An Approach to Language Understanding and Contextual Disambiguation in Human-Robot Interaction

Heiner Markert, Zöhre Kara Kayikci, Günther Palm
Department of Neural Information Processing
University of Ulm

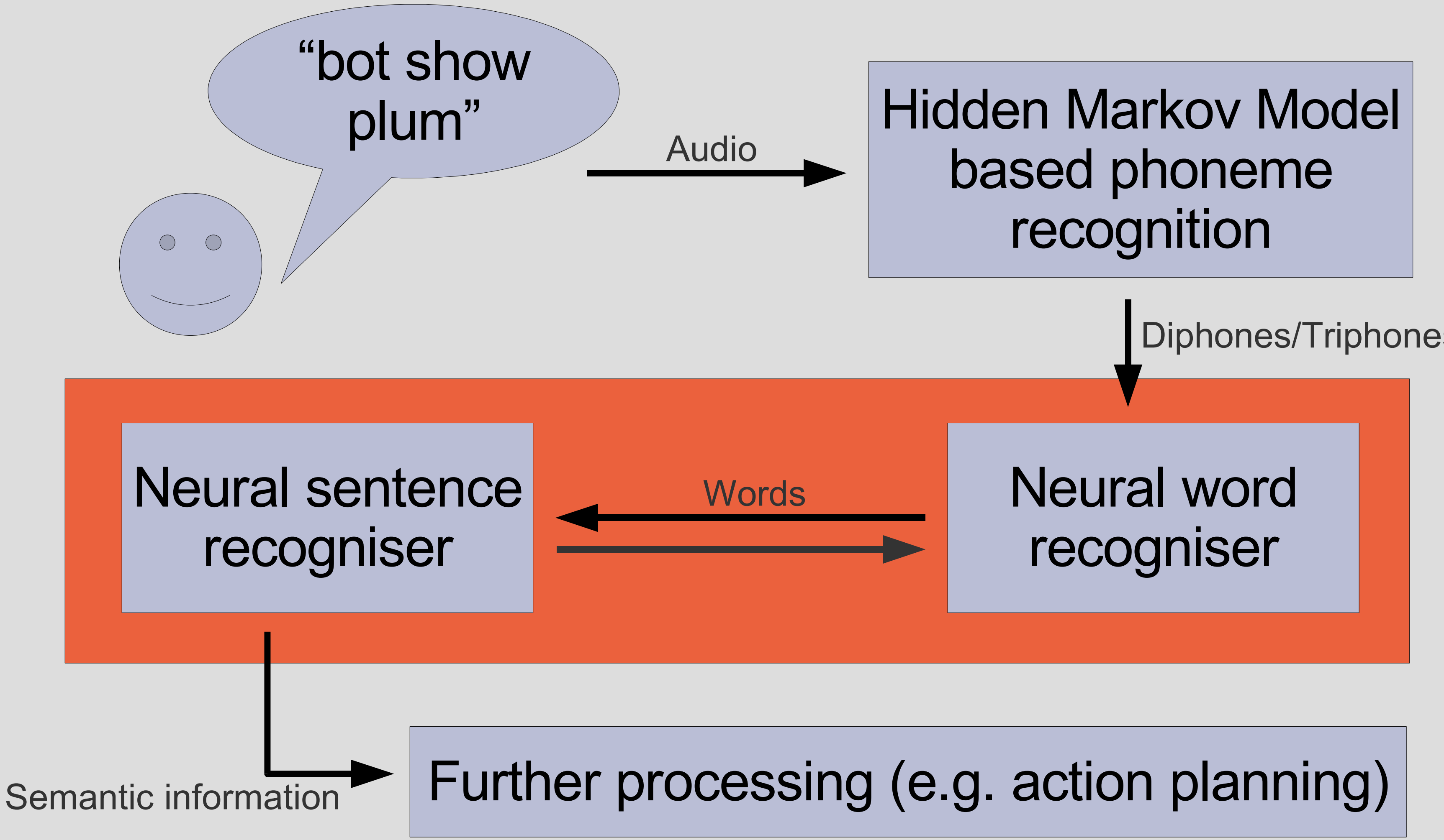
Outline

- Motivation
- Functional blocks of the language processing system
- A simple word recognition network: Overview
- Basic mechanisms of the simulation
  - “cortical modules” implemented as associative memories
  - Display for neural activation
- A simple word recognition network: Example
- The sentence recognition network
  - Overview
  - Quick example
- Conclusions and future work

Motivation

- Functional large scale brain modelling with biologically plausible neural networks
- Fast simulation of neural networks
- Human language understanding is interesting because:
  - Handling of ambiguities required
  - Symbol grounding aspect: Translate sub-symbolic word representations to symbolic (word level or semantic) representations
  - Applications in many fields

