Learning by Abstraction: The Neural State Machine

Nikolaeva Sofya 171

2021

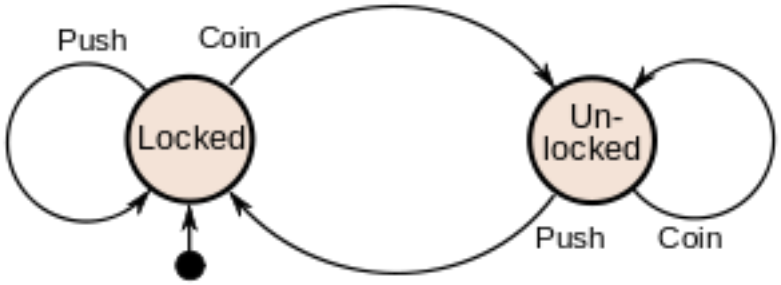
The Neural State Machine

The Neural State Machine (NSM) is a graph-based network that simulates the computation of a finite automaton, and is explored here in the context of VQA, where we are given an image and a question and asked to provide an answer.

State machine

In simple terms, a **state machine** is a computational model that consists of a collection of states, which it iteratively traverses while reading a sequence of inputs, as determined by a transition function.

State diagram for a turnstile



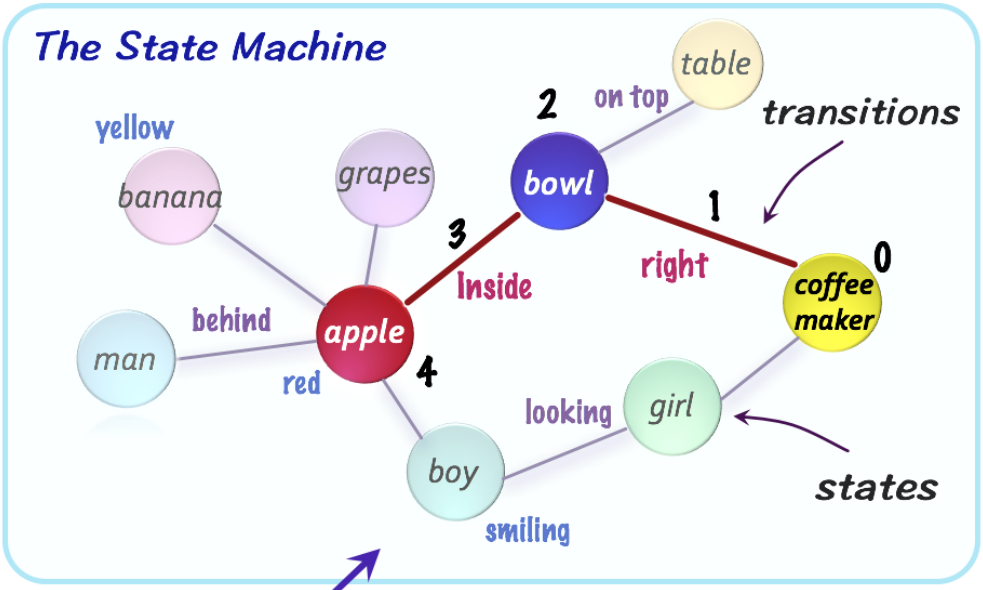
Current State	Input	Next State	Output
Locked	coin	Unlocked	Unlocks the turnstile so that the customer can push through.
	push	Locked	None
Unlocked	coin	Unlocked	None
	push	Locked	When the customer has pushed through, locks the turnstile.

