Learning to Compose Neural Networks for Question Answering

Authors: Jacob Andreas, Marcus Rohrbach , Trevor Darrell, Dan Klein

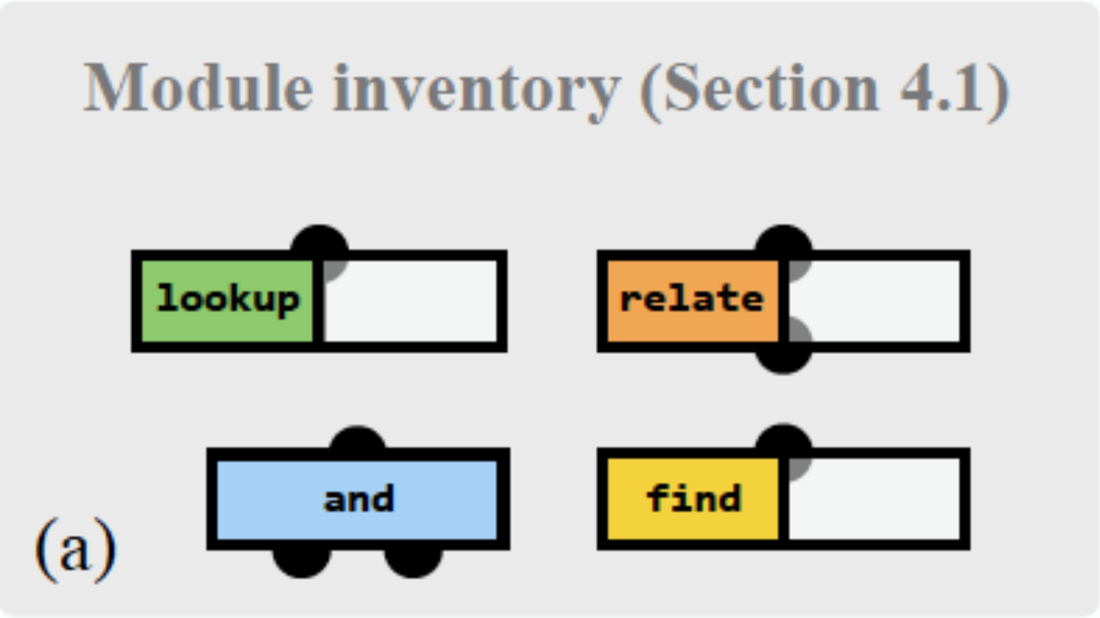
Research Highlight Presented by Zhedi Liu

High Level Overview

A compositional, attentional model for answering questions about a variety of world representations, including images and structured knowledge bases.

