Artificial neural networks and the challenge of compositional generalization

Marco Baroni

Facebook AI Research

Outline

- The (second) neural network coming
- Systematic compositionality
- A compositional challenge for neural networks
- (If time allows): Looking for a compositional neural network in a haystack

The (second) neural network coming

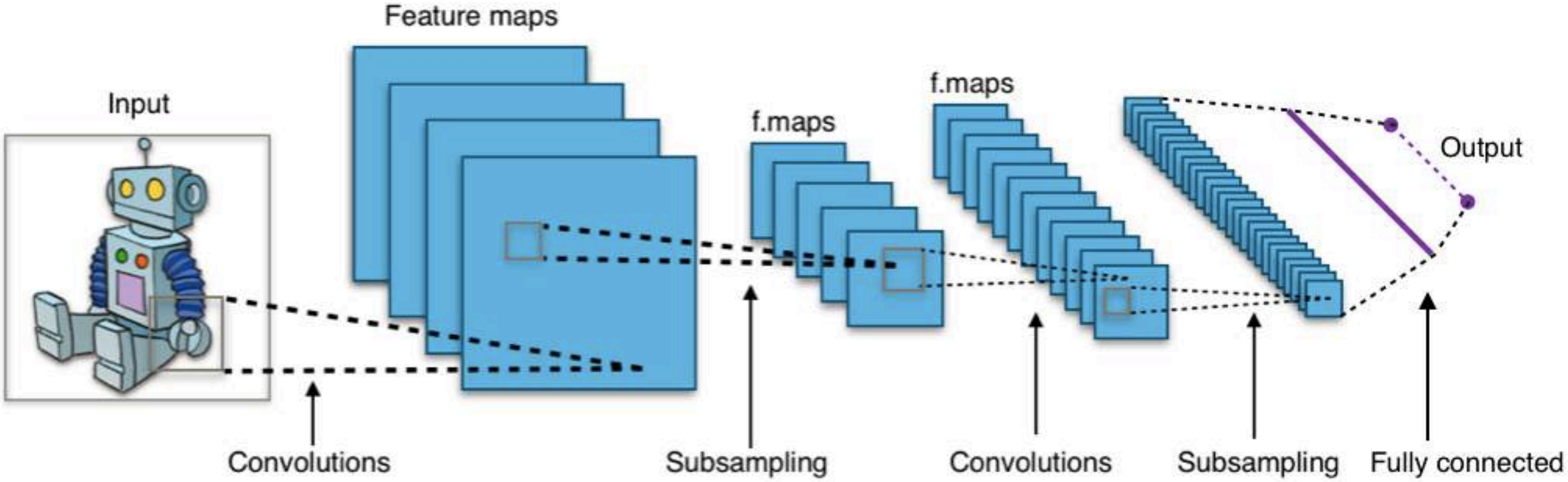


Image credit: Aphex34

The ImageNet Visual Recognition Challenge

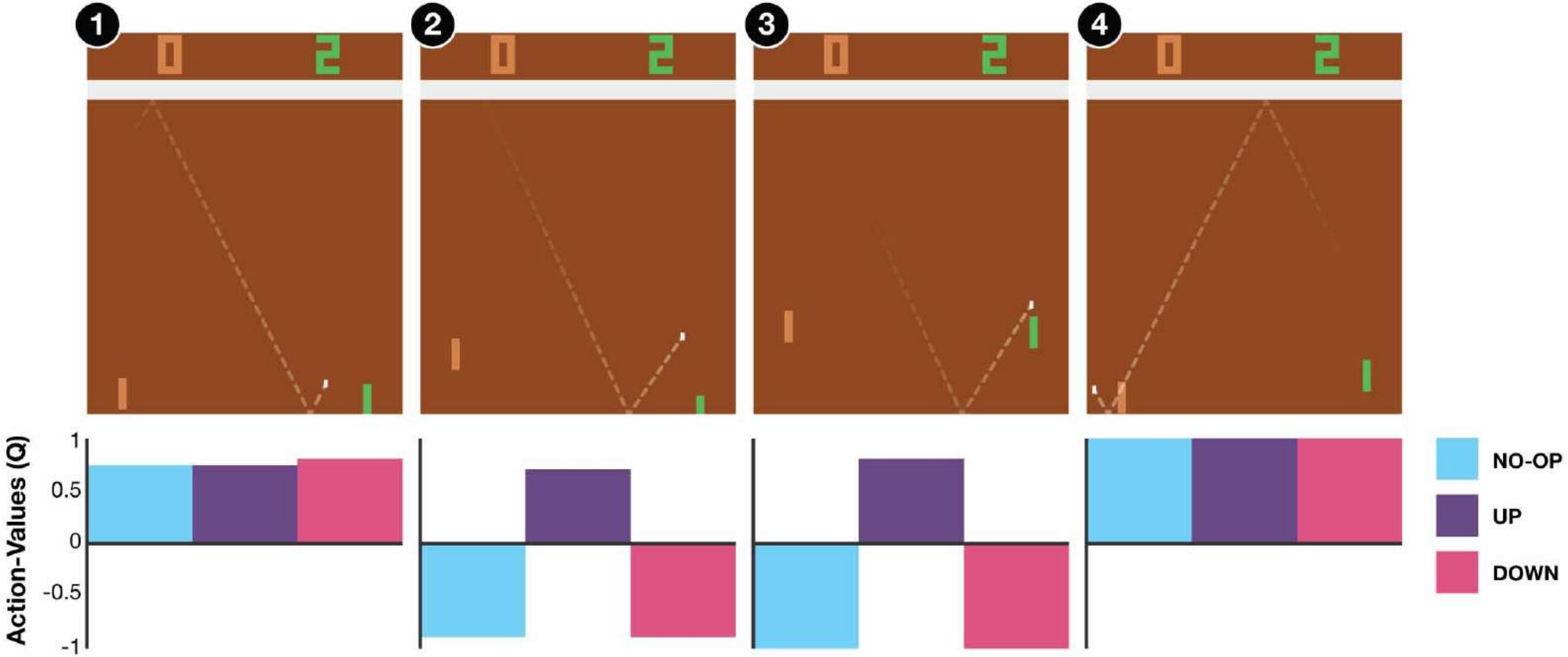
https://cs.stanford.edu/people/karpathy/cnnembed/

- Recognize objects in natural pictures of 1,000 categories
- Including African elephant and Indian elephant, Norfolk terrier and Yorkshire terrier...
- (Top-5) accuracy: from 74% in 2011, to 84% in 2012 when neural networks were first used
- 93% in 2014 (arguably, "super-human)