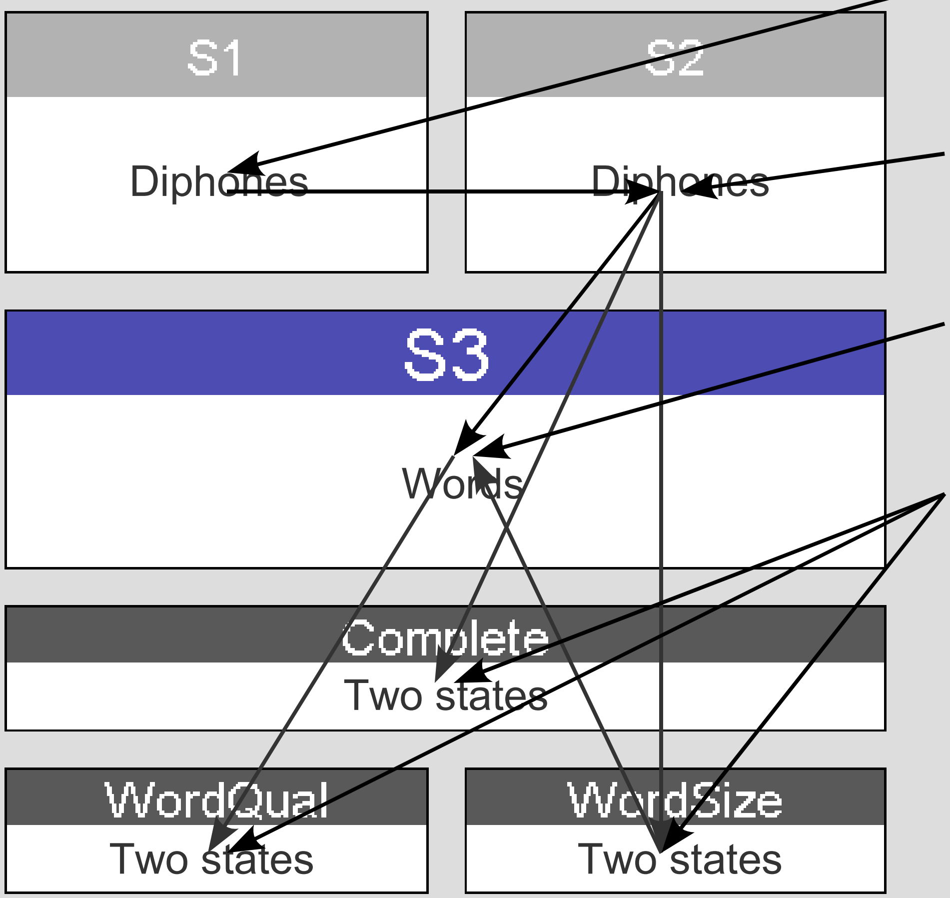
Language processing: Functional blocks



A simple word recognition net

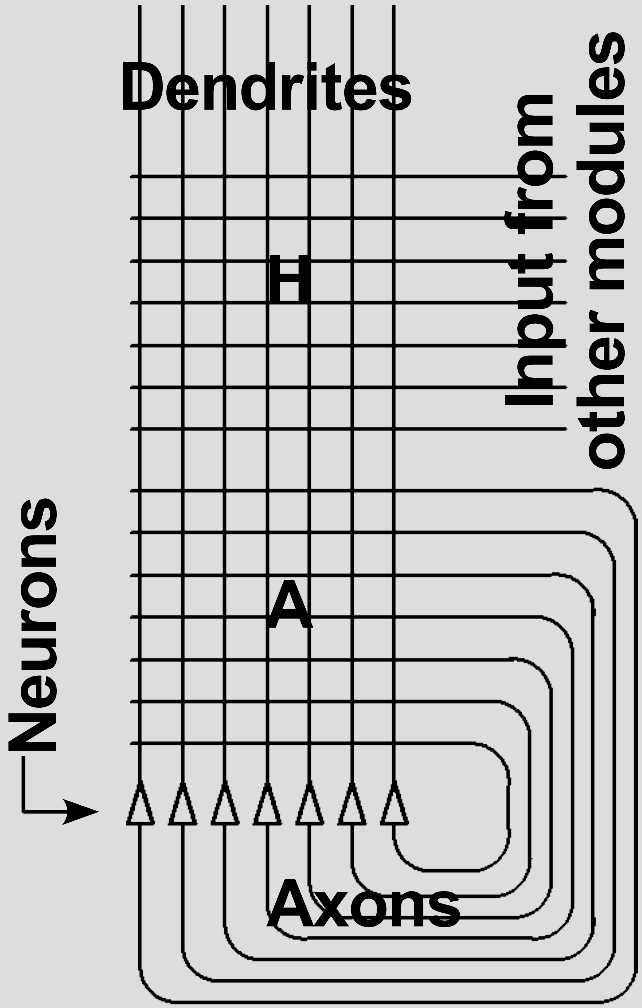
- 6 “cortical modules” (boxes)
- Each cortical module is modelled as a binary autoassociative memory
- “Spike counter populations” using sparse representa-tions
- The modules are connected via heteroassociative links