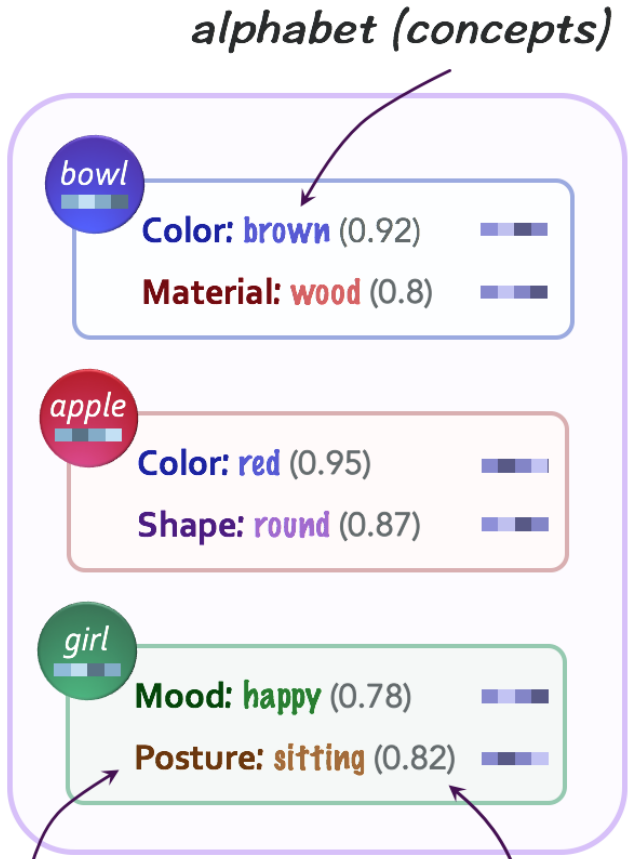
The Neural State Machine

The Neural State Machine (NSM) is a graph-based network that simulates the computation of a finite automaton, and is explored here in the context of VQA, where we are given an image and a question and asked to provide an answer.

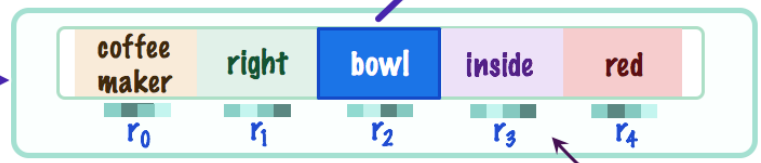
We go through two stages – **modeling** and **inference**, the first to construct the state machine, and the second to simulate its operation.

The Neural State Machine





What is the red fruit inside the bowl to the right of the coffee maker?



apple

$$(C, S, E, \{r_i\}_{\{i=0\}}^{N}, p_0, \delta)$$

- C - the model's alphabet, consisting of a set of concepts
- S - a collection of states
- E - a collection of directed edges that specify valid transitions between the states
- r_i - a sequence of instructions, each of dimension d, that are passed in turn as an input to the transition function δ.
- p_0 : S → [0, 1] a probability distribution of the initial state.
- $\delta_{\{S,E\}} : p_i \times r_i \rightarrow p_{\{i+1\}}$ a state transition function

Concept vocabulary - C

- An embedded concept vocabulary C (initialized with GloVe) will be used to capture and represent the semantic content of input images.
- Concept vocabulary consist of 785 objects, 170 relations, and 303 attributes that are divided into 77 types.

- The vocabulary is grouped into L + 2 *properties* such as
- **object** identity $C_O = C_0$,
- different types of **attributes** $C_A = \cup_{i=1}^{L} C_i$,
- **relations** $C_R = C_{L+1}$.

States and edge transitions

- Graph consists of:

1. A set of object nodes S from the image, each accompanied by a bounding box, a mask, dense visual features, and a collection of discrete probability distributions $\{P_i\}_{\{i=0\}}^{L}$ for each of the object's L + 1 semantic properties, defined over the concept vocabulary $\{C_i\}_{\{i=0\}}^{L}$ presented above;
2. A set of relation edges between the objects, each associated with a probability distribution P_{L+1} of its semantic type (e.g. on top of, eating) among the concepts in C_{L+1}, and corresponding to a valid transition between the machine's states.