Learning from examples leads to acquiring human-like fuzzy categories

DOG

The (second) neural network coming

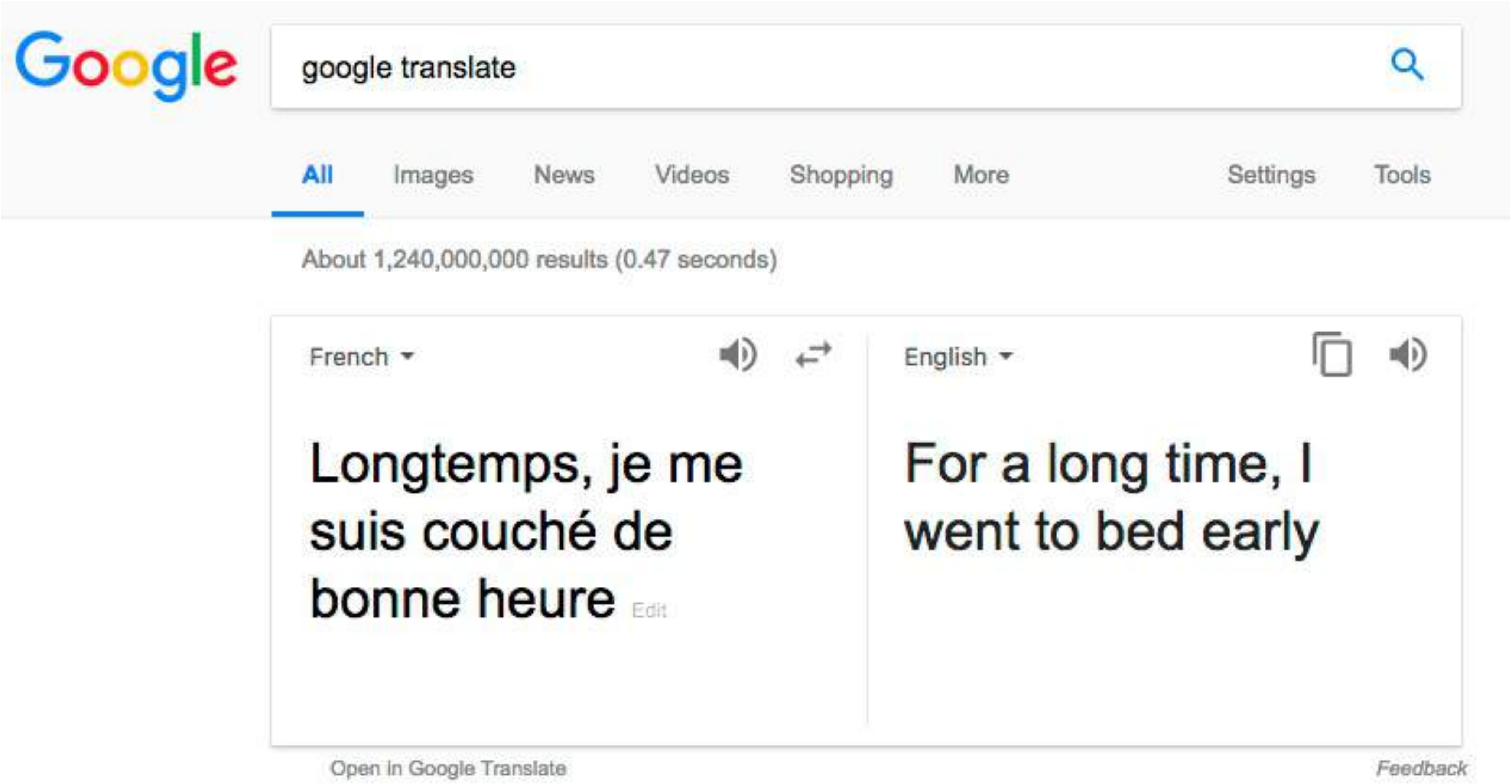


Mnih et al. 2015

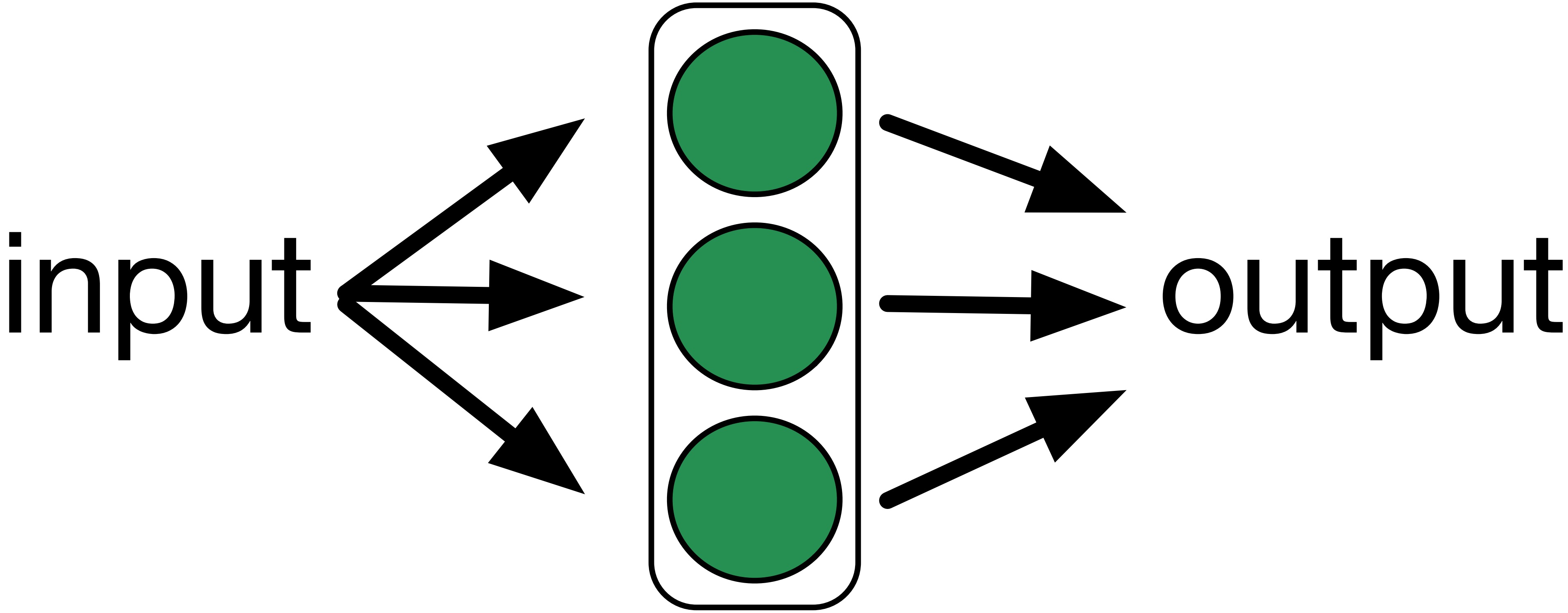
The (second) neural network coming

Silver et al. 2016

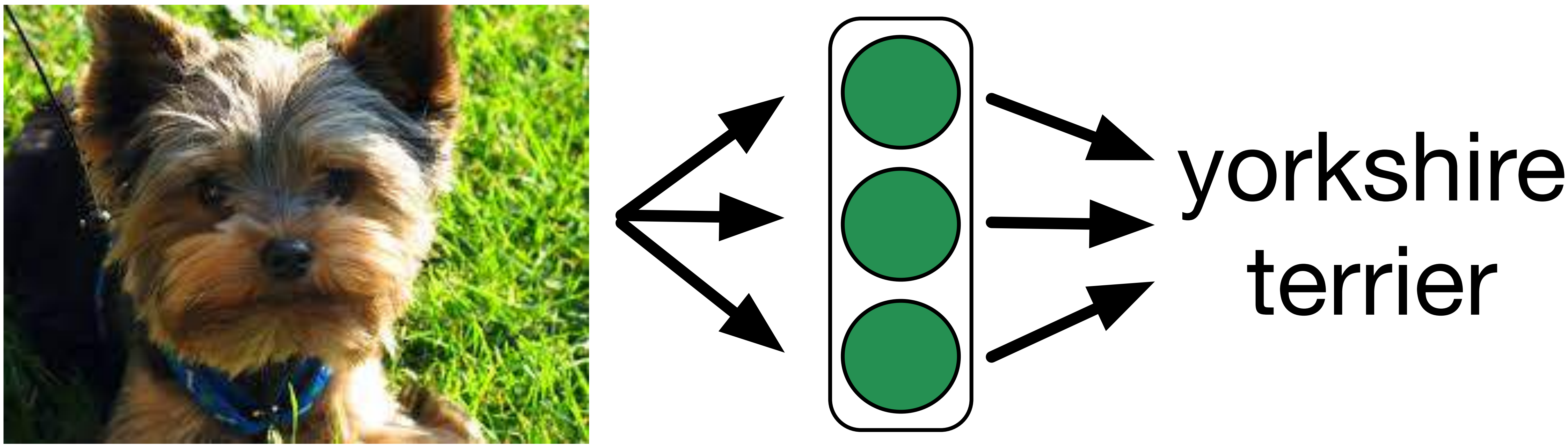
The (second) neural network coming



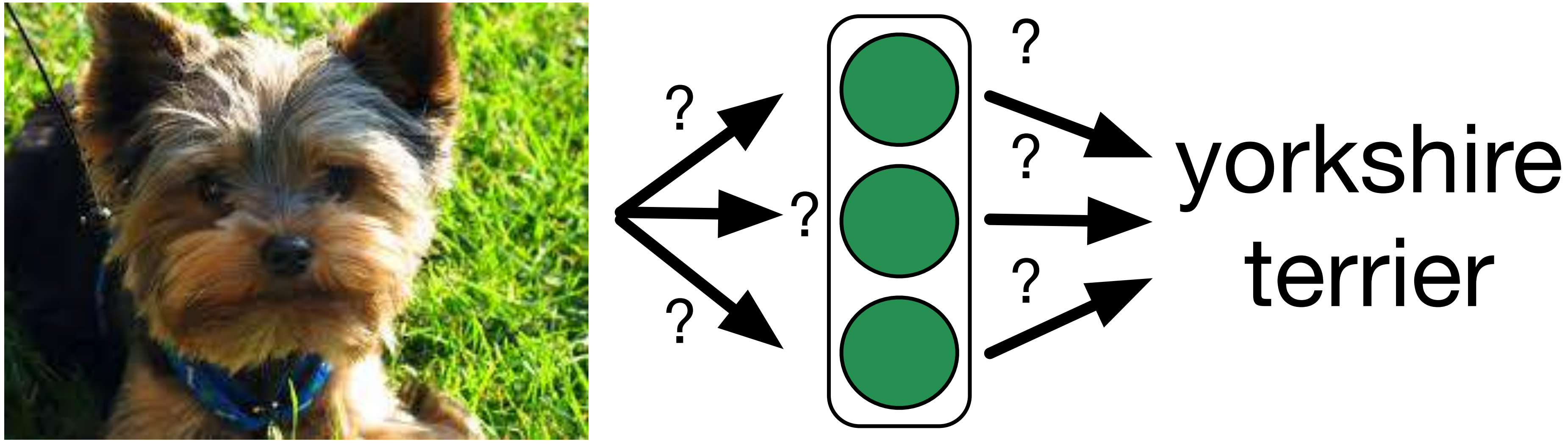
Artificial neural networks



Artificial neural networks



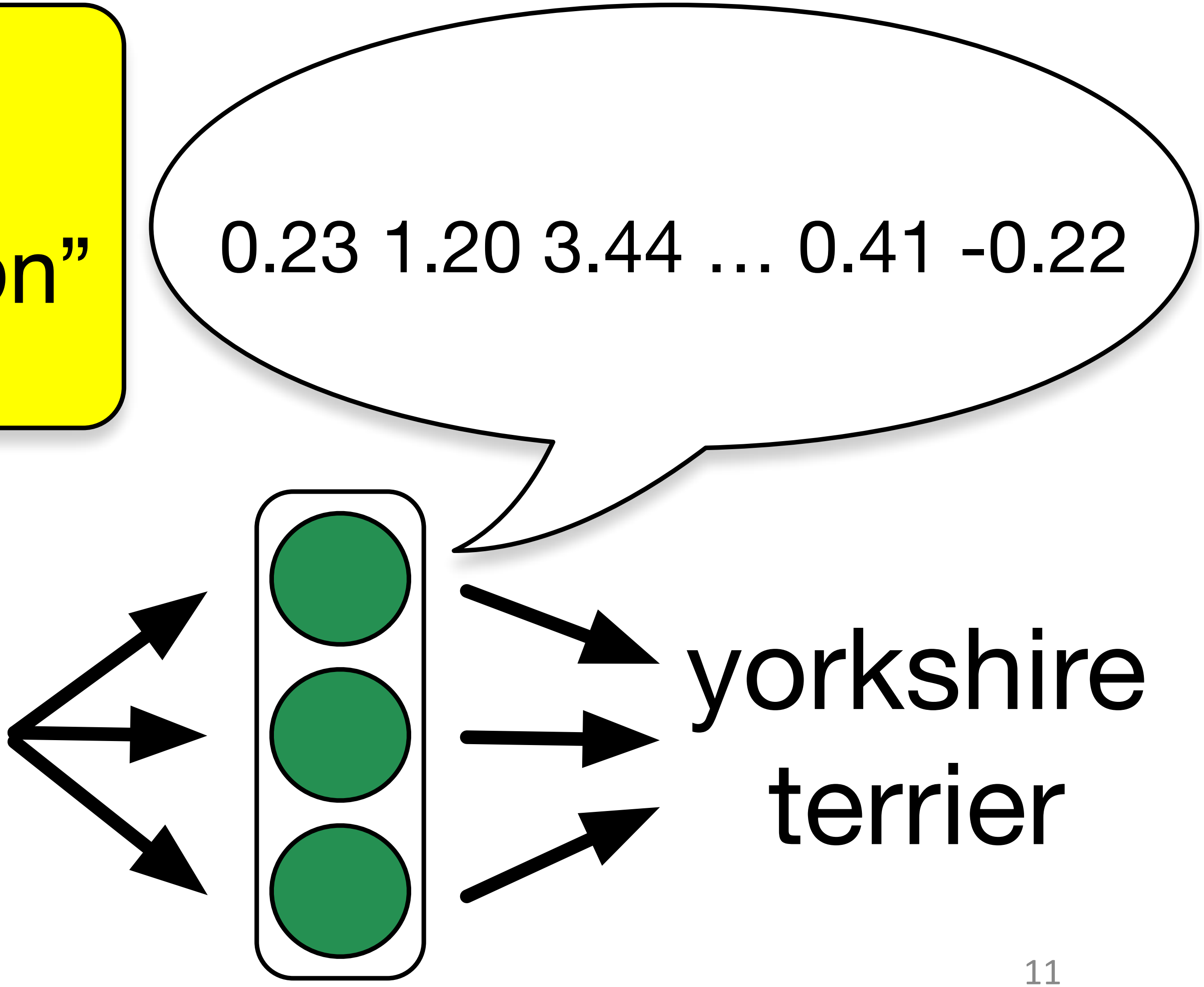
Artificial neural networks



"training" consists in optimally setting network weights to produce right output for each example input