A simple word recognition net



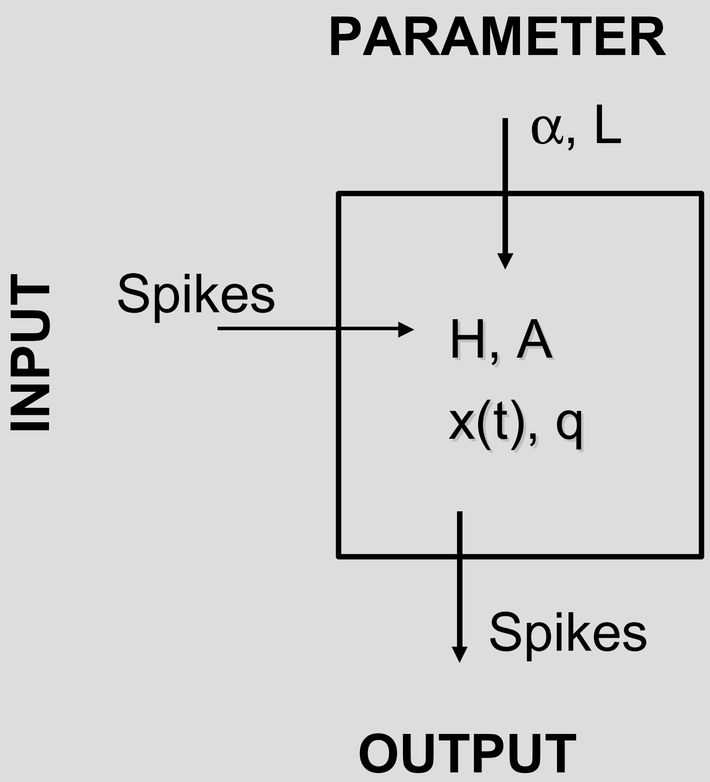
- S1: Input module receiving diphones from HMM
- S2: Holds last ten diphones (as superposition)
- S3: Stores the words known in the system
- Other: Additional status information
- Connectivity: Arrows correspond to heteroassociations

Basic mechanism: Cortical module I



- Cortical modules are modelled as neural associative memory
- A: autoassociative coupling matrix (patterns / assemblies)
- H: heteroassociative coupling matrix (Input from other modules)
- All connection weights are binary (0 or 1)

Basic mechanism: Cortical module II



Parameters:

α: separation strength
L: learn signal
q: quality measure

Coupling matrices:

H: Heteroassociation
A: Autoassociation

Example spike list:

```
queue 0 (index, time, strength):
0 199.998 1
598 214.284 1
1025 219.999 1
1816 225.714 1
131 231.43 1
2 237.145 1
```

Basic mechanism: Cortical module III

The neuron model for one global time step *s* is given by

$$\dot{x}_s(t) = \underbrace{a \cdot c_s^H(t) + b \cdot c_s^H(\infty)}_{\text{Heteroassociation}} \cdot \alpha \cdot \left(c + L \underbrace{\left(\frac{c_s^A(t)}{c_s^\Sigma(t)} \right)}_{\substack{\text{Separating} \\ \text{inhibition}}} \right) + \underbrace{d \cdot c_s^F(t) + e \cdot c_s^F(\infty)}_{\text{local feedback}}$$

where $x_s(0)=0$ and

$$\underbrace{c_s^H(t) \sim r(s-D,t) \cdot H}_{\text{Heteroassociative input}} \quad \underbrace{c_s^A(t) = y_s(t-d) \cdot A}_{\text{Autoassociative input}} \quad \underbrace{c_s^\Sigma(t) = y_s(t) \cdot \mathbf{1}}_{\substack{\text{Current module} \\ \text{activation level}}} \quad \underbrace{c_s^F(t) \sim r(s-1,t) \cdot A}_{\substack{\text{Short-term memory} \\ \text{feedback}}}$$

