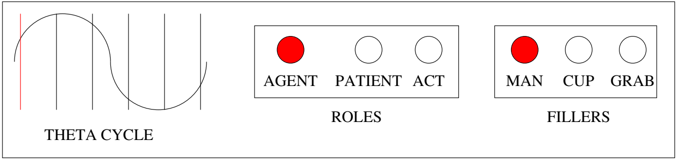
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

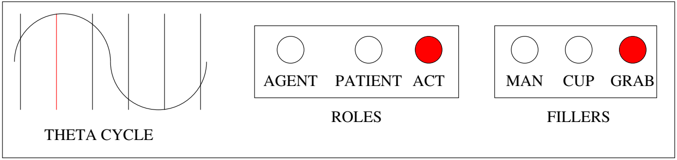
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

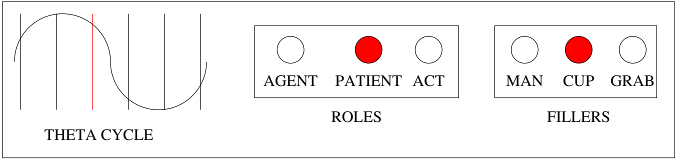
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

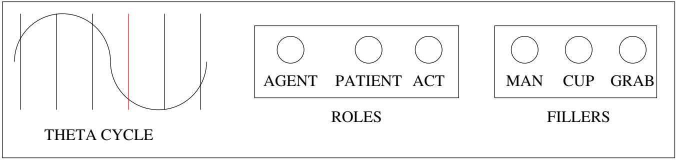
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

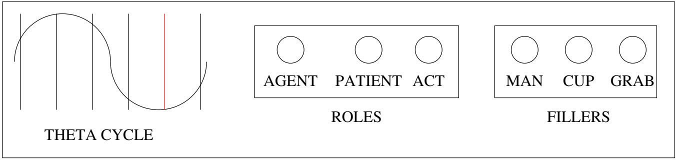
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

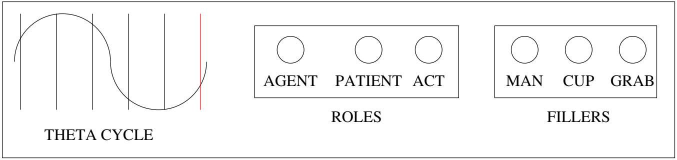
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

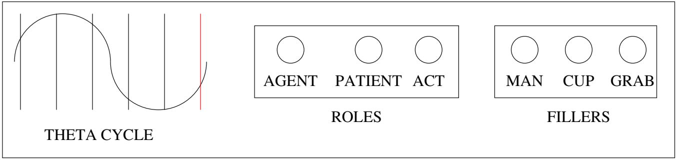
- The idea: role and filler units which fire at the same phase are bound together.



Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

- The idea: role and filler units which fire at the same phase are bound together.

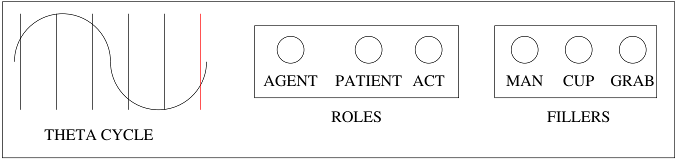


This scheme achieves good systematicity.

Binding by synchrony

In the binding-by synchrony model, some cell assemblies represent roles (AGENT, PATIENT, ACTION etc), and others represent concepts that can fill these roles.

- The idea: role and filler units which fire at the same phase are bound together.

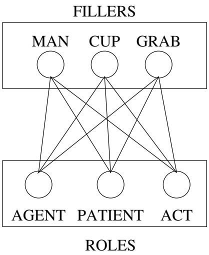


This scheme achieves good systematicity.
But: we're not sure whether this kind of binding happens in the brain.

Binding by connection

The **binding-by-connection** scheme (Chang, 2002) also features separate units for roles and fillers.

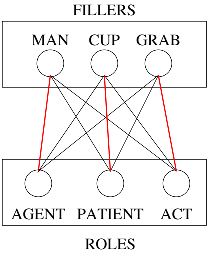
- But here, binding is implemented by short-term connections between role and filler units.



Binding by connection

The **binding-by-connection** scheme (Chang, 2002) also features separate units for roles and fillers.

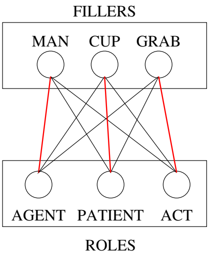
- But here, binding is implemented by short-term connections between role and filler units.



Binding by connection

The **binding-by-connection** scheme (Chang, 2002) also features separate units for roles and fillers.

- But here, binding is implemented by short-term connections between role and filler units.

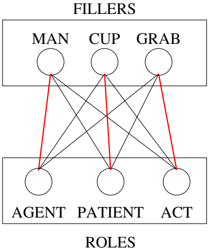


This scheme can also achieve systematicity.

Binding by connection

The **binding-by-connection** scheme (Chang, 2002) also features separate units for roles and fillers.