States and edge transitions

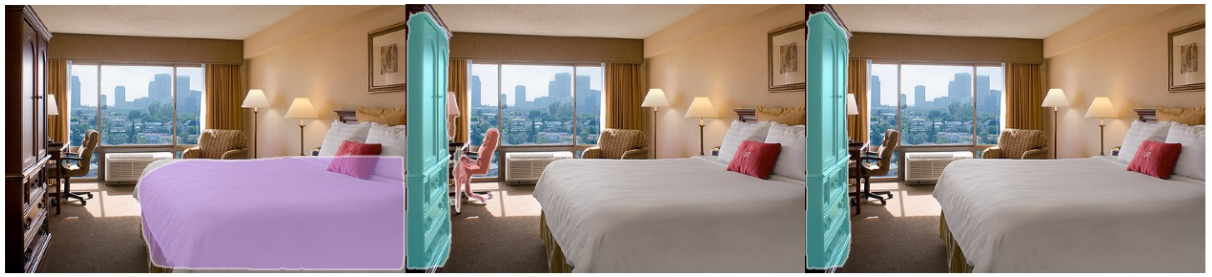
- For each state s ∈ S that corresponds to an object in the scene, we define a set of L + 1 property variables $\{s_j\}_{\{j=0\}}^{L}$ and assign each of them with

$$s^j = \sum c_k \in C_j \;\; P_j(k) c_k$$

- Edge representations are computed in a similar manner, resulting in matching embeddings of their relation type:

$$e' = \sum c_k \in C_{L+1} \;\; P_{L+1}(k) c_k$$

for each edge e ∈ E.



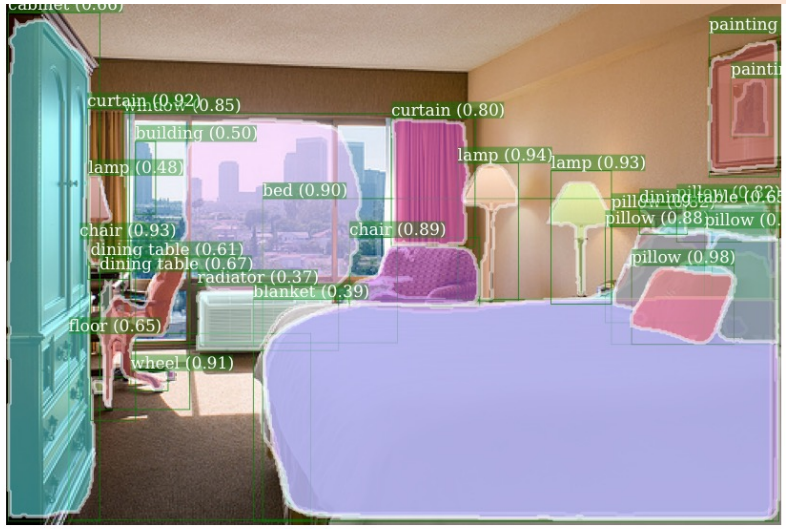
bed → left → tall → made

What is the **tall object** to the **left** of the **bed** made of?

Cabinet: wood (0.95), tall (0.92), shiny (0.86)
Bed: white (0.84), comfortable (0.91)
Lamp: yellow (0.92), on (0.74), thin (0.82)

(cabinet, *left*, bed) (0.82)
(pillow, *on*, bed) (0.74)
...

Wood



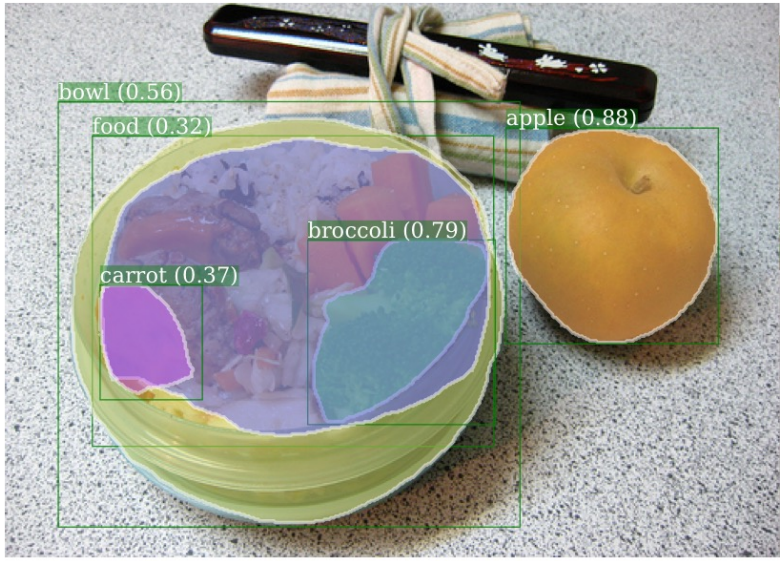
bowl → inside → green → what

What is the **green food** **inside** of the **bowl**?