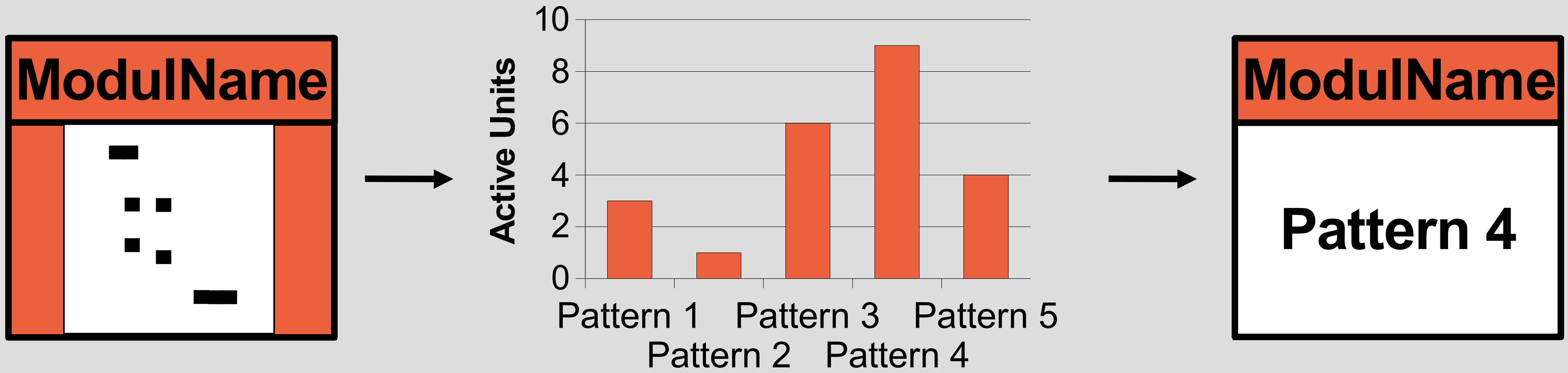
The vector r is called “instantaneous rate” and is defined by

$$r_i(s, t_{\max}) = 1/\min\{t \le t_{\max} : y_i(t) = 1\}$$

The neuron activity y is given by $y_i^s(t) = \mathbf{1}_{[x_i^s(t) \ge \Theta]}(t)$

More details: See Markert/Knoblauch/Palm: “Modelling of syntactical processing in the cortex”, to appear in BioSystems, 2006