- But here, binding is implemented by short-term connections between role and filler units.



This scheme can also achieve systematicity.
Again there's some computational complexity.

Binding by serial position

In my SM model, experiencing the cup-grabbing episode involves a canonical sequence of SM operations and their sensory consequences.

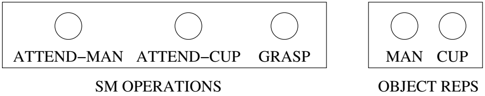
Context	C1	C2	C3	C4
SM operation	ATTEND-MAN	ATTEND-CUP	GRASP	
Sens. conseq	MAN	CUP	MAN	CUP

Binding by serial position

In my SM model, experiencing the cup-grabbing episode involves a canonical sequence of SM operations and their sensory consequences.

Context	C1	C2	C3	C4
SM operation	ATTEND-MAN	ATTEND-CUP	GRASP	
Sens. conseq	MAN	CUP	MAN	CUP

This suggests another scheme, where thematic roles are simply associated with serial positions in a canonical sequence.

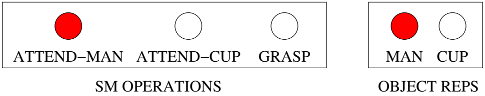


Binding by serial position

In my SM model, experiencing the cup-grabbing episode involves a canonical sequence of SM operations and their sensory consequences.

Context	C1	C2	C3	C4
SM operation	ATTEND-MAN	ATTEND-CUP	GRASP	
Sens. conseq	MAN	CUP	MAN	CUP

This suggests another scheme, where thematic roles are simply associated with serial positions in a canonical sequence.

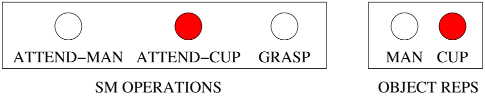


Binding by serial position

In my SM model, experiencing the cup-grabbing episode involves a canonical sequence of SM operations and their sensory consequences.

Context	C1	C2	C3	C4
SM operation	ATTEND-MAN	ATTEND-CUP	GRASP	
Sens. conseq	MAN	CUP	MAN	CUP

This suggests another scheme, where thematic roles are simply associated with serial positions in a canonical sequence.

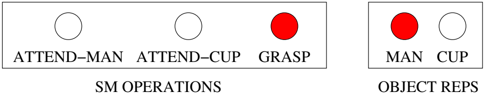


Binding by serial position

In my SM model, experiencing the cup-grabbing episode involves a canonical sequence of SM operations and their sensory consequences.

Context	C1	C2	C3	C4
SM operation	ATTEND-MAN	ATTEND-CUP	GRASP	
Sens. conseq	MAN	CUP	MAN	CUP

This suggests another scheme, where thematic roles are simply associated with serial positions in a canonical sequence.

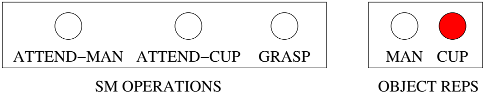


Binding by serial position

In my SM model, experiencing the cup-grabbing episode involves a canonical sequence of SM operations and their sensory consequences.

Context	C1	C2	C3	C4
SM operation	ATTEND-MAN	ATTEND-CUP	GRASP	
Sens. conseq	MAN	CUP	MAN	CUP

This suggests another scheme, where thematic roles are simply associated with serial positions in a canonical sequence.



Representing the semantics of nested sentences

A problem for all the models discussed so far are *nested sentences*.
For example:

> *The man* **who chased the dog** *grabbed the cup*

In these cases, there are *multiple copies* of thematic roles like AGENT, PATIENT, ACTION.

There are lots of ways of addressing this problem.
I won't look at any of them :-)

Outline of today's lecture

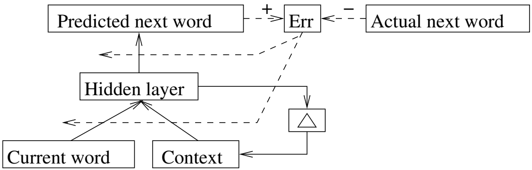
1. Neural network representations of sentence meanings

2. Neural network models of syntactic processing

Recap: SRNs for sentence processing

Recall from Lecture 10, and Assignment 4...

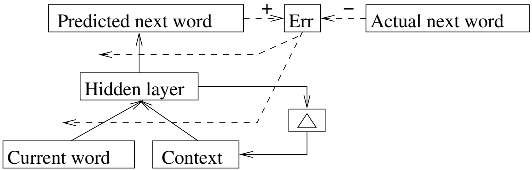
- A SRN can be set up to take a sequence of words as input, and trained to predict the next word.
- It thereby learns a simple model of syntax.
  (And also a neat taxonomy of syntactic/semantic word categories.)



Recap: SRNs for sentence processing

Recall from Lecture 10, and Assignment 4…

- A SRN can be set up to take a sequence of words as input, and trained to predict the next word.
- It thereby learns a simple model of syntax.
  (And also a neat taxonomy of syntactic/semantic word categories.)