Artificial neural networks

network automatically produces its own “distributed representation” of the input



The generality of neural networks

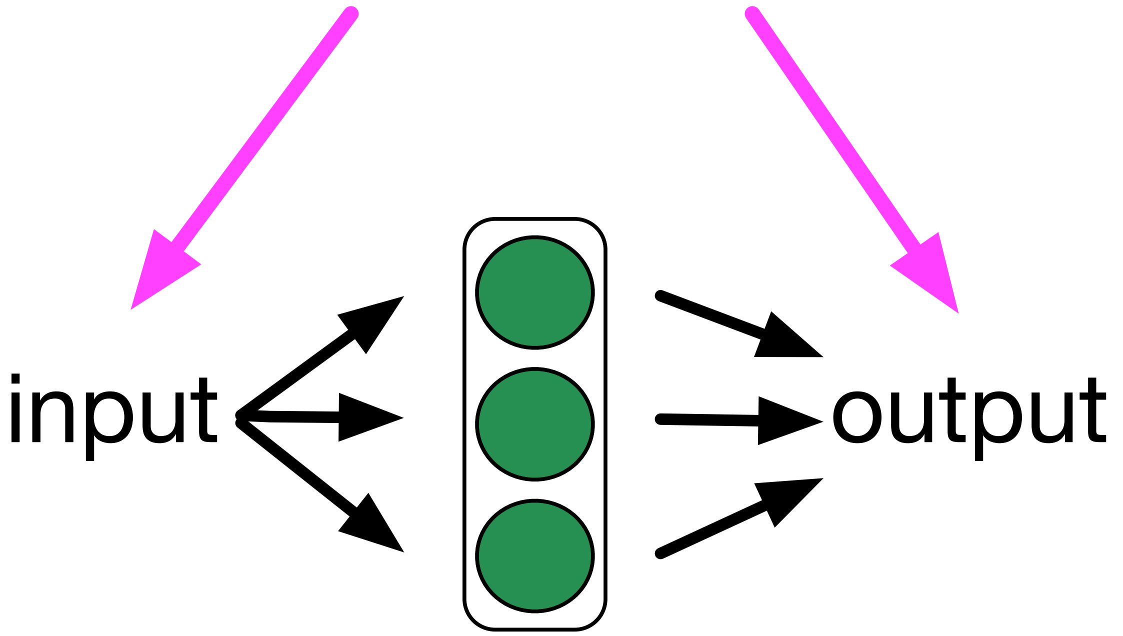
I: images, O: object labels

I: documents, O: topics

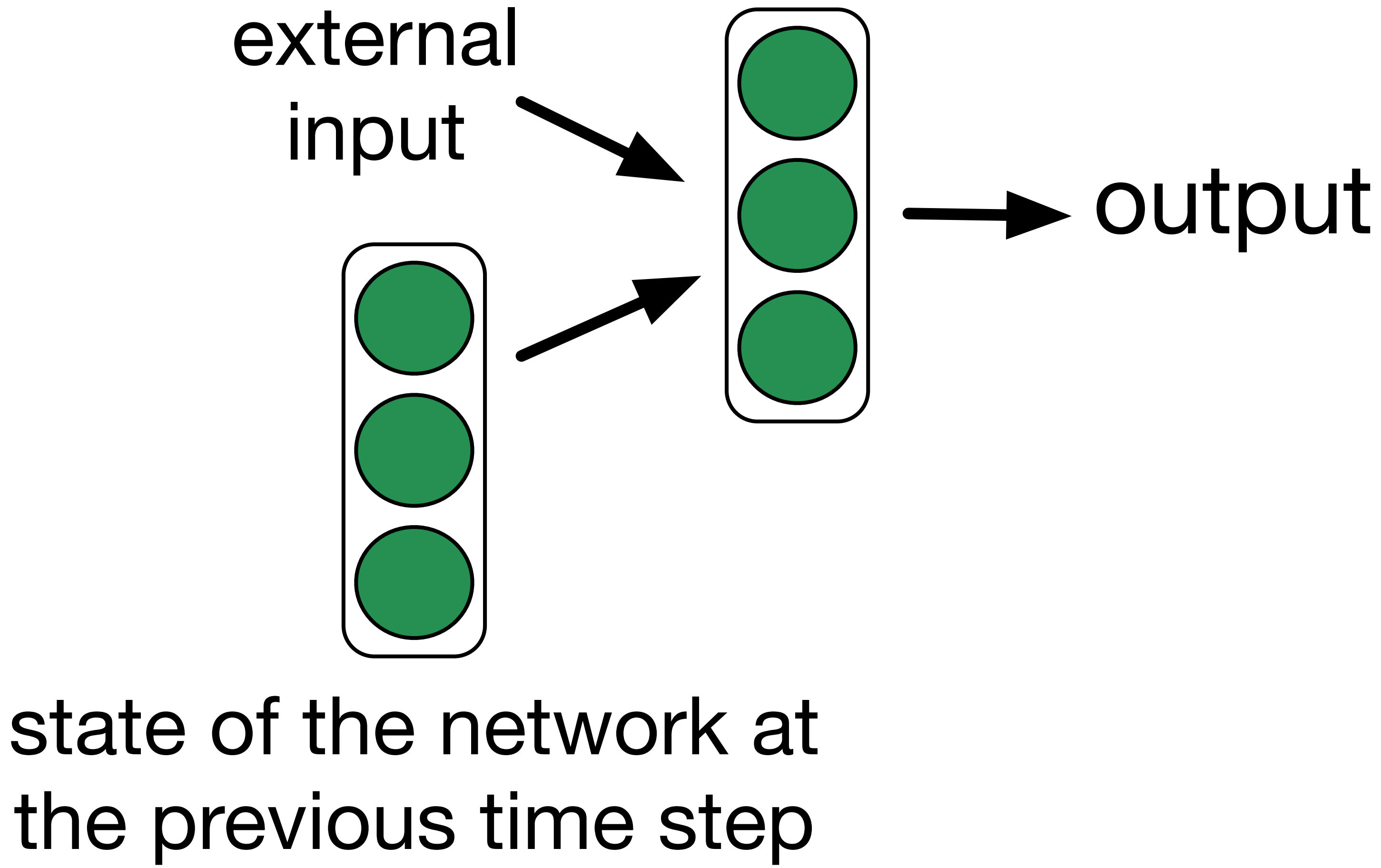
I: pictures of cars, O: voting preferences

…

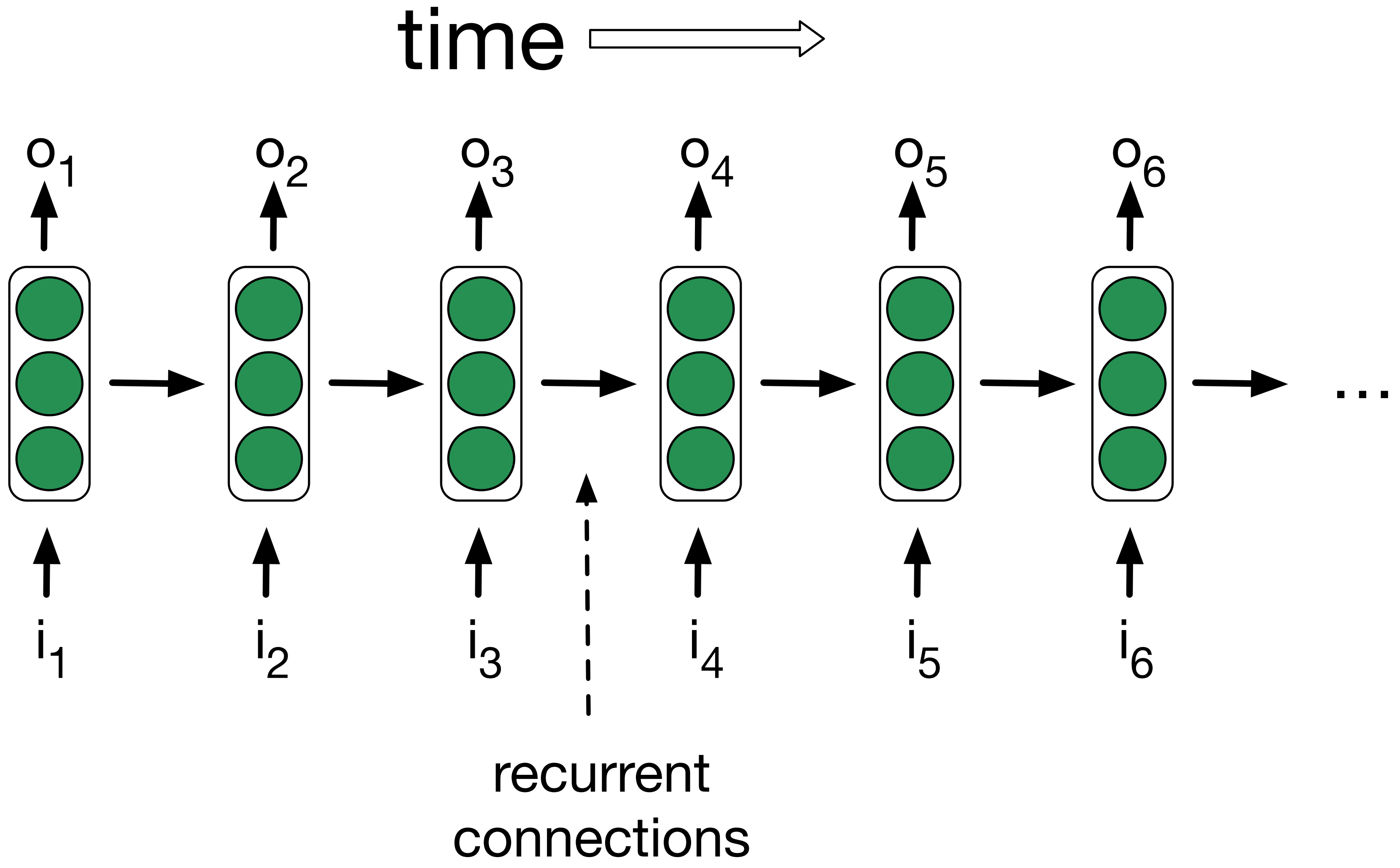
training agnostic to nature of input and output



Taking time into account with recurrent connections



Recurrent neural networks (RNNs)
The "unfolded" view



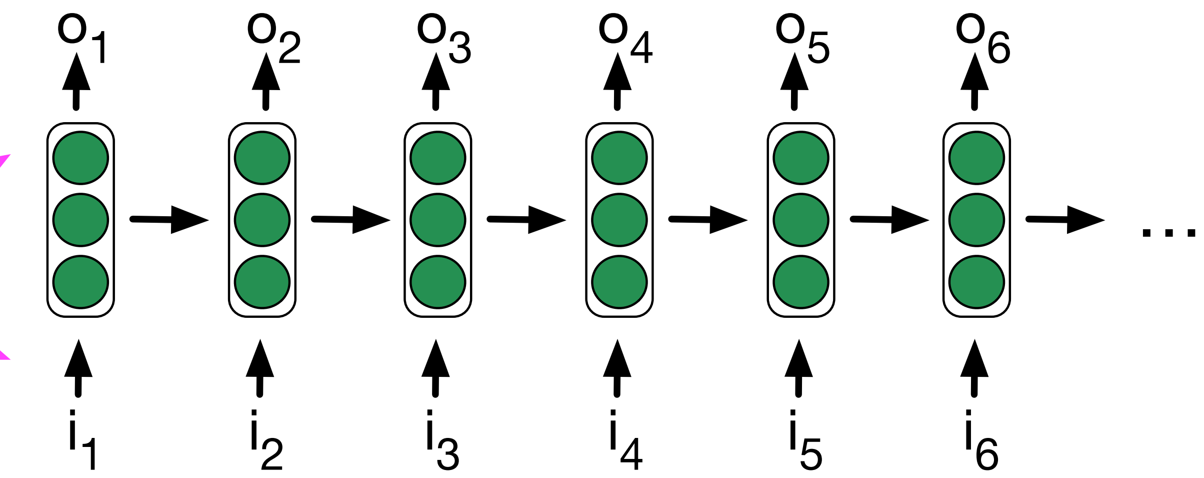
The unreasonable effectiveness of RNNs

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.

The generality of recurrent neural networks

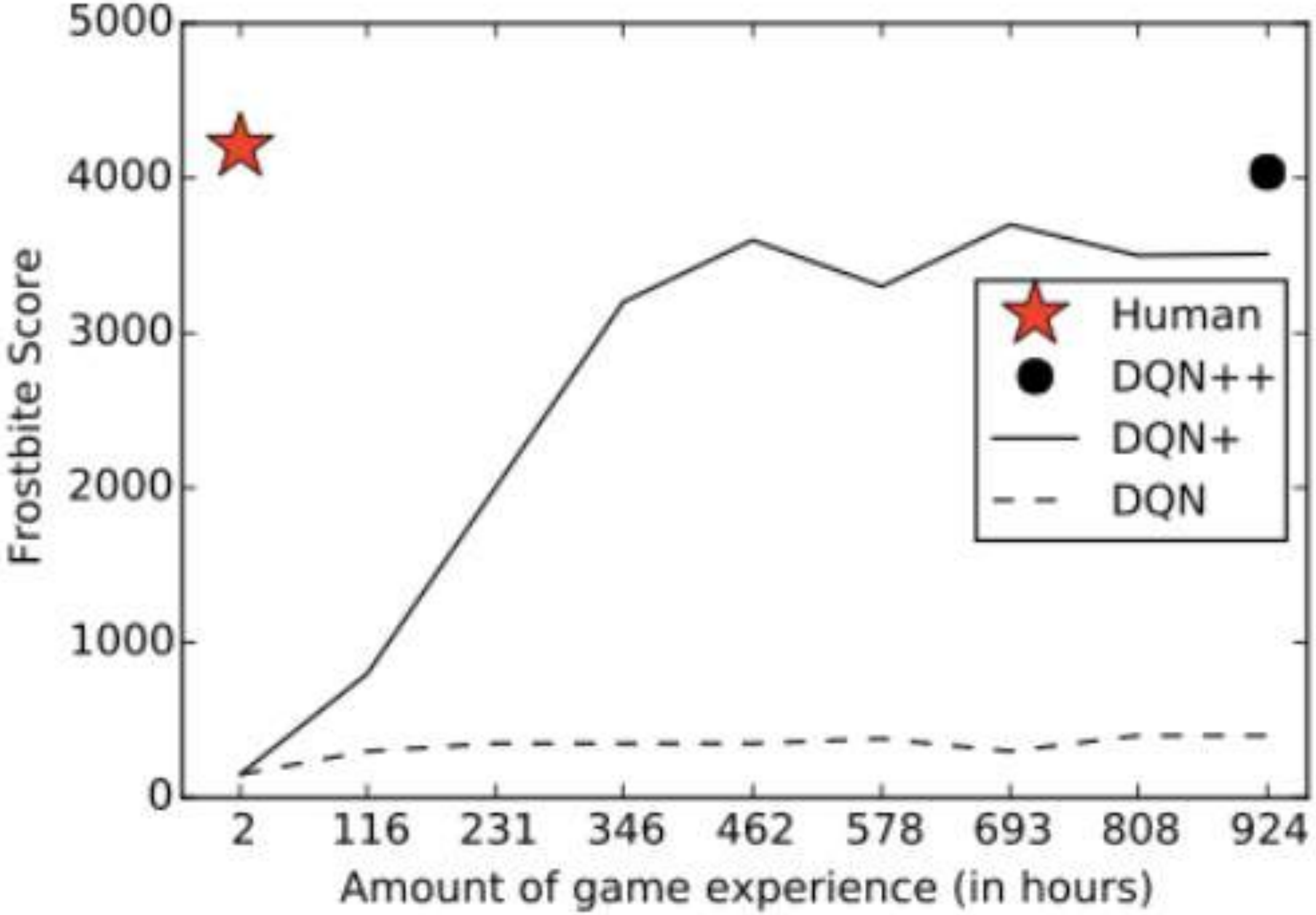
I: English sentences, O: French sentences
I: linguistic instructions, O: action sequences
I: video game states, O: next actions
…



On the verge of general machine intelligence?