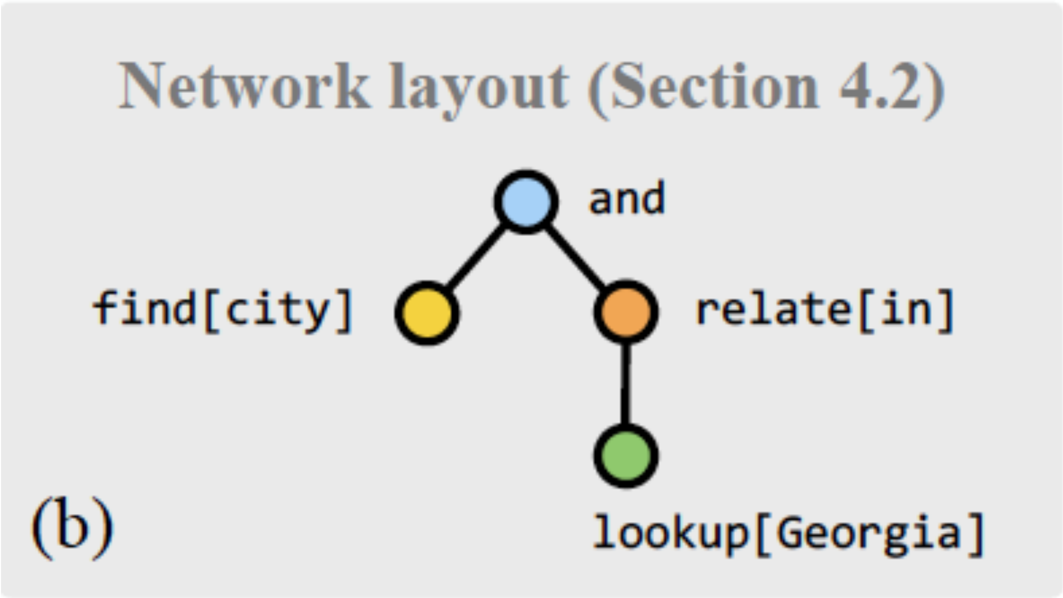
Two components, Trained Jointly

- Query: What cities are in Georgia?
- A collection of neural “modules” that can be freely composed



Two components, Trained Jointly

- Query: What cities are in Georgia?
- A network layout predictor that assembles modules into complete deep networks tailored to each question



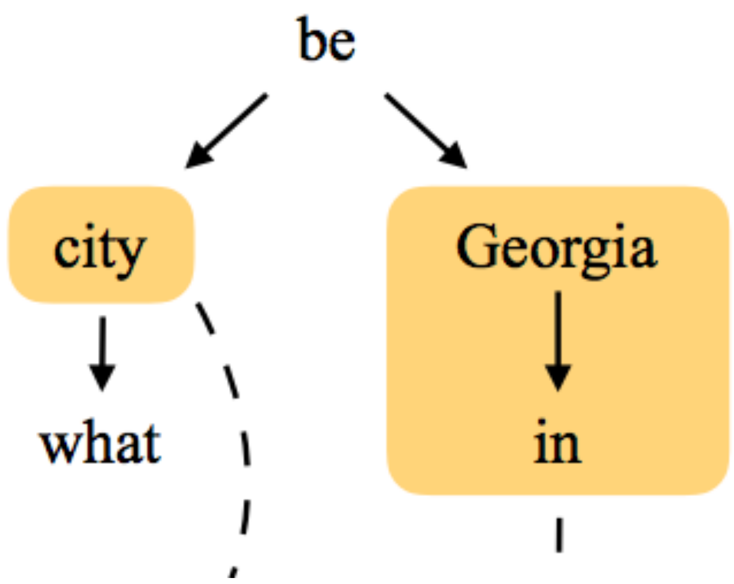
Model: Built around Two Distributions

- **A Layout Model**: $p(z|x;\theta_\ell)$
  - chooses a layout for a sentence
- **An Execution Model**: $p_z(y|w;\theta_e)$
  - applies the network specified a particular layout to a world representation

1. w a world representation
2. x a question
3. y an answer
4. z a network layout
5. θ a collection of model parameters

Layout Model

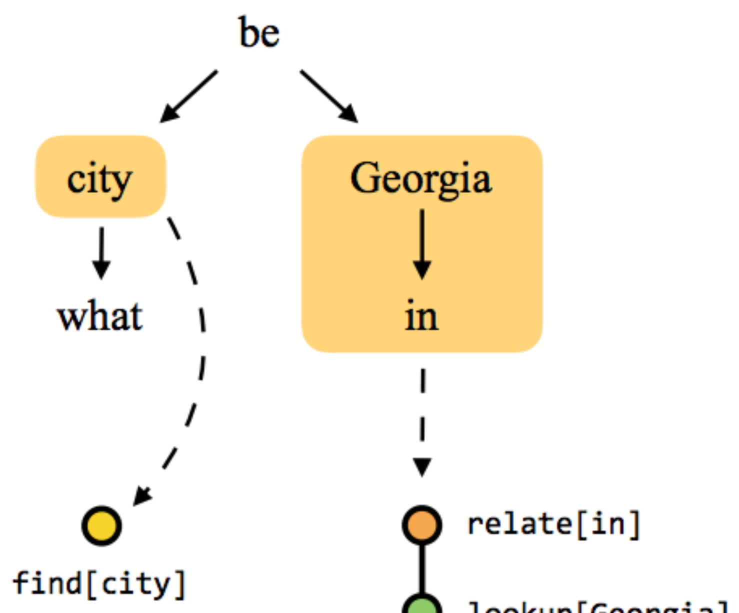
Step 1: Represent the input sentence as a dependency tree.