That's very useful…
However, if we want to model the mapping between syntax and semantics, we need to extend the model.

Extending a SRN for sentence interpretation

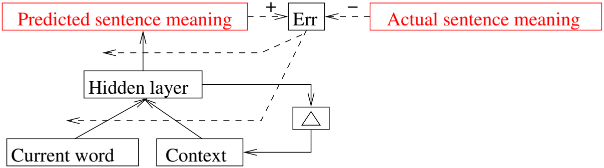
We can also train a SRN to map word sequences onto episode representations.

- Pick your favourite episode representation scheme.
- Create a set of training items, each of which pairs a word sequence with an episode representation.
- For each training item, present the sequence of words to the SRN as input, and the episode representation *as a static output*.

Extending a SRN for sentence interpretation

We can also train a SRN to map word sequences onto episode representations.

- Pick your favourite episode representation scheme.
- Create a set of training items, each of which pairs a word sequence with an episode representation.
- For each training item, present the sequence of words to the SRN as input, and the episode representation *as a static output*.



SRNs and incremental sentence interpretation

After training, this SRN will interpret input sentences *incrementally*.

Assume a simple binding-by-space representation of episodes.
As soon as the first word of a sentence is produced, the network will predict a *distribution* of whole episode representations.

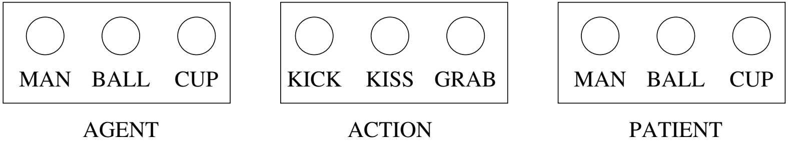
Say the first word is *man*.

SRNs and incremental sentence interpretation

After training, this SRN will interpret input sentences *incrementally*.

Assume a simple binding-by-space representation of episodes.
As soon as the first word of a sentence is produced, the network will predict a *distribution* of whole episode representations.

Say the first word is *man*.

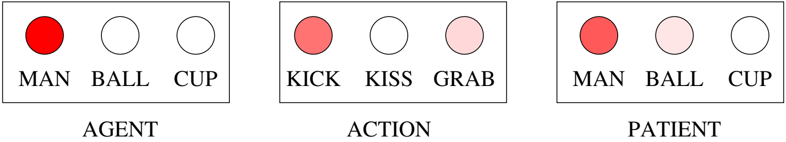


SRNs and incremental sentence interpretation

After training, this SRN will interpret input sentences *incrementally*.

Assume a simple binding-by-space representation of episodes.
As soon as the first word of a sentence is produced, the network will predict a *distribution* of whole episode representations.

Say the first word is *man*.

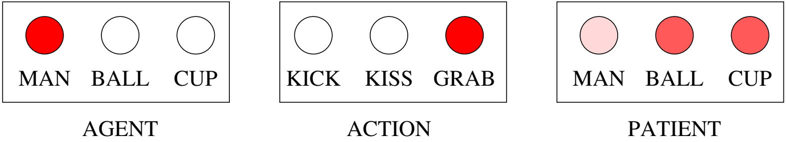


SRNs and incremental sentence interpretation

After training, this SRN will interpret input sentences *incrementally*.

Assume a simple binding-by-space representation of episodes.
As soon as the first word of a sentence is produced, the network will predict a *distribution* of whole episode representations.

Say the first word is *man*. And the next word is *grab*. . .



Extending a SRN for sentence generation

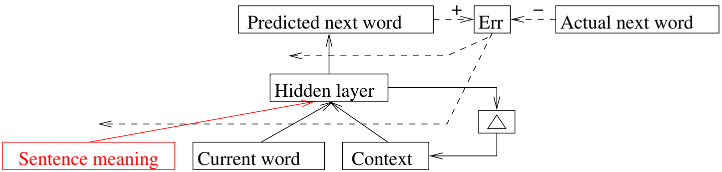
You can also convert a word-sequencing SRN into a sentence generator.

- You use the same sort of training items.
- But now you present an episode rep as an additional *input* to the network, and train it to predict a sequence of words.

Extending a SRN for sentence generation

You can also convert a word-sequencing SRN into a sentence generator.

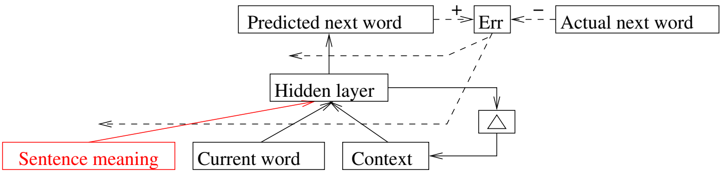
- You use the same sort of training items.
- But now you present an episode rep as an additional *input* to the network, and train it to predict a sequence of words.



Extending a SRN for sentence generation

You can also convert a word-sequencing SRN into a sentence generator.

- You use the same sort of training items.
- But now you present an episode rep as an additional *input* to the network, and train it to predict a sequence of words.