Basic mechanism: Display of neural activity

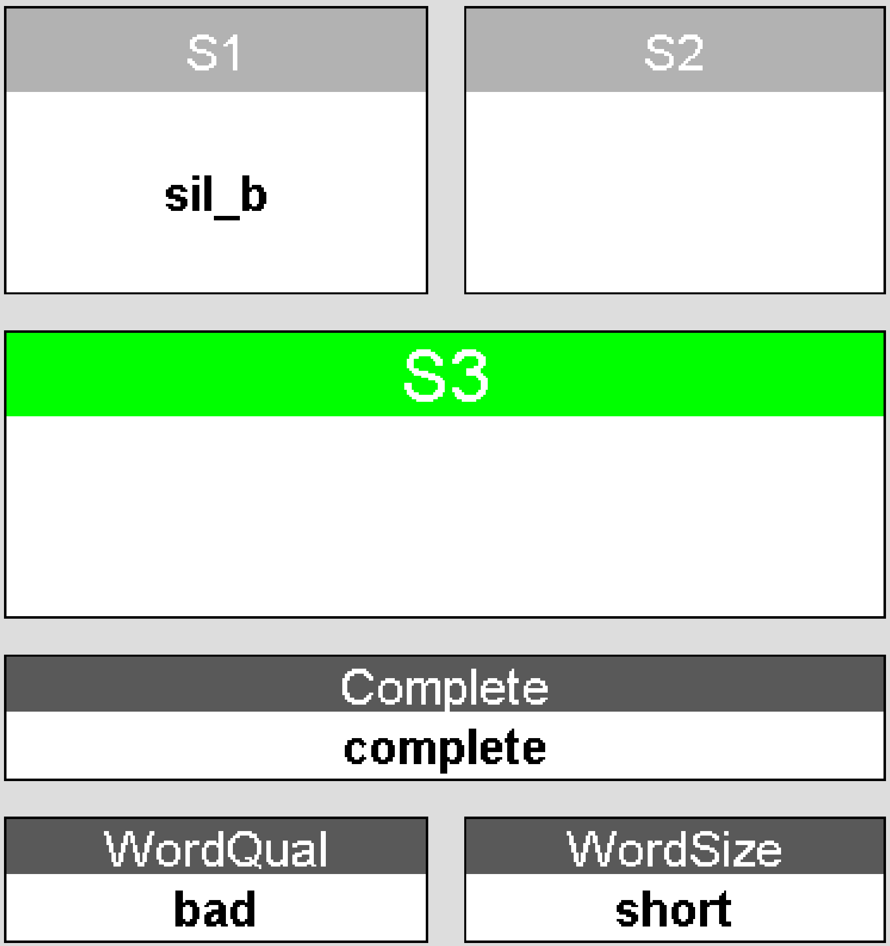


Step 1:
Pattern of active neurons

Step 2:
Histogram of pattern activation

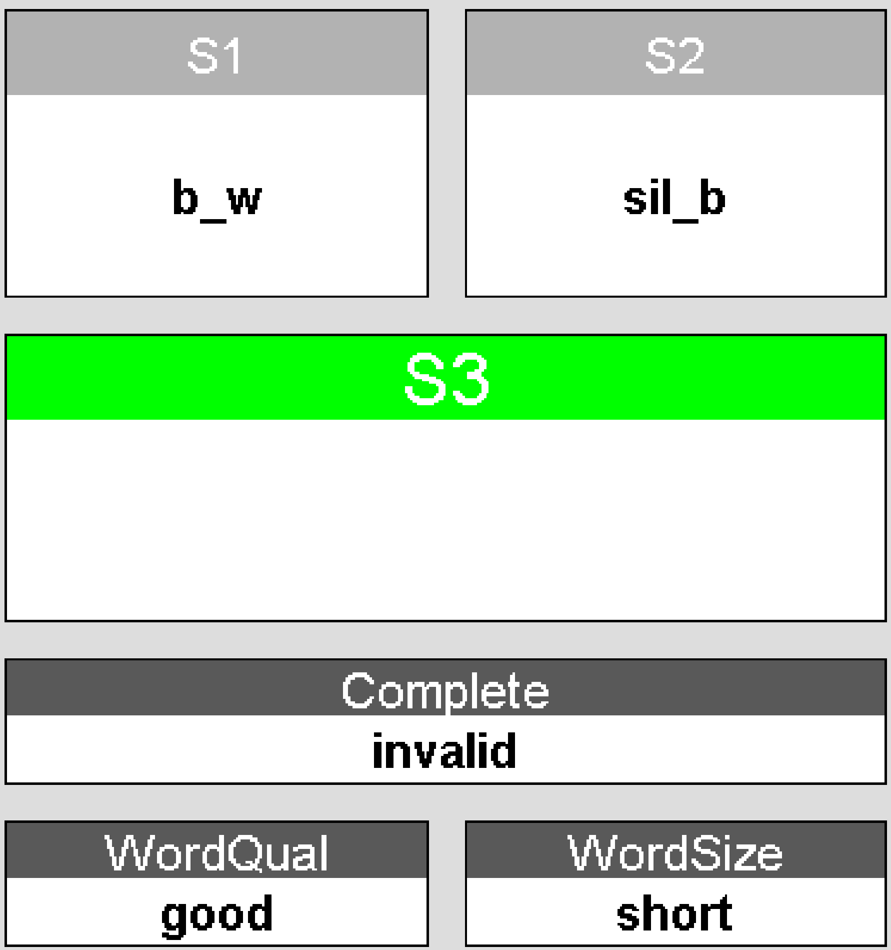
Step 3:
Display name of best matching pattern

Example: “bwall”



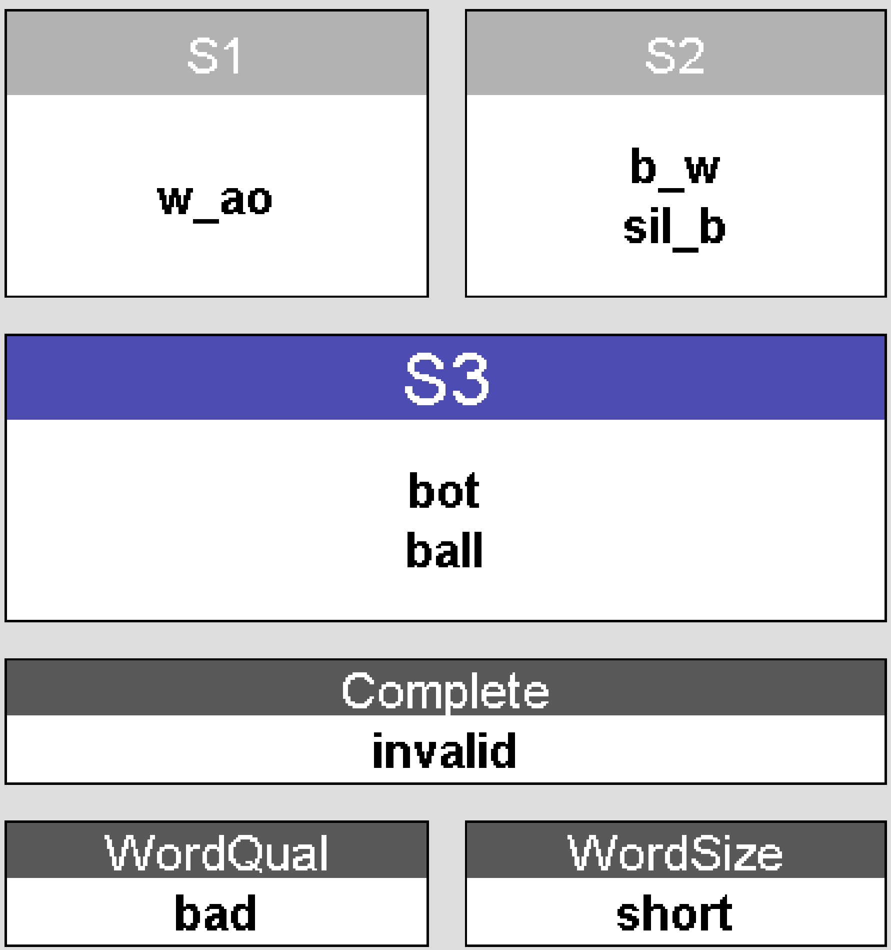
- “sil_b” is entering S1

Example: “bwall”