- Although the final performance of these agents is impressive, **these techniques usually require several orders of magnitude more interactions with their environment than a human** in order to reach an equivalent level of expected performance. [Pritzel et al. 2017]
- People learning new concepts can often generalize successfully from just a single example, yet **machine learning algorithms typically require tens or hundreds of examples** to perform with similar accuracy. [Lake et al. 2018]
- Notably, **humans and large primates learn new knowledge even from limited experience**. [Shin et al. 2017]
- **Learning quickly is a hallmark of human intelligence**, whether it involves recognizing objects from a few examples or quickly learning new skills after just minutes of experience. [Finn et al. 2017]
- Notably, **performance in such tasks is typically evaluated after extensive, incremental training on large data sets. In contrast, many problems of interest require rapid inference from small quantities of data.** [Santoro et al. 2016]

On the verge of general machine intelligence?

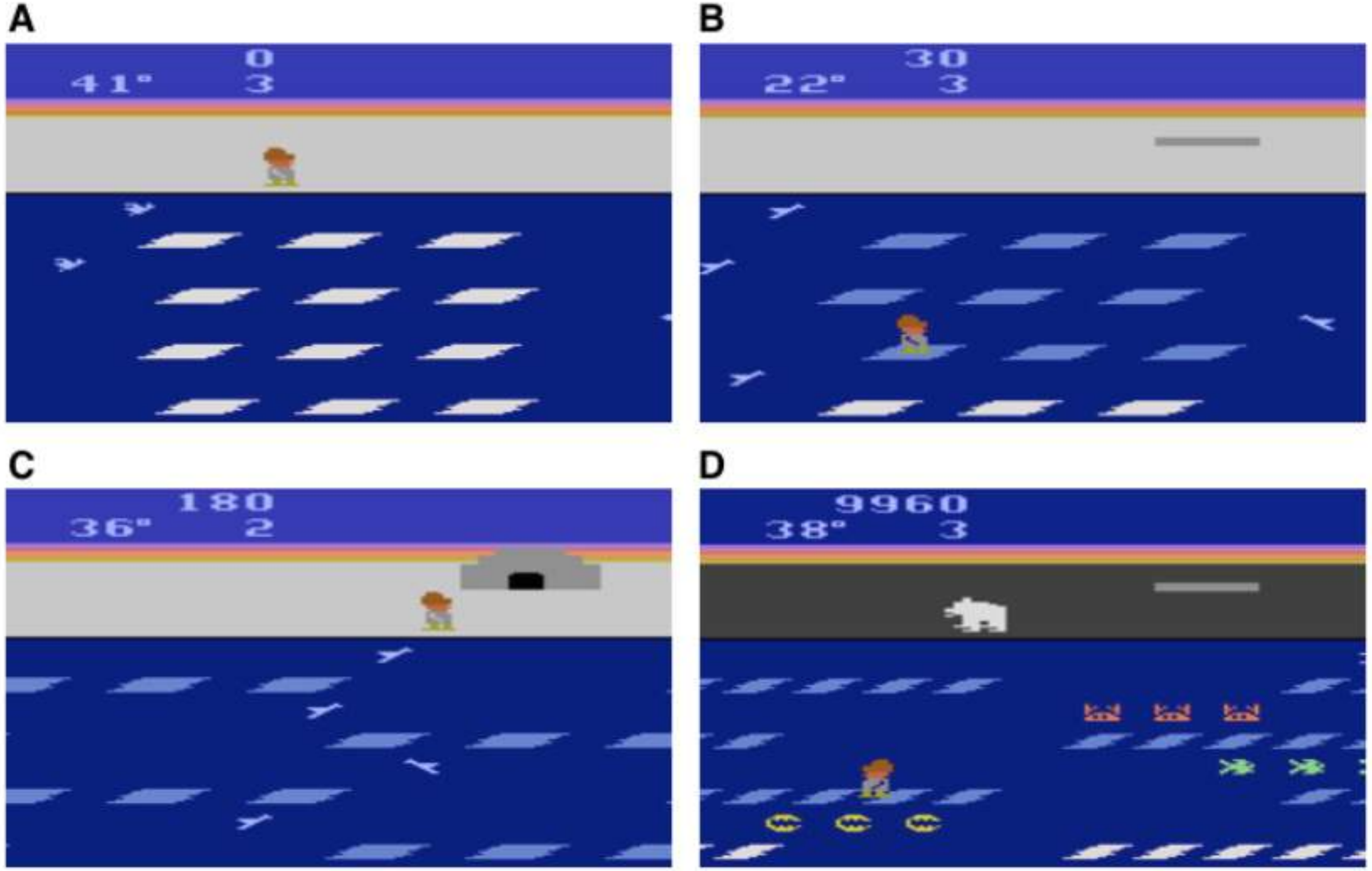


Lake et al. 2018

When are we humans fast at learning?

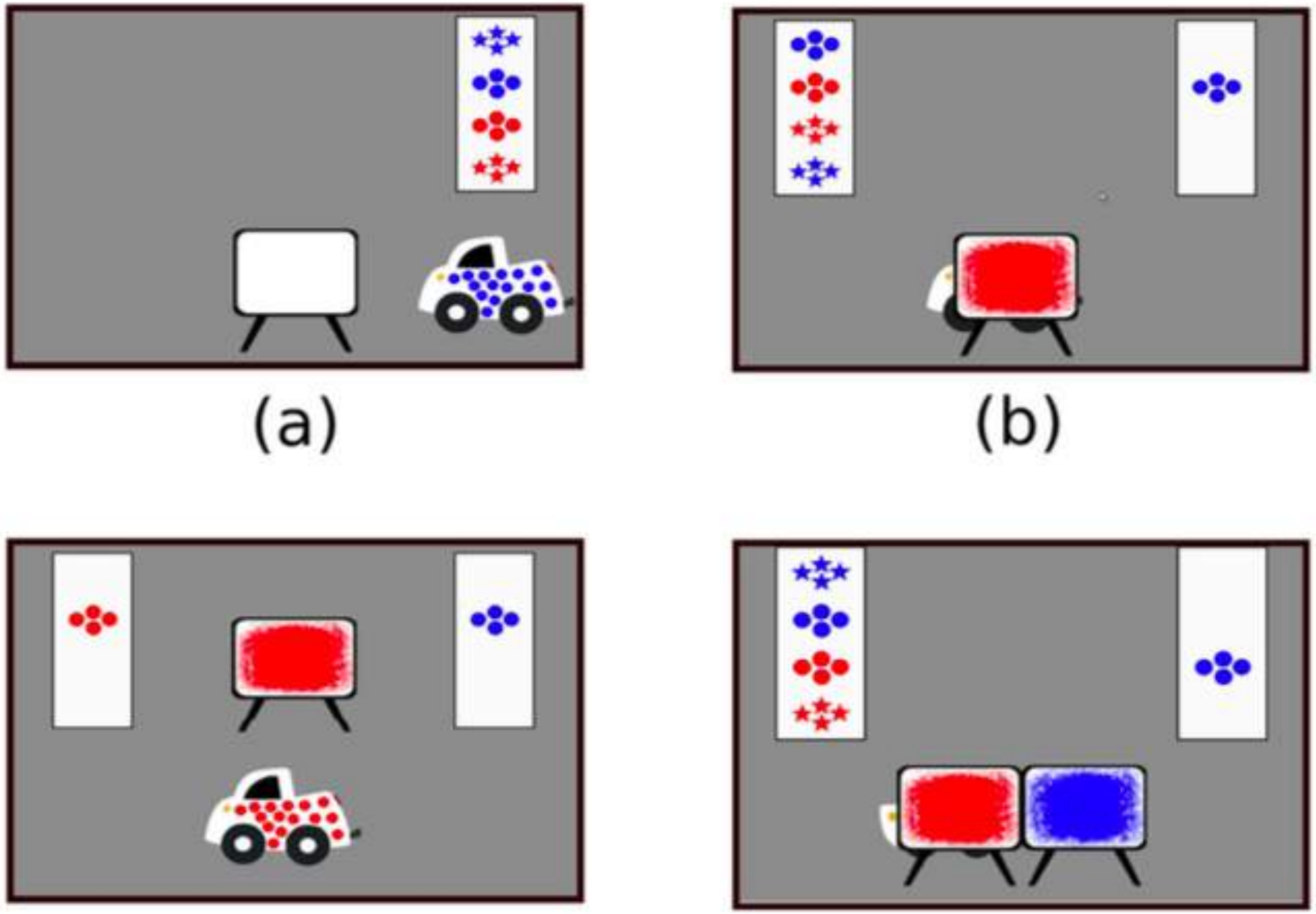
- When evolution has done the slow learning work for us
  - Perception and categorization, naïve physics and psychology, motor skills, core language faculties, reasoning...
- When new problems can be solved by combining old tricks (compositionality)

Prior knowledge and compositionality to play Frostbite



Lake et al. 2018

Compositional reasoning in 4-month olds



(a) (b)

(c) (d)

Piantadosi and Aslin 2016

Outline

- The (second) neural network coming
- **Systematic compositionality**
- A compositional challenge for neural networks
- (If time allows): Looking for a compositional neural network in a haystack

Systematic compositionality in language
Fodor and Pylyshyn 1988, Marcus 2003, 2018...

- Walk
- Walk twice
- Walk three times
- Run
- Run twice
- Run three times

Systematic compositionality in language

Fodor and Pylyshyn 1988, Marcus 2003, 2018...

- Walk
- Walk twice
- Walk three times
- Run
- Run twice
- Run three times
- Dax

Systematic compositionality in language

Fodor and Pylyshyn 1988, Marcus 2003, 2018...

- Walk
- Walk twice
- Walk three times
- Run
- Run twice
- Run three times
- Dax
- Dax twice
- Dax three times

Systematic compositionality in language

Fodor and Pylyshyn 1988, Marcus 2003, 2018...

- Walk
- Walk twice
- Walk three times
- Run
- Run twice
- Run three times
- Dax
- Dax twice
- Dax three times

[[X twice]] = [[X]][[X]]

[[X three times]] = [[X]][[X]][[X]]

[[dax]] = perform daxing action

Systematic compositionality in language

Fodor and Pylyshyn 1988, Marcus 2003, 2018...