N.B. The current word and context enforce *syntactic* constraints on the output sentence, while the sentence meaning enforces *semantic* constraints.

SRNs and systematicity

SRNs are good at learning sequential patterns of words, as we've seen. But can they produce (or interpret) patterns of words they've never seen before?

Say we use a simple static episode representation scheme (e.g. binding by space).

- For a generating SRN: you're asking it to produce a word sequence it's never produced before.
- For an interpreting SRN: you're giving it a word sequence it's never been given before.

Strong and weak systematicity

How well an SRN can handle new sequences depends on how new they are.

Say the network has seen the following sentences:

man grab cup
woman grab spoon

And we ask it to generate

woman grab cup

This is a test of **weak systematicity**: the ability to deal with new *combinations* of words appearing in *familiar positions*.

An SRN has some ability to do that. (Provided you don't train it too much.)

Strong and weak systematicity

Say the network has seen the following sentences:

man grab cup
woman grab spoon

And we ask it to generate

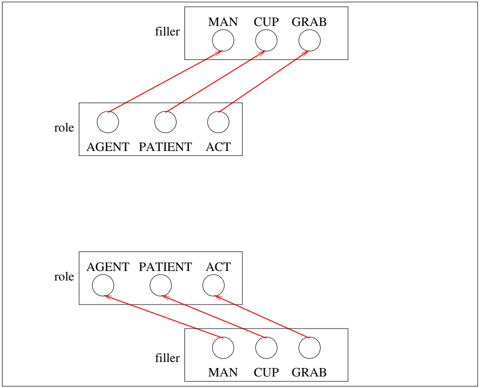
man grab woman

This is a test of **strong systematicity**: the ability to deal with words appearing in *new positions*.

An ordinary SRN using a simple semantic scheme like binding-by-space just can't do that.

Chang's sentence-generating SRN

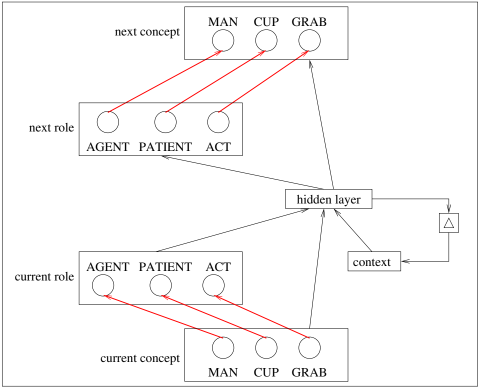
Chang (2002) modified an SRN to use episode reps encoded with a binding-by-connection scheme.



Chang's sentence-generating SRN

Chang (2002) modified an SRN to use episode reps encoded with a binding-by-connection scheme.

The SRN learns to sequence *thematic roles* as well as word meanings.



Chang's model

Chang's model is very empiricist:

- It learns syntactic structures from scratch.
- It learns a mixture of surface and abstract syntactic rules.

In the last part of the lecture, I'll introduce a sentence generation network which implements a more nativist model of syntax, based on my SM interpretation of Minimalism.

A nativist model of syntactic processing

Here's the SM model of episode representations, to start with:

- When we 'entertain the meaning' of a concrete sentence, we internally rehearse a SM sequence stored in WM.

We need to give a model of how children learn to generate *surface word sequences* from these SM replay operations.
I.e. of **how children learn the mapping from LF to PF**.

- The model should ideally include an account of learned surface patterns in language. (Which is problematic for the traditional Minimalist model.)

Deciding when to read out overt words

Here's the sequence of SM signals again:

Sustained signals	Transient signals		
Planned action signals	Context signals	Action signals	Reafferent signals
$plan_{attend\_agent / attend\_cup / grasp}$ ↓ ↓	C_1	*attend_agent*	*attending_agent*
$plan_{attend\_agent / attend\_cup / grasp}$ ↓ ↓	C_2	*attend_cup*	*attending_cup*
$plan_{attend\_agent / attend\_cup / grasp}$ ↓ ↓	C_3	*grasp*	*attending_agent*
$plan_{attend\_agent / attend\_cup / grasp}$ ↓ ↓	C_4		*attending_cup*

Deciding when to read out overt words

Only some of these SM areas have interfaces with phonology:

Sustained signals	Transient signals		
Planned action signals	Context signals	Action signals	Reafferent signals
$plan_{attend\_agent / attend\_cup / grasp}$ ↓ ↓	C_1	*attend_agent*	*attending_agent*
$plan_{attend\_agent / attend\_cup / grasp}$ ↓ ↓	C_2	*attend_cup*	*attending_cup*
$plan_{attend\_agent / attend\_cup / grasp}$ ↓ ↓	C_3	*grasp*	*attending_agent*
$plan_{attend\_agent / attend\_cup / grasp}$ ↓ ↓	C_4		*attending_cup*

Deciding when to read out overt words

A selective read-out for English SVO structure:

Sustained signals	Transient signals		
Planned action signals	Context signals	Action signals	Reafferent signals
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_1	*attend_agent*	**man**
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_2	*attend_cup*	*attending_cup*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_3	*grasp*	*attending_agent*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_4		*attending_cup*