- “b_w” is entering S1
- “sil_b” is stored in S2

Example: “bwall”



- “w_ao” is entering S1
- “sil_b” is stored in S2
- “b_w” is stored in S2

- S3 suggests “bot” or “ball”

Example: “bwall”

S1	S2
ao_l	w_ao b_w sil_b

S3
wall bot ball

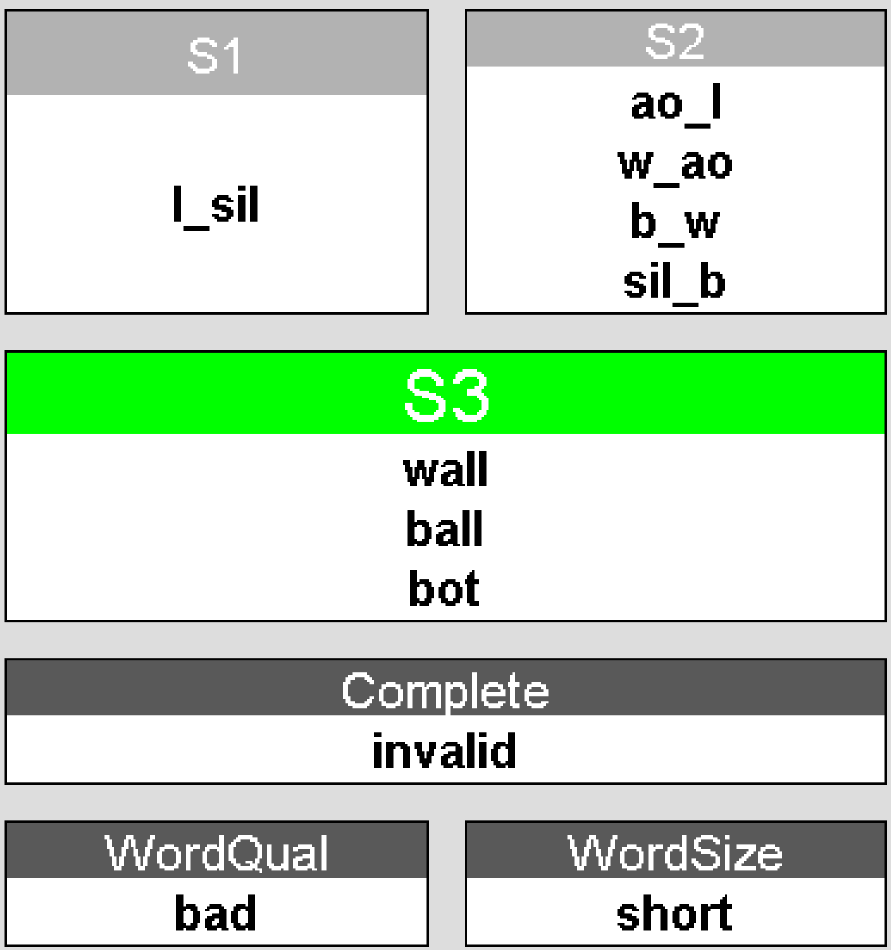
Complete
invalid

WordQual	WordSize
bad	short

- “ao_l” is entering S1
- “sil_b” is stored in S2
- “b_w” is stored in S2
- “w_ao” is stored in S2

- S3 suggests “wall”, “bot” or “ball”

Example: “bwall”



- “l_sil” is entering S1
- “sil_b” is stored in S2
- “b_w” is stored in S2
- “w_ao” is stored in S2
- “ao_l” is stored in S2

- S3 suggests “wall”, “ball” or “bot”

Example: “bwall”

S1

S2
l_sil
ao_l
w_ao
b_w
sil_b

S3
wall
ball

Complete
complete

WordQual
bad

WordSize
short

- “sil_b” is stored in S2
- “b_w” is stored in S2
- “w_ao” is stored in S2
- “ao_l” is stored in S2
- “l_sil” is stored in S2

- S3 suggests “wall” or “ball”