Apple: yellow (0.58), round (0.95), healthy (0.91)
Broccoli: green (0.94), leafy (0.93), fresh (0.92)
Bowl: plastic (0.72), transparent (0.84)

(broccoli, *inside*, bowl) (0.68)
(bowl, *left*, apple) (0.85)
...

Broccoli



Reasoning instructions

- Now we translate the question into a sequence of reasoning instructions.
- We begin by embedding all the question words using GloVe (dimension d = 300).
- For each embedded word w_i we compute a similarity-based distribution

$$\boldsymbol{P_i} = \boldsymbol{softmax}(\boldsymbol{w_i^T WC})$$

  W - identity matrix, **C** - the matrix of all embedded concepts + an additional learned default embedding **c′** to account for structural or other non-content words.

- Next, we translate each word into a concept-based representation:

$$\boldsymbol{v_i} = \boldsymbol{P_i}(\boldsymbol{c'})\boldsymbol{w_i} + \sum_{\boldsymbol{c}\in \boldsymbol{C}\setminus\{\boldsymbol{c'}\}} \boldsymbol{P_i}(\boldsymbol{c})\mathbf{c}$$

- Finally, we process the normalized question words with an attention-based encoder-decoder:

$$r_i = softmax(h_i V^T)V, \qquad V^{P\times d} = \{v_i\}_{i=1}^{\mathrm{P}}$$

Model simulation

- Basically, we will begin with a uniform initial distribution p_0 over the states, and at each reasoning step i, read an instruction r_i, and use it to redistribute our attention over the states by shifting probability along the edges.
- Implementing a neural module for the state transition function $\delta_{\{S,E\}} : p_i \times r_i \rightarrow p_{\{i+1\}}$
- At each step i, the module takes a distribution p_i over the states as an input and computes an updated distribution p_{i+1} , guided by the instruction r_i .
- **Our goal** is to determine what next states to traverse to (p_{i+1}) based on the states we are currently attending to (p_i).
- **Our first goal** is thus to find the instruction type that is most relevant to the instruction r_i . We compute the distribution R_i = softmax($r_i^T \circ$ D) over the L + 2 embedded properties D.
- We further denote R_i (L + 1) ∈ [0, 1] that corresponds to the relation property as r_i' , measuring the degree to which that reasoning instruction is concerned with semantic relations.

Model simulation

- We compare the instruction to all the states s ∈ S and edges e ∈ E, computing for each of them a relevance score:

$$\gamma_i(s) = \sigma(\sum_{j=0}^{L} R_i(j)\ (r_i \circ W_j\ s^j\ )\ )$$

$$\gamma_i(e) = \sigma(r_i \circ W_{L+1}\ e')$$

- Achieving **the key goal** of this section: shifting the model's attention pi from the current nodes (states) s ∈ S to their most relevant neighbors – the next states:

$$p_{i+1}^{s} = softmax_{s \in S}(W_s \cdot \gamma_i(s))$$

$$p_{i+1}^{r} = softmax_{s \in S}(\sum_{(s',s) \in \mathrm{E}} p_i(s') \cdot \gamma_i((s',s)))$$