Deciding when to read out overt words

A selective read-out for English SVO structure:

Sustained signals	Transient signals		
Planned action signals	Context signals	Action signals	Reafferent signals
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_1	*attend_agent*	**man**
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_2	*attend_cup*	*attending_cup*
grabs ↓ ↓	C_3	*grasp*	*attending_agent*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_4		*attending_cup*

Deciding when to read out overt words

A selective read-out for English SVO structure:

Sustained signals	Transient signals		
Planned action signals	Context signals	Action signals	Reafferent signals
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_1	*attend_agent*	**man**
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_2	*attend_cup*	*attending_cup*
grabs ↓ ↓	C_3	*grasp*	*attending_agent*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_4		**cup**

Deciding when to read out phonological items

A selective read-out for Māori VSO structure:

Sustained signals	Transient signals		
Planned action signals	Context signals	Action signals	Reafferent signals
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_1	*attend_agent*	*attending_agent*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_2	*attend_cup*	*attending_cup*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_3	*grasp*	*attending_agent*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_4		*attending_cup*

Deciding when to read out phonological items

A selective read-out for Māori VSO structure:

Sustained signals	Transient signals		
Planned action signals	Context signals	Action signals	Reafferent signals
$plan_{attend\_agent/attend\_cup/grasp}$ $\downarrow$ $\downarrow$	C_1	*attend_agent*	*attending_agent*
$plan_{attend\_agent/attend\_cup/grasp}$ $\downarrow$ $\downarrow$	C_2	*attend_cup*	*attending_cup*
$plan_{attend\_agent/attend\_cup/grasp}$ $\downarrow$ $\downarrow$	C_3	*grasp*	*attending_agent*
$plan_{attend\_agent/attend\_cup/grasp}$ $\downarrow$ $\downarrow$	C_4		*attending_cup*

Deciding when to read out phonological items

A selective read-out for Māori VSO structure:

Sustained signals	Transient signals		
Planned action signals	Context signals	Action signals	Reafferent signals
grabs ↓ ↓	C_1	*attend_agent*	*attending_agent*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_2	*attend_cup*	*attending_cup*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_3	*grasp*	*attending_agent*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_4		*attending_cup*

Deciding when to read out phonological items

A selective read-out for Māori VSO structure:

Sustained signals	Transient signals		
Planned action signals	Context signals	Action signals	Reafferent signals
grabs ↓ ↓	C_1	*attend_agent*	*attending_agent*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_2	*attend_cup*	*attending_cup*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_3	*grasp*	**man**
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_4		*attending_cup*

Deciding when to read out phonological items

A selective read-out for Māori VSO structure:

Sustained signals	Transient signals		
Planned action signals	Context signals	Action signals	Reafferent signals
grabs ↓ ↓	C_1	*attend_agent*	*attending_agent*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_2	*attend_cup*	*attending_cup*
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_3	*grasp*	**man**
$plan_{attend\_agent/attend\_cup/grasp}$ ↓ ↓	C_4		**cup**

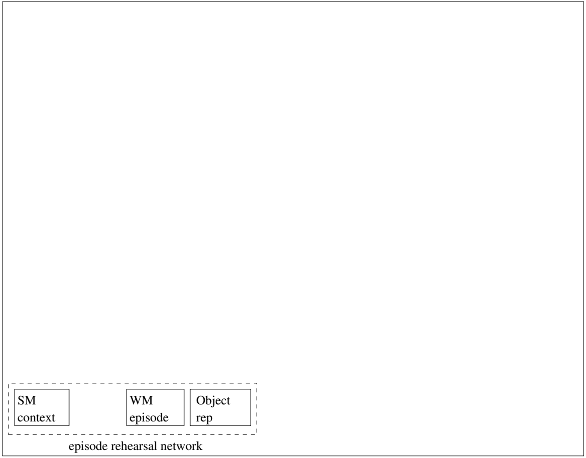
A neural network model of parameter-setting

I have developed a neural network model of sentence generation (jointly with Martin Takac and Lubica Benuskova).

- The network takes a SM sequence as input, and generates a word sequence as output.
- It's trained on pairs of (replayed) SM sequences and word sequences.
- It learns:
  (i) a vocabulary (a mapping from SM signals to words);
  (ii) rules about when the word associated with a SM signal should be pronounced.

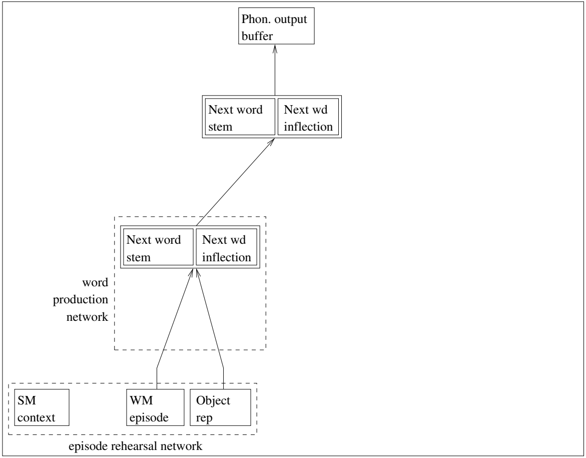
A neural network sentence generator

Here's the episode-rehearsal network.