- Walk
- Walk twice
- Walk three times
- Run
- Run twice
- Run three times
- Dax
- Dax twice three times

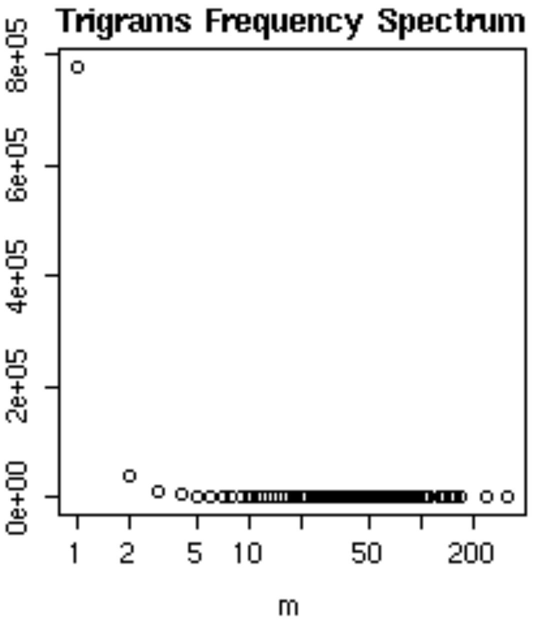
[[X twice]] = [[X]][[X]]

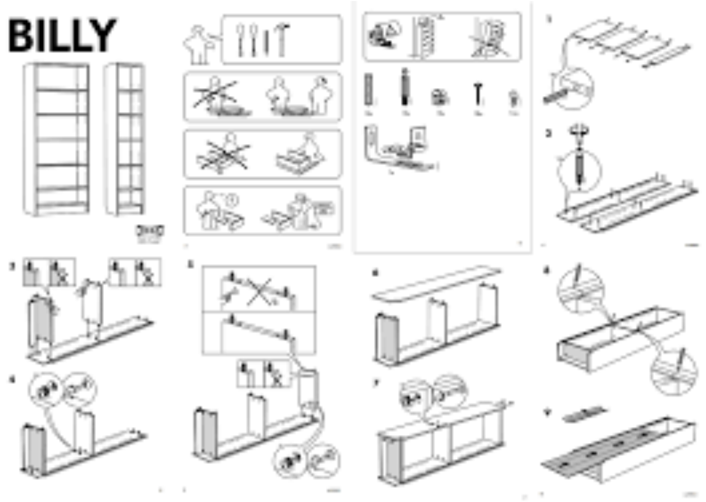
[[X three times]] = [[X]][[X]][[X]]

[[dax]] = perform daxing action

Systematic compositionality everywhere

A menacing crowd of rabid lizards surrounded the crimson castle.

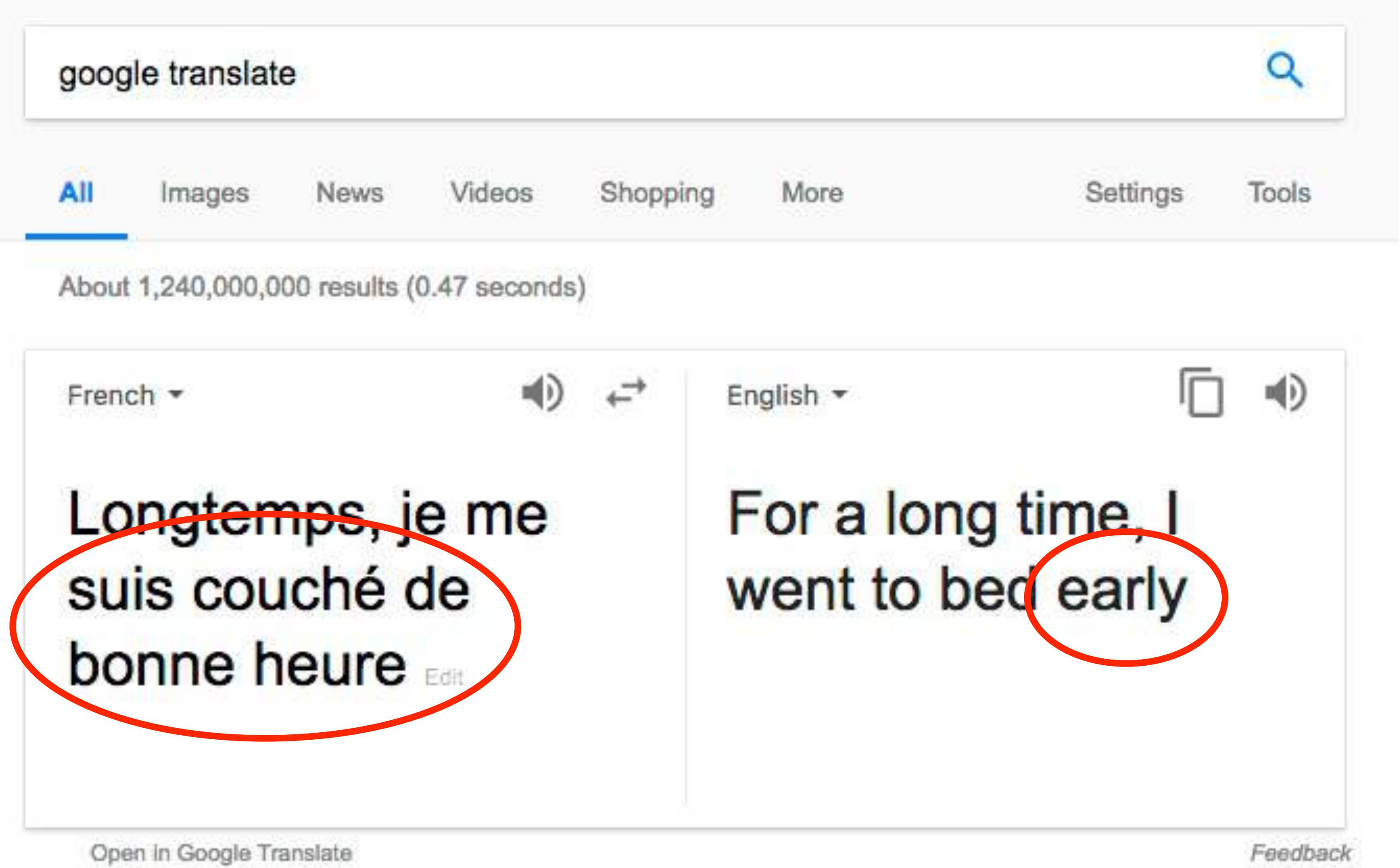




$(23+58)x(3-9) = ???$

Fregean compositionality is a special instance of systematic compositionality!

... but also non-systematic compositionality!



Outline

- The (second) neural network coming
- Systematic compositionality
- **A compositional challenge for neural networks**
- (If time allows): Looking for a compositional neural network in a haystack

Systematic compositionality in a simple grounded environment

In collaboration with Brenden Lake

Lots of earlier work on neural networks and systematicity, main novelty here is that we test latest-generation, state-of-the-art architectures!

- https://github.com/brendenlake/SCAN/
- https://arxiv.org/abs/1711.00350

Systematic compositionality in a simple grounded environment

walk and turn left!

Testing generalization

TRAINING PHASE

walk
WALK

jump after walk
WALK JUMP

walk and turn left
WALK LTURN

run thrice
RUN RUN RUN

run around
RUN RUN RUN RUN

look right and
walk left
RTURN LOOK
LTURN WALK

walk and run
RUN WALK

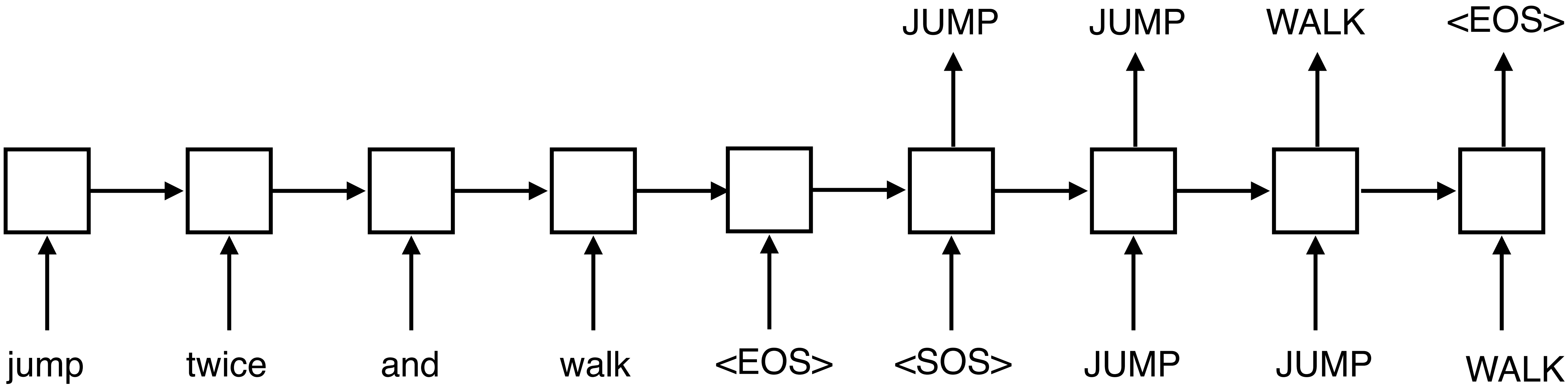
TEST TIME

jump around
and
turn left

The SCAN commands: examples

- Primitive commands:
  - run -> RUN
  - walk -> WALK
  - turn left -> LTURN
- Modifiers:
  - walk left -> LTURN WALK
  - run twice -> RUN RUN
- Conjunctions:
  - walk left and run twice -> LTURN WALK RUN RUN
  - run twice after walk left -> RUN RUN LTURN WALK
- Simplifications:
  - No scope ambiguity ("walk and [run twice]")
  - No recursion ("walk and run" vs *"walk and run and walk")

Sequence-to-sequence RNNs for SCAN



General methodology

- Train sequence-to-sequence RNN on 100k commands and corresponding action sequences
- At test time, only *new* composed commands presented
- Each test command presented once
- RNN must generate right action sequence at first try

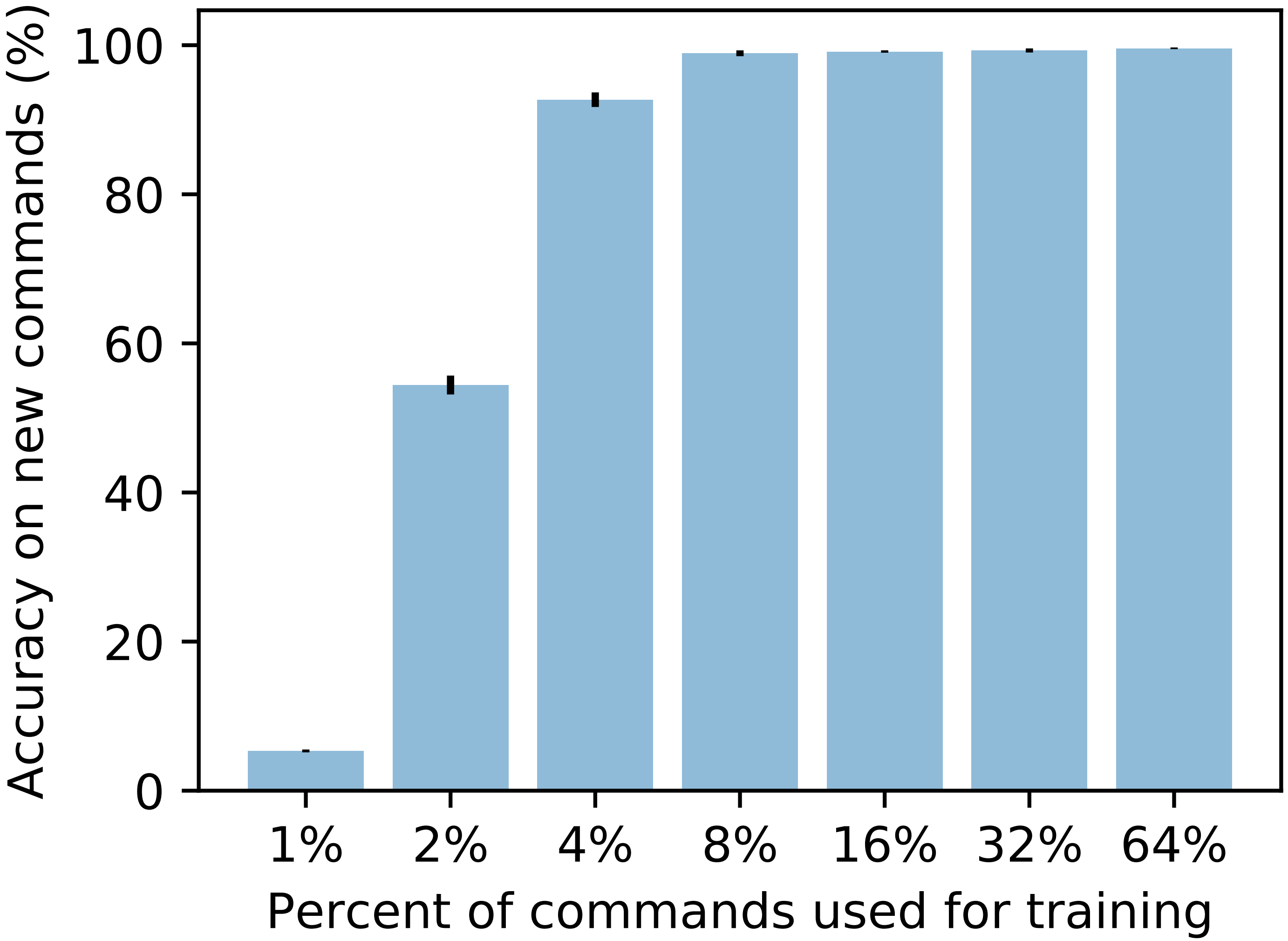
- Training details: ADAM optimization with 0.001 learning rate and 50% teacher forcing
- Best model overall:
  - 2-layer LSTM with 200 hidden units per layer, no attention, 0.5 dropout

