

# Experiments

Dieuwke Hupkes

## 1 Datasets

I define the following set of languages:

Name	Numeric leaves	Example
$L_2$	2	$(x_1 \text{ op } x_1)$
$L_3$	3	$((x_1 \text{ op } x_2) \text{ op } x_3)$
$L_4$	4	$((x_1 \text{ op } x_2) \text{ op } (x_3 \text{ op } x_4))$
...		

Where  $x_i \in \{-19, 19\}$ , and  $\text{op} \in \{+, -\}$ . The meaning  $y$  of e sentences is the result of the arithmetic expression expressed by the language. We restrict the languages to include only expressions such that  $y \in \{-60, 60\}$ .

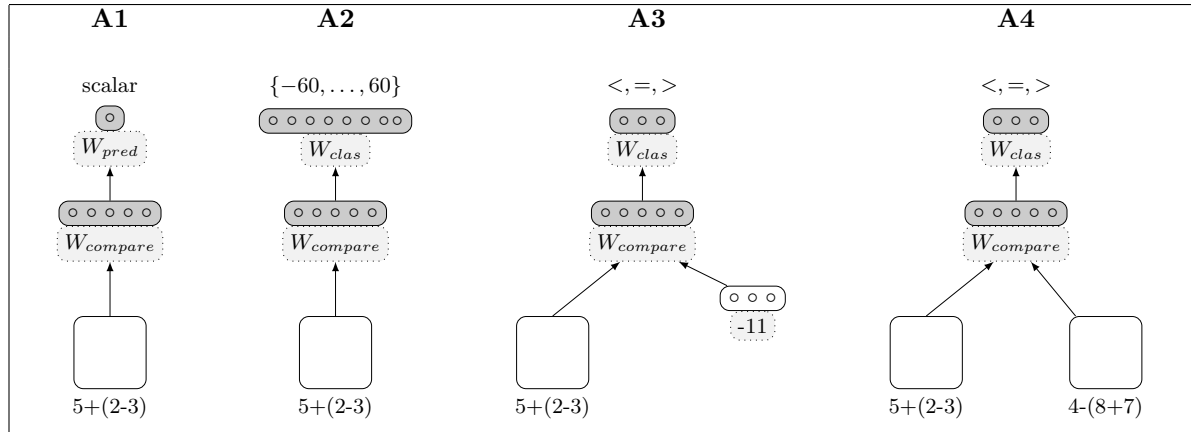
We define the following subsets of the languages defined above:

Name	Restriction	Example	
$L_i+$	$\text{op} == +$	$(.(.(x_1 + x_2) + \dots x_i)$	Structurally non ambiguous
$L_i-$	$\text{op} == -$	$(.(.(x_1 - x_2) - \dots x_i)$	
$L_i\text{rb}$	only right branching trees	$(.(.(x_1 \text{ op } x_2) \text{ op } x_3) \text{ op } \dots x_i)$	Structurally non ambiguous
$L_i\text{lb}$	only left branching trees	$(x_1 \text{ op } (x_2 \text{ op } (\dots \text{ op } (x_{i-1} \text{ op } x_i).).))$	Structurally non ambiguous

The datasets that the networks will be trained and tested on are (subsets of) unions of the languages described above.

## 2 Architectures

I use four different architectures (explanation?):



### 3 Experiments

I will start by running a sequence of experiments to determine if the networks can learn to compose the meaning of sentences from the structurally non ambiguous languages  $L_2$ ,  $L_3+$ ,  $L_3lb$  and  $L_3rb$ . Depending on the results I will move on to more complicated languages In principle, I would like to do all (possible) combinations that can be made by combining elements from the following table,<sup>1</sup> starting with architectures A1 and A2 and then expanding to A3 and A4.

Network	Language	Architecture	Dimensionality	Initialisation	Embeddings
SRN	$L_2$	A1	10	Random	fixed
GRU	$L_3+$	A2	6	Gray	trained
LSTM	$L_3rb$	A3	2	one-hot?	
	$L_3lb$	A4			

### 4 More concrete plan

Wat moet ik doen?

- Generate datasets (or think of how to do that on the fly)
- Implement architectures
- Train networks and plot results

---

<sup>1</sup>Of course excluding non-sensical combinations, such as Gray encoding in two dimennions