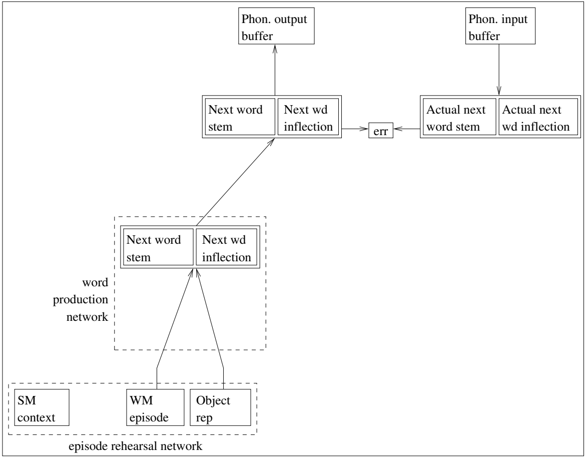
A neural network sentence generator

A **word production network** (WPN) maps SM signals to words in a **phonological output buffer.**



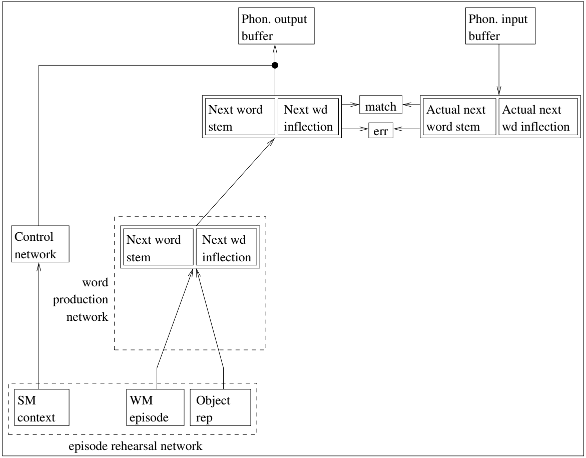
A neural network sentence generator

The WPN is trained to reproduce words replayed from a **phonological input buffer**.



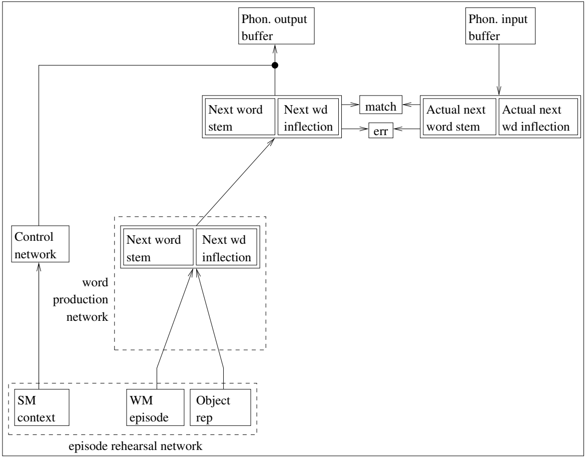
A neural network sentence generator

A **control network** decides when to *pronounce* and when to *withold* words produced by the WPN.



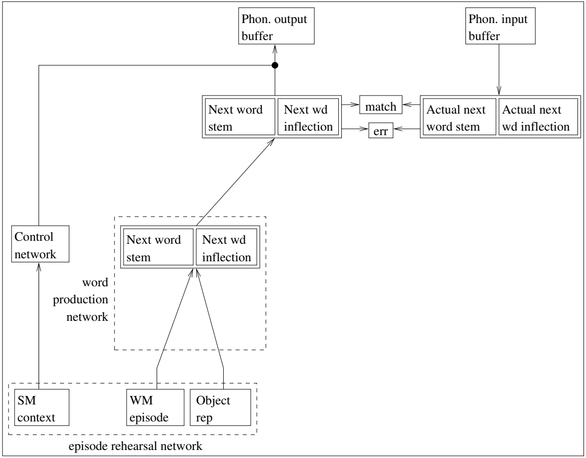
A neural network sentence generator

The control network's rules make no reference to words or word meanings; only to *contexts*.



A neural network sentence generator

The control network learns different *policies* for pronouncing/witholding words in different languages.



The policies learned by the control network

Here's the policy the network learns for a VSO training language:

Context	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*

Here's the policy it learns for an SVO language:

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

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the ‘next word’ in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the ‘next word’).
- If not, it learns to *withold* words in this context (and doesn’t update the ‘next word’).

Here’s a typical training item in a VSO language.

Context	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	**grabs man cup**						
SM signal							
output of WPN							
actual next wd							
‘match’ signal							
training signal for ctrl network							

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the 'next word' in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the 'next word').
- If not, it learns to *withold* words in this context (and doesn't update the 'next word').

Here's a typical training item in a VSO language.