Experiment 1: random train/test split

- Included in training tasks:
  - look around left twice
  - look around left twice and turn left
  - jump right twice
  - run twice and jump right twice

- Presented during testing:
  - look around left twice and jump right twice

Random train/test split results



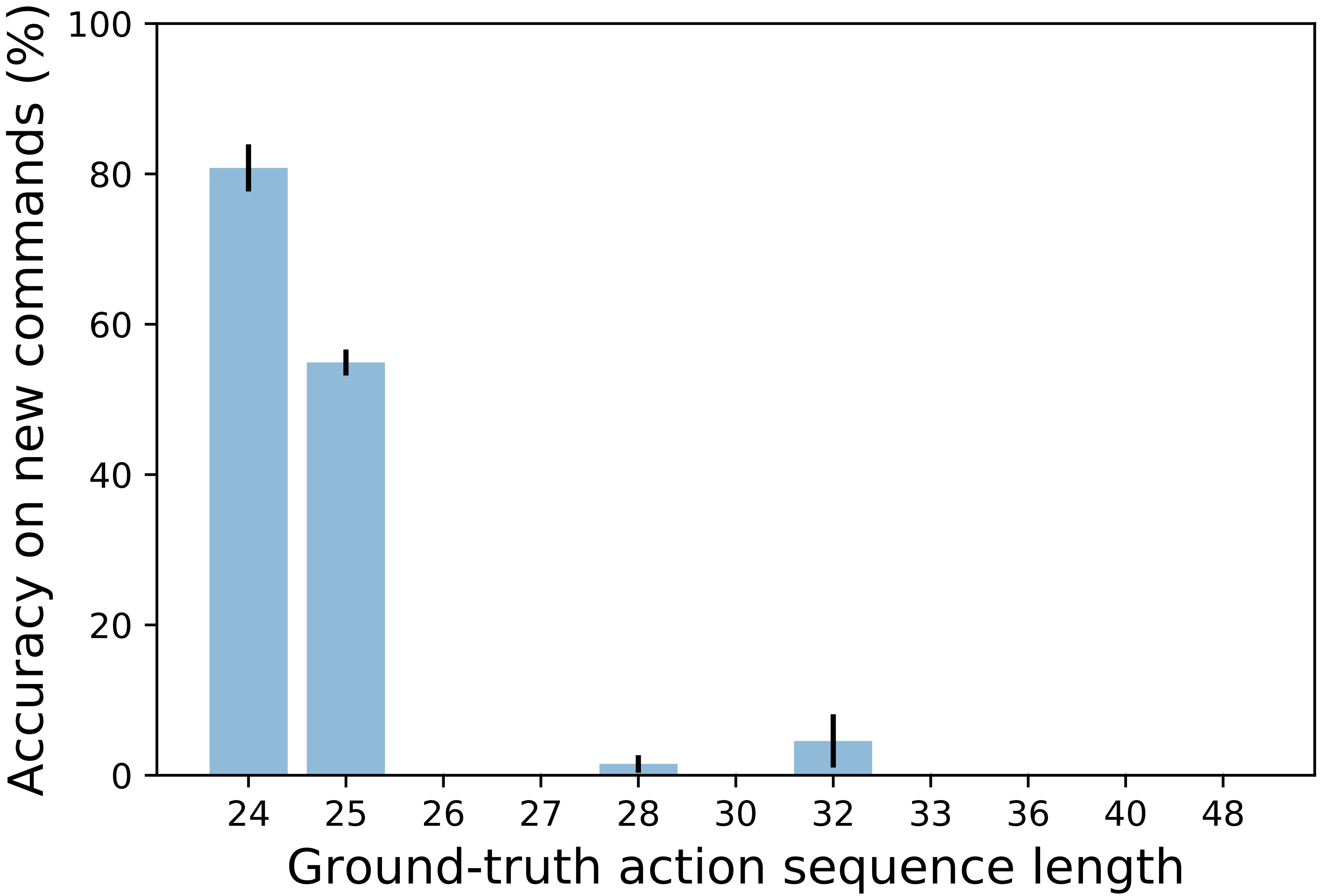
Experiment 2: split by action length

A grammar must reflect and explain the ability of a speaker to produce and understand new sentences which may be longer than any he has previously heard (Chomsky 1956)

- Train on commands requiring shorter action sequences (up to 22 actions)
  - jump around left twice (16 actions)
  - walk opposite right thrice (9 actions)
  - jump around left twice and walk opposite right twice (22 actions)

- Test on commands requiring longer actions sequences (from 24 to 48 actions)
  - jump around left twice and walk opposite right thrice (25 actions)

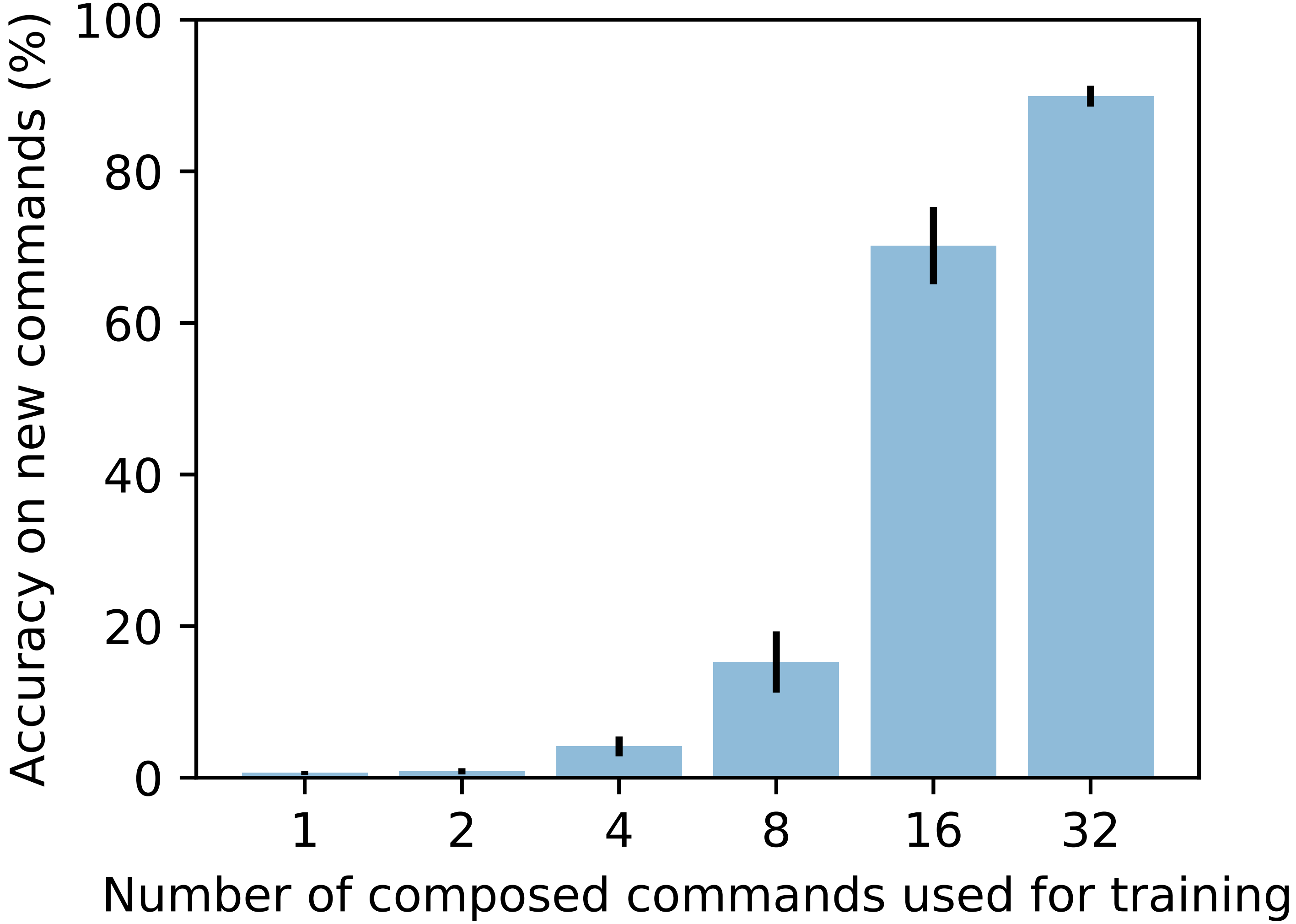
Length split results



Experiment 3: generalizing composition of a primitive command (the "dax" experiment)

- Training set contains all possible commands with "run", "walk", look", "turn left", "turn right":
  - "run", "run twice", "turn left and run opposite thrice", "walk after run", ...
- *but only a small set of composed "jump" commands:*
  - "jump", "jump left", "run and jump", "jump around twice"
- System tested on all remaining "jump" commands:
  - jump twice
  - jump left and run opposite thrice
  - walk after jump
  - ...