Word recognition: Performance

- Timit speech corpus
- 1 fold of a 105-fold cross validation:
  - Training uses 624 out of 630 speakers with all 5 sentences per speaker, meaning 20483 words in total
  - Test data: 6 remaining speakers with 5 sentences per speaker, meaning 221 test words in total

- Triphone HMM:
  80% correct triphones
- Word level:
  - HMM:
    74.6% correct words
  - Simple network*:
    65.2% correct words

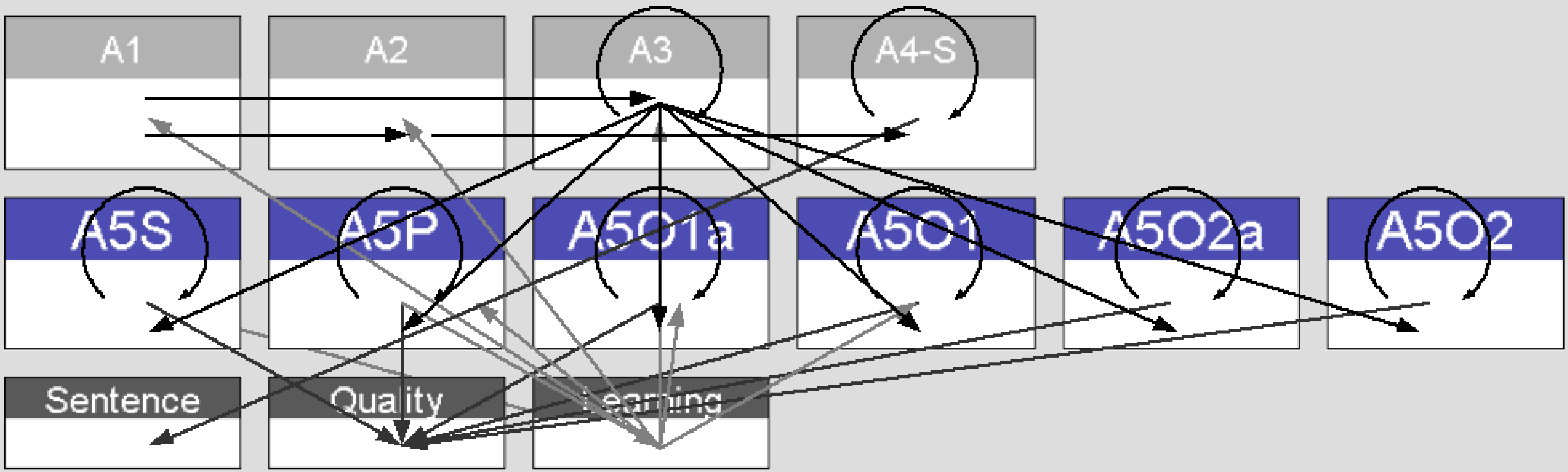
* For this evaluation, a classical binary associative memory (Willshaw's model) with some simple postprocessing to decide for exactly one word was used.

Sentence recognition network

A1 Words, Delimiters	A2 Delimiters	A3 Words	A4-S States		
A5S Subject-words	A5P Verbs	A5O1a Adjectives	A5O1 Object-words	A5O2a Adjectives	A5O2 Object-words
Sentence Two states	Quality Two states	Learning Two states			