Query: What cities are in Georgia?

Layout Model

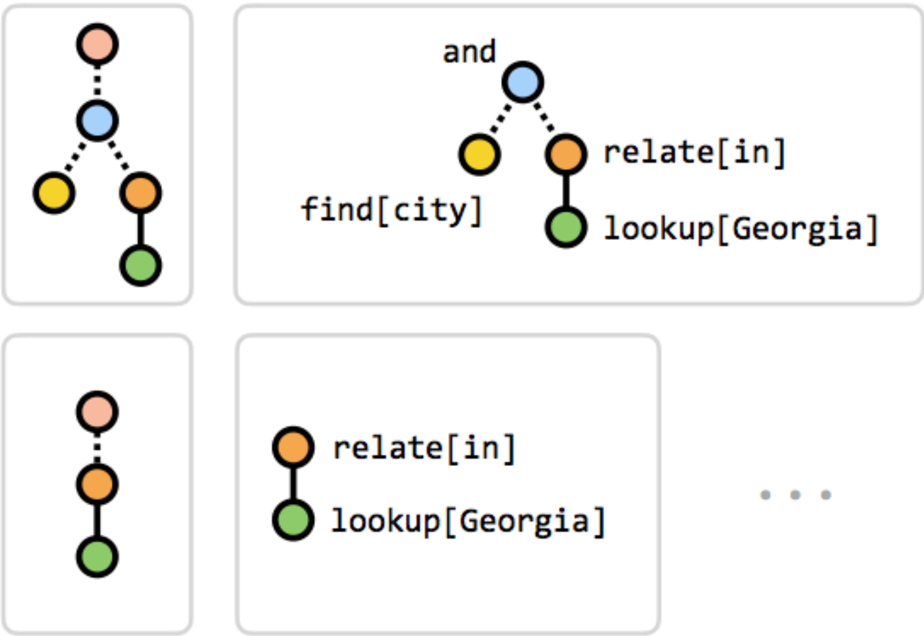
Step 2: Associate fragments of the dependency parse with appropriate modules



Query: What cities are in Georgia?