Context	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	**grabs man cup**						
SM signal	MAN						
output of WPN							
actual next wd							
'match' signal							
training signal for ctrl network							

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the ‘next word’ in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the ‘next word’).
- If not, it learns to *withold* words in this context (and doesn’t update the ‘next word’).

Here’s a typical training item in a VSO language.

Context	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	**grabs man cup**						
SM signal	MAN						
output of WPN	*man*						
actual next wd							
‘match’ signal							
training signal for ctrl network							

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the 'next word' in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the 'next word').
- If not, it learns to *withold* words in this context (and doesn't update the 'next word').

Here's a typical training item in a VSO language.

Context	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	**grabs man cup**						
SM signal	MAN						
output of WPN	*man*						
actual next wd	*grabs*						
'match' signal							
training signal for ctrl network							

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the 'next word' in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the 'next word').
- If not, it learns to *withold* words in this context (and doesn't update the 'next word').

Here's a typical training item in a VSO language.

Context	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	**grabs man cup**						
SM signal	MAN						
output of WPN	*man*						
actual next wd	*grabs*						
'match' signal	no						
training signal for ctrl network							

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the 'next word' in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the 'next word').
- If not, it learns to *withold* words in this context (and doesn't update the 'next word').

Here's a typical training item in a VSO language.

Context	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	**grabs man cup**						
SM signal	MAN						
output of WPN	*man*						
actual next wd	*grabs*						
'match' signal	no						
training signal for ctrl network	—						

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the 'next word' in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the 'next word').
- If not, it learns to *withold* words in this context (and doesn't update the 'next word').

Here's a typical training item in a VSO language.

Context	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	**grabs man cup**						
SM signal	MAN	GRAB-PLAN					
output of WPN	*man*						
actual next wd	*grabs*						
'match' signal	no						
training signal for ctrl network	—						

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the 'next word' in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the 'next word').
- If not, it learns to *withold* words in this context (and doesn't update the 'next word').

Here's a typical training item in a VSO language.

Context	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	**grabs man cup**						
SM signal	MAN	GRAB-PLAN					
output of WPN	*man*	*grabs*					
actual next wd	*grabs*						
'match' signal	no						
training signal for ctrl network	—						

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the ‘next word’ in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the ‘next word’).
- If not, it learns to *withold* words in this context (and doesn’t update the ‘next word’).

Here’s a typical training item in a VSO language.

Context	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	**grabs man cup**						
SM signal	MAN	GRAB-PLAN					
output of WPN	*man*	*grabs*					
actual next wd	*grabs*	*grabs*					
‘match’ signal	no						
training signal for ctrl network	—						

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the 'next word' in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the 'next word').
- If not, it learns to *withold* words in this context (and doesn't update the 'next word').

Here's a typical training item in a VSO language.

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

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the ‘next word’ in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the ‘next word’).
- If not, it learns to *withold* words in this context (and doesn’t update the ‘next word’).

Here’s a typical training item in a VSO language.

Context	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	**grabs man cup**						
SM signal	MAN	GRAB-PLAN					
output of WPN	*man*	*grabs*					
actual next wd	*grabs*	*grabs*					
‘match’ signal	no	yes					
training signal for ctrl network	—	↓					

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the 'next word' in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the 'next word').
- If not, it learns to *withold* words in this context (and doesn't update the 'next word').

Here's a typical training item in a VSO language.

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

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the ‘next word’ in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the ‘next word’).
- If not, it learns to *withold* words in this context (and doesn’t update the ‘next word’).

Here’s a typical training item in a VSO language.

Context	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	**grabs man cup**						
SM signal	MAN	GRAB-PLAN	CUP				
output of WPN	*man*	*grabs*	*cup*				
actual next wd	*grabs*	*grabs*					
‘match’ signal	no	yes					
training signal for ctrl network	—	↓					

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the 'next word' in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the 'next word').
- If not, it learns to *withold* words in this context (and doesn't update the 'next word').

Here's a typical training item in a VSO language.

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

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the ‘next word’ in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the ‘next word’).
- If not, it learns to *withold* words in this context (and doesn’t update the ‘next word’).

Here’s a typical training item in a VSO language.

Context	C1a	C1b	C2a	C2b	C3a	C3b	C4a
Target utterance	**grabs man cup**						
SM signal	MAN	GRAB-PLAN	CUP				
output of WPN	*man*	*grabs*	*cup*				
actual next wd	*grabs*	*grabs*	*man*				
‘match’ signal	no	yes	no				
training signal for ctrl network	—	↓					

How the control network learns

The control network compares the word predicted by the WPN from the current SM signal to the 'next word' in the training utterance.

- If they match, it learns to *pronounce* words in this context (and updates the 'next word').
- If not, it learns to *withold* words in this context (and doesn't update the 'next word').

Here's a typical training item in a VSO language.