$$p_{i+1} = r_i' \cdot p_{i+1}^{r} + (1 - r_i') \cdot p_{i+1}^{s}$$

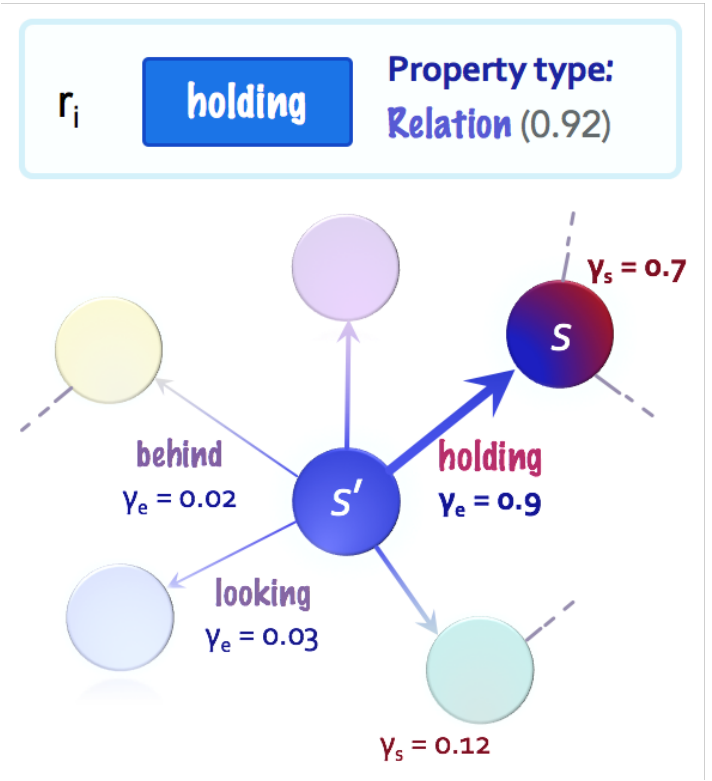
p_{i+1} - the distribution over the next states , p_{i+1}^{s} - probability based on each potential next state's own internal properties, p_{i+1}^{r} - probability, that considers the next states contextual relevance, relative to the current states the model attends to.

Model simulation

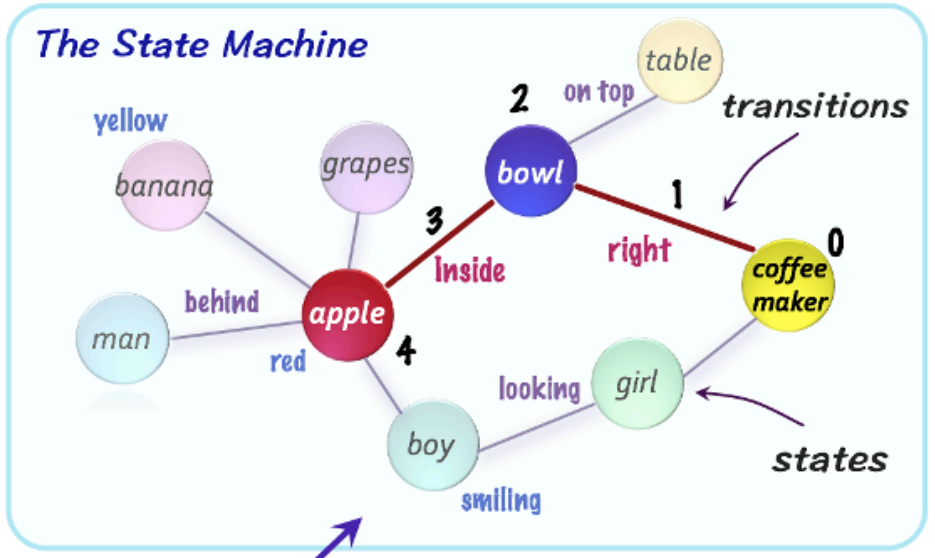
- After completing the final computation step, and in order to predict an answer, we use a standard 2-layer fully-connected softmax classifier that receives the concatenation of the question vector q as well as an additional vector m that aggregates information from the machine's final states:

$$m = \sum_{s \in S} p_N(s) \left(\sum_{j=0}^{L} R_N(j) \cdot s^j \right)$$

- Where m reflects the information extracted from the final states as guided by the final reasoning instruction r_N : averaged first by the reasoning instruction type, and then by the attention over the final states, as specified by p_N .