Composed-"jump" split results



Proof-of-concept replication in Machine Translation

- Training: 100k sentences including:
  - I am **daxy** -> je suis **daxiste**
  - ... and many more simple sentences illustrating the paradigm below with other adjectives
- Test set includes:
  - you are **daxy** -> tu es **daxiste**
  - he is **daxy** -> il est **daxiste**
  - I am not **daxy** -> je ne suis pas **daxiste**
  - you are not **daxy** -> tu n'es pas **daxiste**
  - he is not **daxy** -> il n'est pas **daxiste**
  - I am very **daxy** -> je suis très **daxiste**
  - you are very **daxy** -> tu es très **daxiste**
  - he is very **daxy** -> il est très **daxiste**

Proof-of-concept replication in Machine Translation

- Out best RNN model gets only 1/8 daxy translation right ("he is daxy")

- For comparison:
  - "tired" occurred in 80 separate constructions in training
  - Model correctly translated equivalent "tired" sentences with 8/8 accuracy

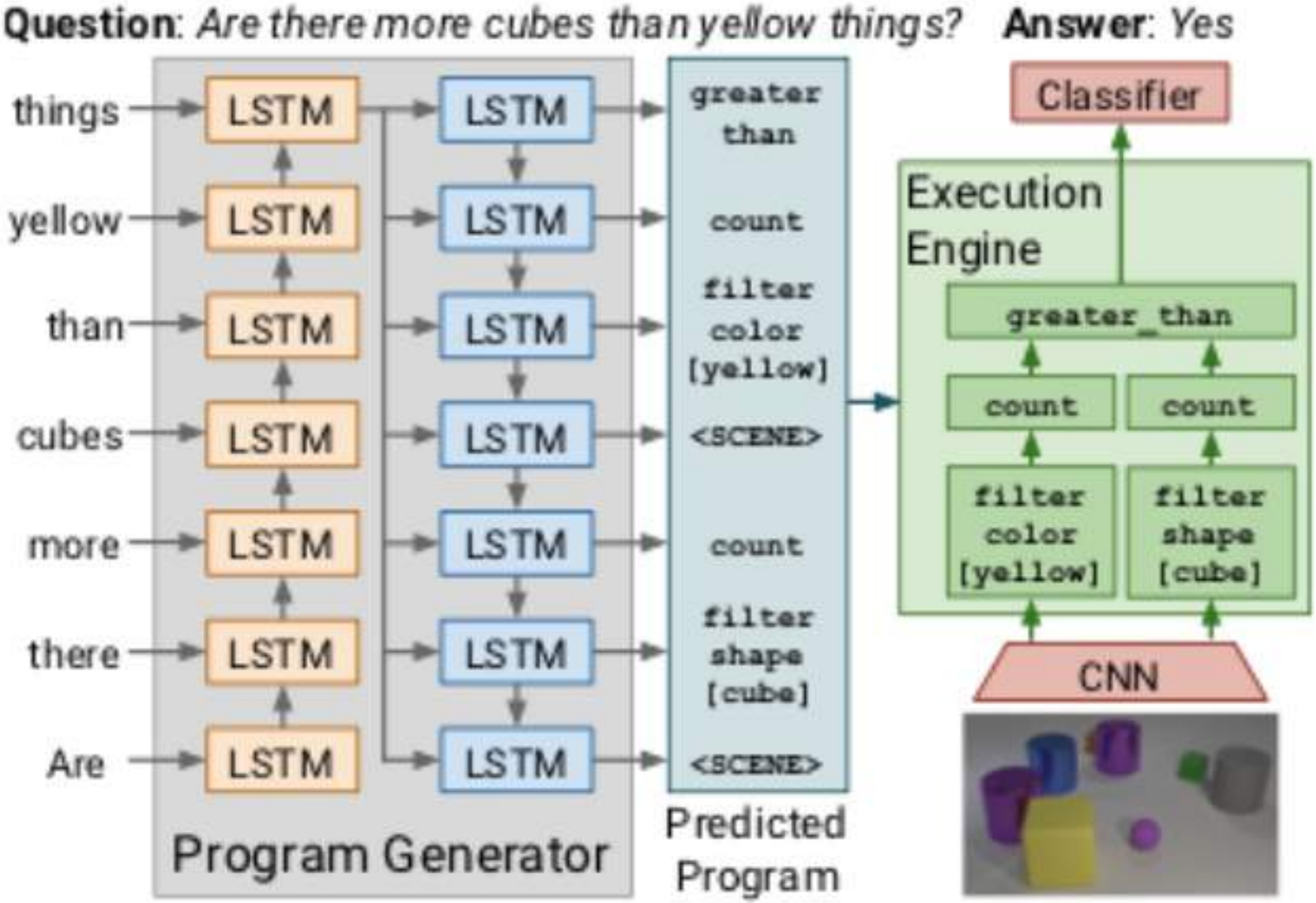
Ad-interim conclusion

- State-of-the-art "Seq2Seq" Recurrent Neural Networks achieve considerable degree of generalization (Exp 1)...
- ... but this generalization does not appear to be "systematic" in the Fodorian sense (Exp 2, Exp3, MT pilot)
- Are there conditions under which we can teach Neural Networks to generalize compositionally?

Outline

- The (second) neural network coming
- Systematic compositionality
- A compositional challenge for neural networks
- **Looking for a compositional neural network in a haystack**

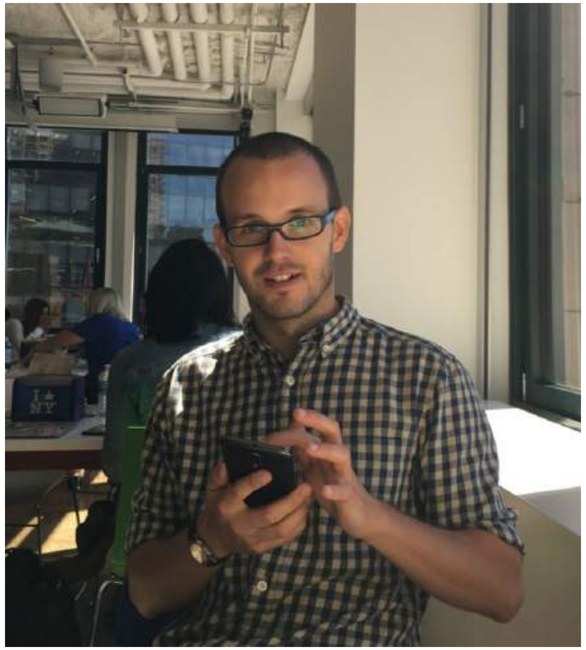
Making neural networks compositional... at the expense of generality



Johnson et al. 2017

Can a generic RNN learn to behave compositionally?

Work in progress with Adam Liska and Germán Kruszewski



The table lookup domain

t1
00 → 10
01 → 11
10 → 01
11 → 00

t2
00 → 11
01 → 00
10 → 01
11 → 10

t3
00 → 00
01 → 01
10 → 10
11 → 11

t4
00 → 11
01 → 10
10 → 00
11 → 01

t5
00 → 10
01 → 00
10 → 01
11 → 11

…

t1(00)=10
t3(00)=00

t4(t5(01))=11
t5(t4(01))=01
t2(t2(10))=00

t1(t4(t5(11)))=11
t1(t5(t1(10)))=10

thanks to Angeliki Lazaridou and José Hernandez-Orallo for the lookup task idea!

The table lookup domain

t1

00 ⇒	10
01 ⇒	11
10 ⇒	01
11 ⇒	00

t2

00 ⇒	11
01 ⇒	00
10 ⇒	01
11 ⇒	10

t3

00 ⇒	00
01 ⇒	01
10 ⇒	10
11 ⇒	11

t4

00 ⇒	11
01 ⇒	10
10 ⇒	00
11 ⇒	01

t5

00 ⇒	10
01 ⇒	00
10 ⇒	01
11 ⇒	11

...

t1(00)=10
t3(00)=00

nothing smart about primitive lookup learning: tables can only be memorized

t4(t5(01))=11
t5(t4(01))=01
t2(t2(10))=00

infinite expressions by finite means

t1(t4(t5(11)))=11
t1(t5(t1(10)))=10

Testing compositional generalization

t1	t2	t3	t4	t5	...
00 ⇨ 10	00 ⇨ 11	00 ⇨ 00	00 ⇨ 11	00 ⇨ 10	
01 ⇨ 11	01 ⇨ 00	01 ⇨ 01	01 ⇨ 10	01 ⇨ 00	
10 ⇨ 01	10 ⇨ 01	10 ⇨ 10	10 ⇨ 00	10 ⇨ 01	
11 ⇨ 00	11 ⇨ 10	11 ⇨ 11	11 ⇨ 01	11 ⇨ 11	

red = must be generated by RNN

Training phase #1: simple lookups

t1:00.**10.** t4:10.**00.** t301.**01.** ...

Training phase #2: simple and composed lookups

ct1t4:00:**00.** t3:10.**10.** ct5t5:01.**10.** ...

Test phase: composed lookups seen during training, with **novel** inputs:

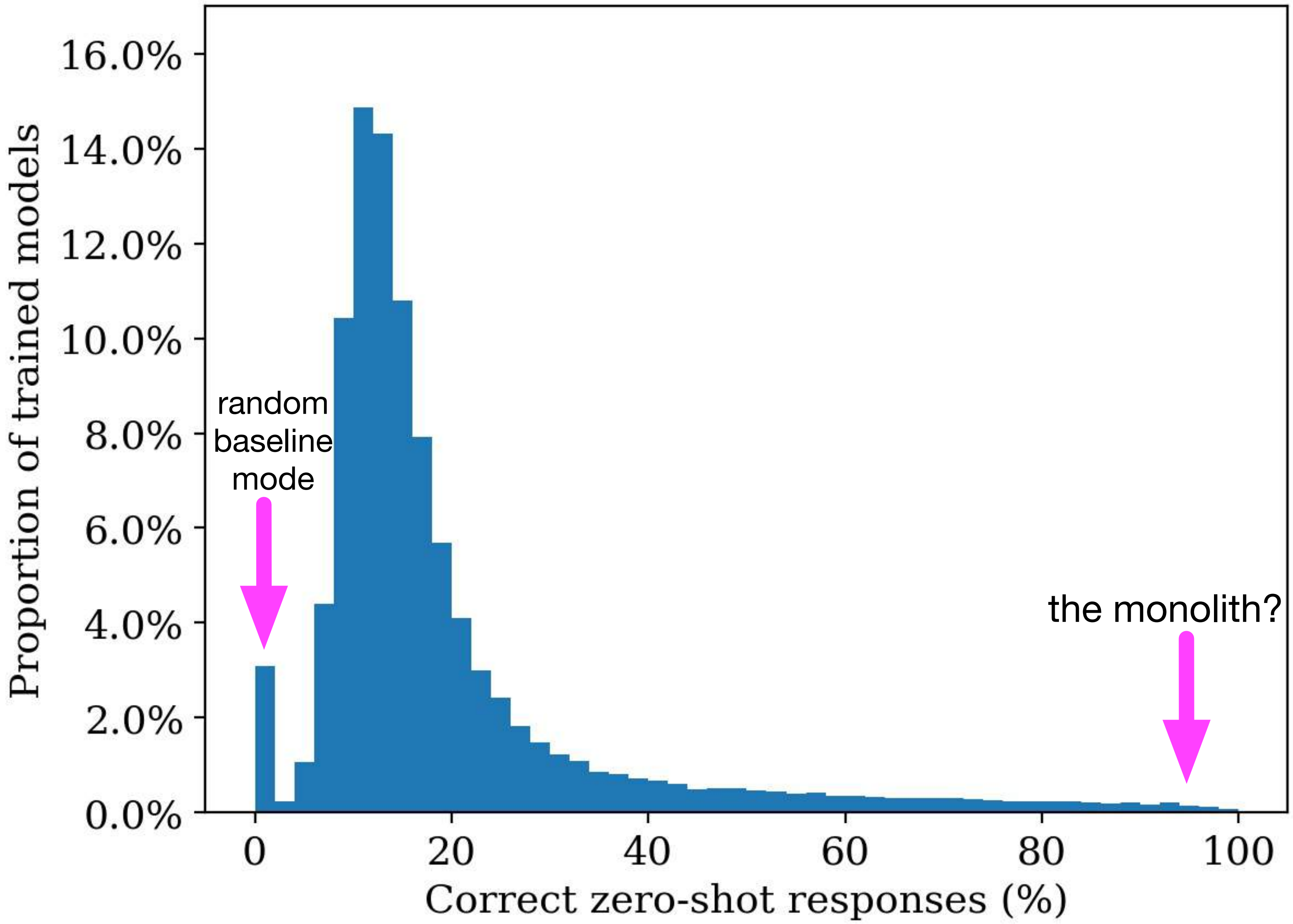
ct1t4:01:**01.** ct5t5:00.**01.** ct3t2:10.**01.**