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

Syntactic agreement

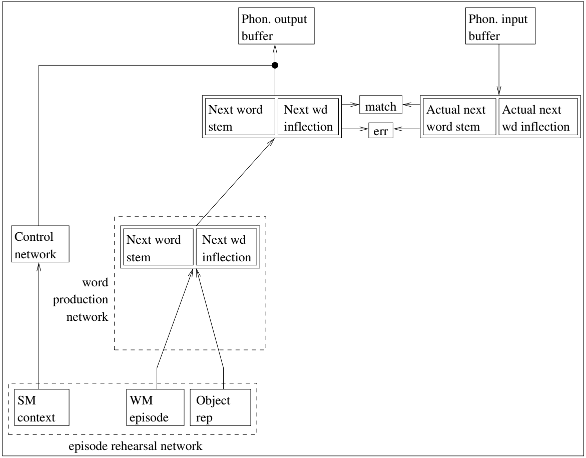
Our training language included a rich system of subject-verb agreement.

> *helen-3sg chase-3sg rabbit*

The network can learn such agreement rules, because WM episodes carry information about planned attentional operations as well as planned motor operations.

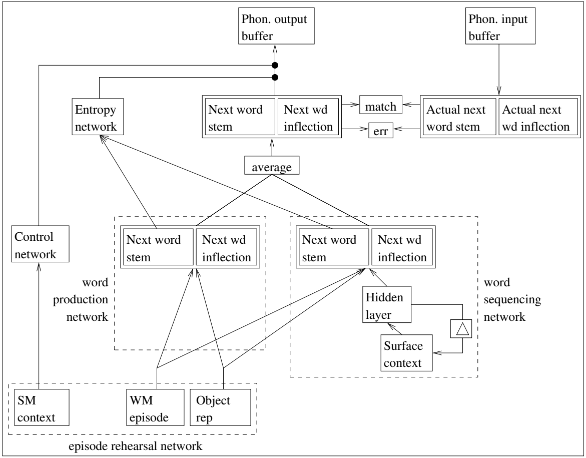
Adding an ability to learn surface word patterns

The model also includes a **word sequencing network** (**WSN**), which learns surface patterns of words in the training language.



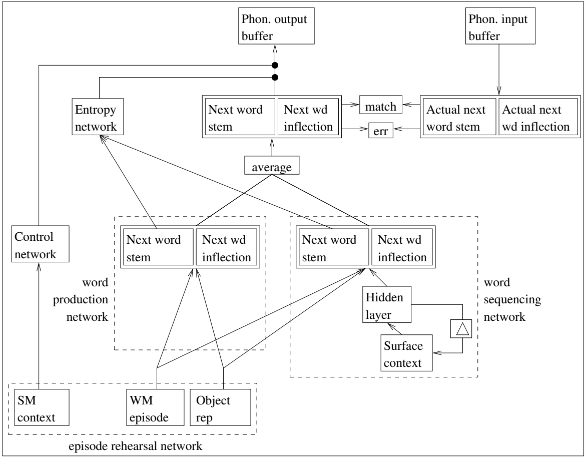
Adding an ability to learn surface word patterns

The model also includes a **word sequencing network** (**WSN**), which learns surface patterns of words in the training language.



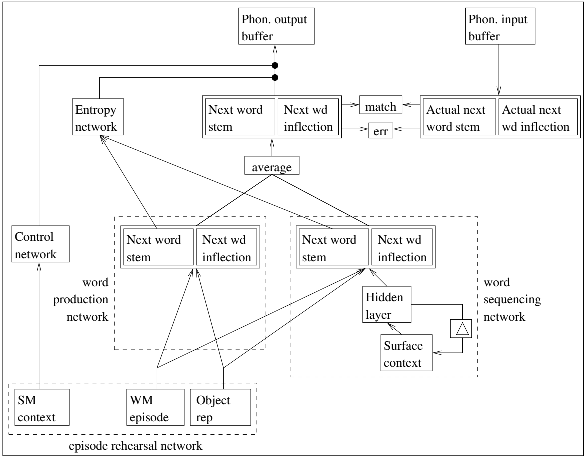
Adding an ability to learn surface word patterns

The WSN is a **recurrent network**, which learns to predict the next word on the basis of the words it has recently produced.



Adding an ability to learn surface word patterns

Recurrent networks are a traditional 'empiricist' language modelling tool.



A training language including idioms

We trained the network on a language including regular transitive sentences, but also various types of idiomatic construction.

> *daddy grabs cup*
> **winnie the pooh** *grabs cup*
> *daddy* **gives** *helen* **a hug**

The network can learn idiomatic constructions, as well as abstract syntactic rules.

(The **entropy network** decides when to produce idiomatic expressions.)

Simulating the timecourse of language development

The network learns vocabulary and idiomatic constructions at the same time as abstract rules.

During training it goes through developmental stages, which somewhat resemble those of children.

- A stage of single-word utterances
- A stage of item-based syntactic constructions
- A stage of 'mature syntax', with knowledge of abstract rules as well as idiomatic constructions.

Summary

Our network is something which empiricist linguists will recognise as a SRN-based sentence-processing model.

We hope that Minimalist linguists also recognise it as a model of:

- how infants bring ‘innate knowledge’ to the process of learning language;
- how infants learn the parameter values which define their own native language.