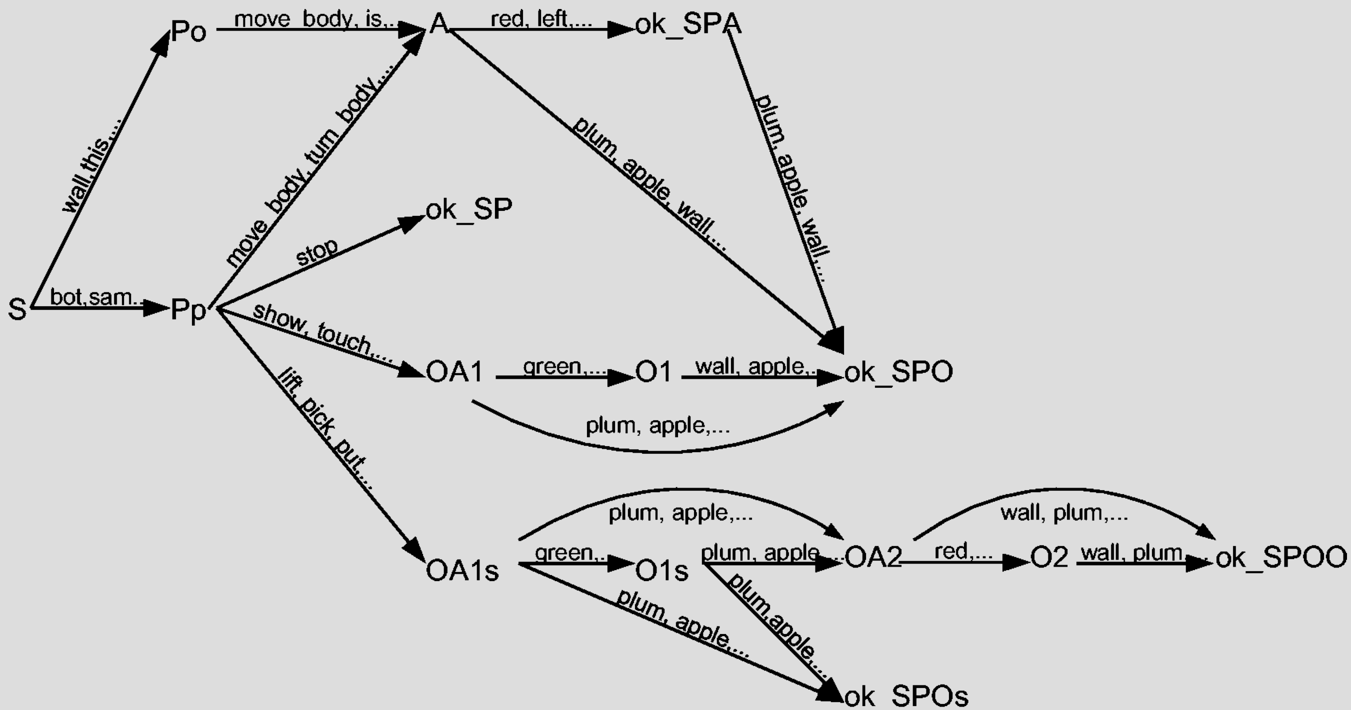
- Sentence recognition network parses stream of words with respect to a given regular grammar
- Symbol grounding
  - Input is sub-symbolic (word level) representation of words
  - Output is symbolic (semantic/syntactical) representation

Sentence recognition network



- Sentence recognition network parses stream of words with respect to a given regular grammar
- Symbol grounding
  - Input is sub-symbolic (word level) representation of words
  - Output is symbolic (semantic/syntactical) representation

Sentence recognition: Graph memory A4



- Each path represents one allowed sentence type
- Our architecture allows for modelling arbitrary deterministic finite automata with neural networks.

Sentence recognition
Example “bot lift bwall”

Step 16

A1	A2	A3	A4-S
bot			**S**

A5S	A5P	A5O1a	A5O1	A5O2a	A5O2

Sentence	Quality	Learning
incomplete	**good**	**false**

Sentence recognition Example “bot lift bwall”

Step 18

A1	A2	A3	A4-S
bot	**bot**	**_word**	**S**

A5S	A5P	A5O1a	A5O1	A5O2a	A5O2
bot					

Sentence	Quality	Learning
incomplete	**good**	**false**

Sentence recognition Example "bot lift bwall"

Step 37

A1	A2	A3	A4-S		
lift	lift	_word	Pp		
A5S	**A5P**	**A5O1a**	**A5O1**	**A5O2a**	**A5O2**
bot	lift				
Sentence	Quality	Learning			
incomplete	good	false			

Sentence recognition
Example “bot lift bwall”

Step 56

A1	A2	A3	A4-S		
wall ball	wall ball	_word	OA1s		
A5S	**A5P**	**A5O1a**	**A5O1**	**A5O2a**	**A5O2**
bot	lift	_obj	wall ball		

Sentence	Quality	Learning
incomplete	good	false

Sentence recognition Example “bot lift bwall”

Step 57

A1	A2	A3	A4-S
wall ball	wall ball	_word	OA1s

A5S	A5P	A5O1a	A5O1	A5O2a	A5O2
bot	lift	_obj	ball apple lemon orange tangerine		

Sentence	Quality	Learning
incomplete	bad	false

Sentence recognition
Example “bot lift bwall”

Step 66

A1	A2	A3	A4-S
			ok_SPOs

A5S	A5P	A5O1a	A5O1	A5O2a	A5O2
bot	**lift**	**_obj**	**ball**		

Sentence	Quality	Learning
incomplete	**bad**	**false**

Sentence recognition
Example “bot lift bwall”

Step 68

A1	A2	A3	A4-S		
			ok_SPOs		
A5S	**A5P**	**A5O1a**	**A5O1**	**A5O2a**	**A5O2**
bot	lift	_obj	ball		

Sentence	Quality	Learning
complete	good	false

Conclusions / Future Work

- Conclusions
  - Word and language understanding is possible with simplified neural networks
  - Representing, handling and resolving ambiguities is well supported by our architecture
  - Close to real time simulation is possible on standard laptop machines
- Future Work
  - Improvement of word recognition network (more sophisticated architecture) to increase recognition rate
  - Top-down information from language to word recognition
  - Handle ambiguities on the grammar level (e.g. “bot put orange orange orange plum”)
  - More vocabulary, more sentence types

Thank you for your attention!