Model simulation



What is the red fruit inside the bowl to the right of the coffee maker?

apple

instructions

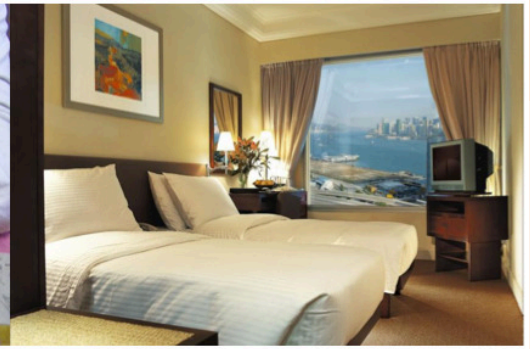
pro

Question examples along with answers predicted by the NSM.

1) What is the **giraffe** looking at? ***person*** ✓
2) Is the **fence** in front of the **giraffe** made of metal? ***no*** ✓
3) Is the **woman**'s **shirt** blue or yellow? ***blue*** ✓
4) On which side of the image is the **person**? ***right*** ✓
5) Is there a **child** behind the **giraffe**? ***no*** ✗

1) What is the **fruit** to the right of the **salad**? ***strawberries*** ✓
2) Is the **fork** to the right of the **salad**? ***no*** ✓
3) Is the **plate** white and square? ***no*** ✓
4) Is the **cup** behind the round **plate**? ***yes*** ✓
5) What is the **plate** made of? ***paper*** ✗

1) Are there either **scarves** or **hats** that are not pink? ***no*** ✓
2) Do the **bear**'s **dress** and the **person**'s **shirt** have the same color? ***yes*** ✓
3) Is the **bear** sitting or standing? ***sitting*** ✓
4) What is the green **object** that the **bear** is sitting on? ***book*** ✓
5) Is the **bear** wearing white **shoes**? ***yes*** ✗



1) Are there either a **chair** or a **clock** in the image? ***no*** ✓
2) Are there any **flowers** behind the **bed** on the left of the **room**? ***yes*** ✓
3) What color is the **appliance** on the right? ***black*** ✓
4) Is the **carpet** brown or blue? ***brown*** ✓
5) Is the **TV** turned on? ***yes*** ✗

Experiments

1. changes in the answer distribution between the training and the test sets,
2. contextual generalization for concepts learned in isolation
3. unseen grammatical structures

Experiments: changes in the answer distribution between the training and the test sets

- VQA-CP - a split of the VQA dataset that has been particularly designed to test generalization skills across changes in the answer distribution between the training and the test sets.

Model	Accuracy
SAN [86]	24.96
HAN [59]	28.65
GVQA [3]	31.30
RAMEN [73]	39.21
BAN [46]	39.31
MuRel [15]	39.54
ReGAT [52]	40.42
NSM	**45.80**

Experiments: contextual generalization for concepts learned in isolation and unseen grammatical structures

- The GQA dataset which focuses on real-world visual reasoning and compositional question answering.

Model	Content	Structure
Global Prior	8.51	14.64
Local Prior	12.14	18.21
Vision	17.51	18.68
Language	21.14	32.88
Lang+Vis	24.95	36.51
BottomUp [5]	29.72	41.83
MAC [40]	31.12	47.27
NSM	**40.24**	**55.72**

Structure Generalization

training	testing
What is the <obj> **covered by**?	What is **covering the** <obj>?
Is there a <obj> in the **image**?	**Do you see any** <obj>s in the **photo**?
What is the <obj> **made of**?	What **material makes up** the <obj>?
What's the name of the <obj> **that is** <attr>?	**What is the** <attr> <obj> **called**?

Content Generalization

training	testing
Only questions that **do not** refer to any type of **food** or **animal** (do not include any word from these categories)	Only questions that refer to **foods** or **animals** (include a word from one of these categories)

Questions

- 1. Дайте определение сети Neural State Machine
- 2. Через какие два этапа проходит Neural State Machine и для чего нужен каждый из них?
- 3. Опишите один из этих этапов.