figure of merit:
0-shot accuracy

Experimental setup

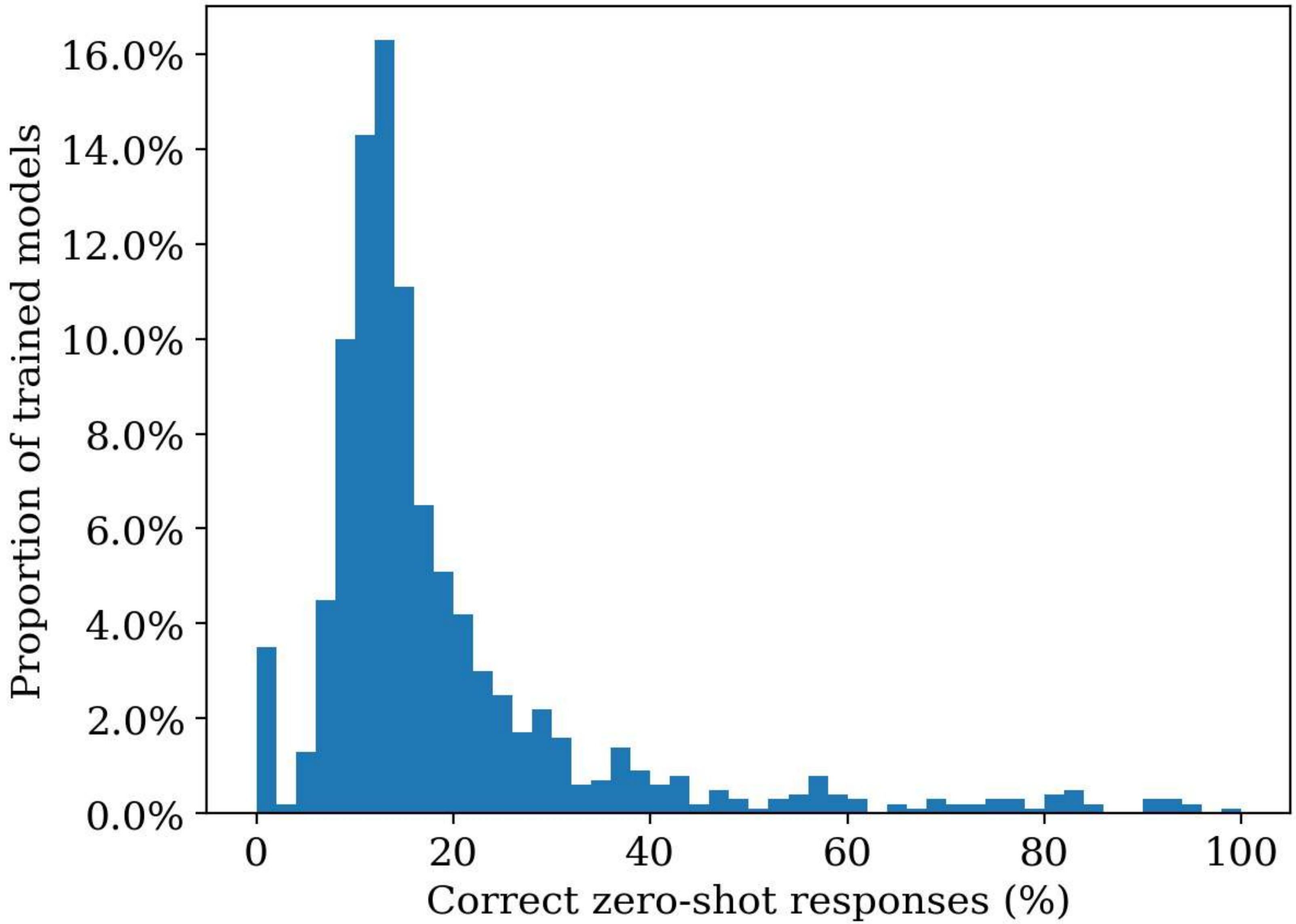
- Recurrent network with two hidden layers
  - Recurrent 60-unit LSTM layer
  - 10-unit sigmoid layer
  - **This architecture *can theoretically* encode a compositional solution**
- Model reads instructions and produces output character-by-character
  - RNN's own output at *t-1* also fed with input at *t*
- Experimenting with 3-bit tables, first-order composition only:
  - 1M examples in training phases #1 and #2
  - 128 inputs left-out for testing (2 per possible first-order table composition)
- Standard training: backpropagate cross-entropy losss and update parameters with stochastic gradient descent (parallel updates from 40 CPUs)
- Experiment repeated 50k times from random initializations
  - From uniform [-0.1, 0.1] range

Looking for a compositional RNN in a haystack



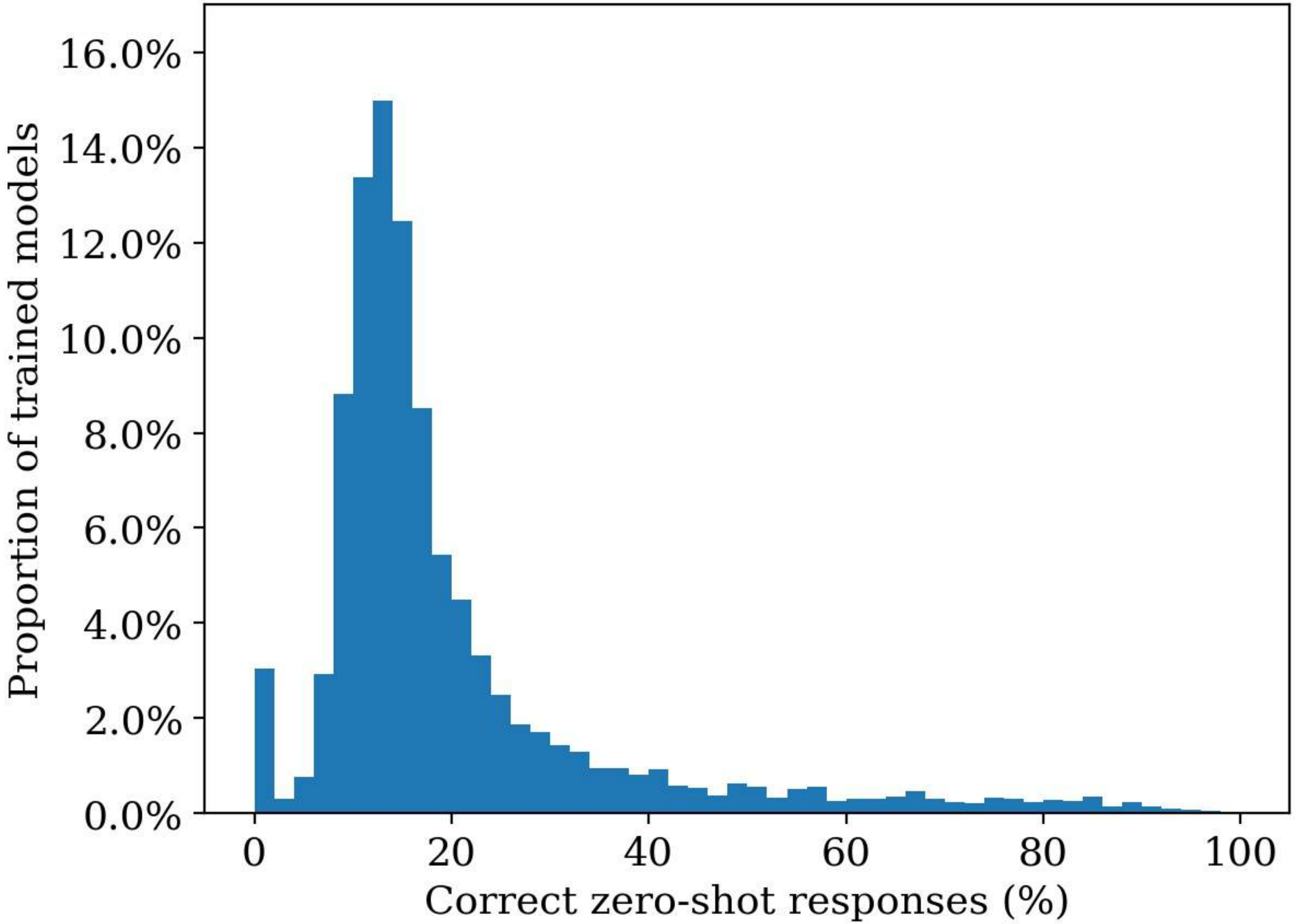
The compositional RNN in a haystack

Same initialization, different runs



The compositional RNN in a haystack

Making the prompts opaque



e.g., composition of t1 and t2 is denoted by ct5t4 instead of ct1t2

no sign of Fregean compositionality!

Conclusion 1

- (Recurrent) neural networks are remarkably powerful and general
  - Agnostic "end-to-end" learners from input-output pairs
- They can generalize to new inputs that are different from those they were trained on...
- ... but their generalization skills do not display **systematic compositionality**
  - Thus, they cannot adapt fast to continuous stream of new inputs in domains such as language, math, and more generally reasoning

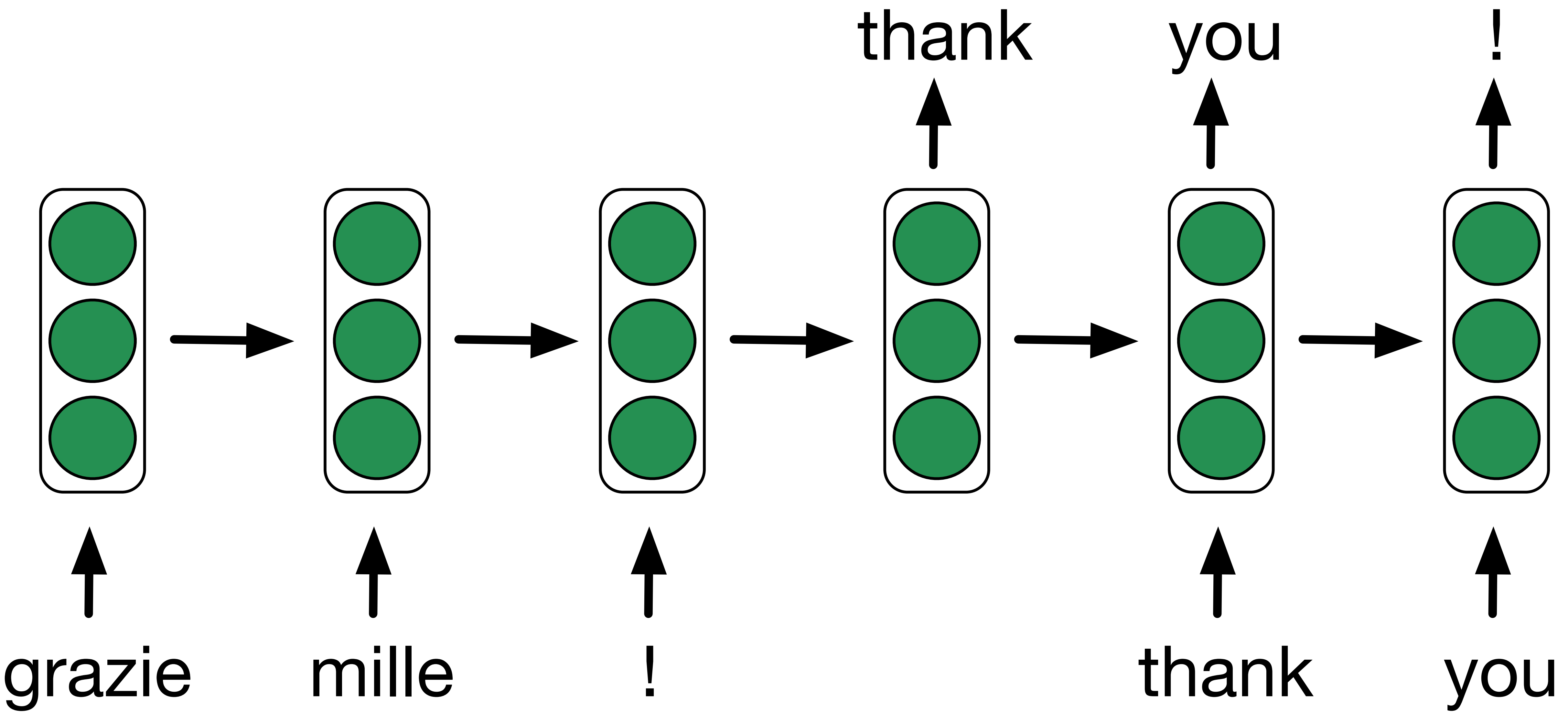
Conclusion 2

- We could hard-code compositionality into neural network architectures...
- ... but this might dramatically affect their generality and effectiveness
  - Each new domain will require a new hand-coded set of modules and composition rules
  - Generic (recurrent) neural networks are still the workhorse of successful deep learning applications
- General RNN architecture can learn to encode compositional solutions
- ... but standard training methods do not easily converge to such solutions

Conclusion 3

- Given a sufficiently diverse environment where fast generalization plays a crucial role...

- could compositional neural networks be naturally selected by evolutionary pressures?
  - What's the right environment?
  - How can we speed up *evolution*?


thank
you
!
grazie
mille
!
thank
you