Layout Model

Step 3: Assemble fragments into full layouts



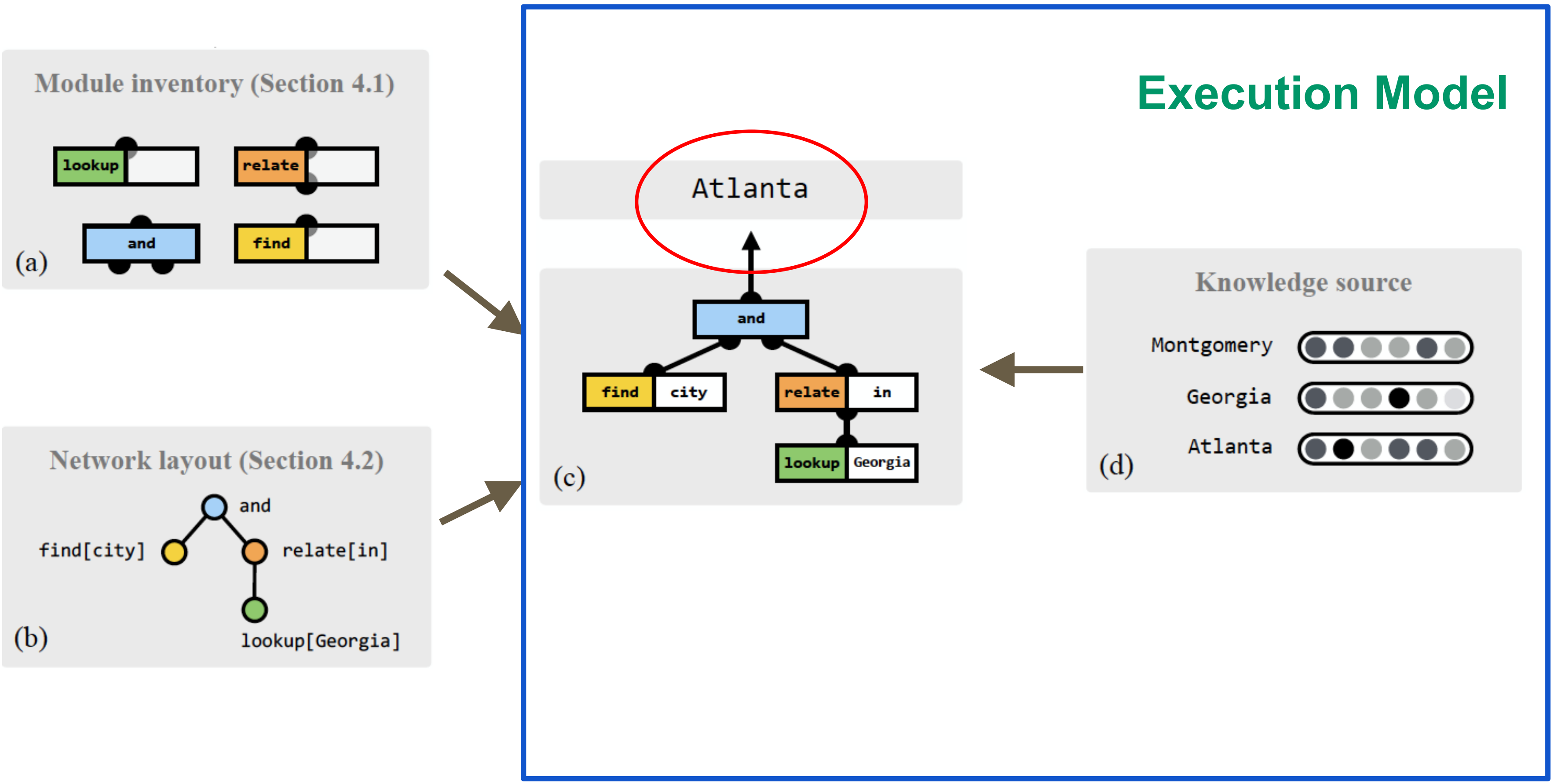
Query: What cities are in Georgia?

Layout Scoring Model

- Produce an LSTM representation of the question, a feature-based representation of the query, and pass both representations through a multilayer perceptron

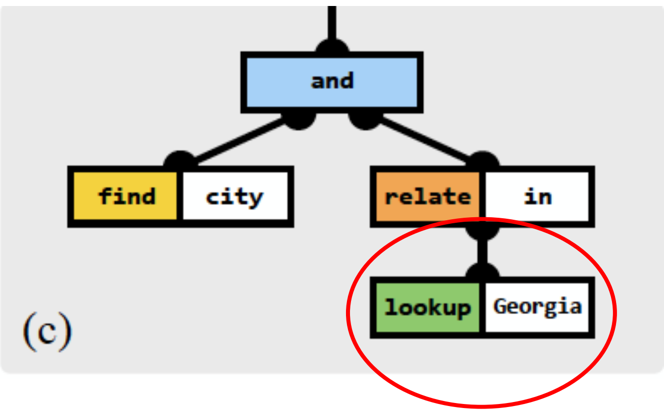
- The update to the layout-scoring model at each timestep is simply **the gradient of the log-probability of the chosen layout, scaled by the accuracy of that layout's predictions**

Execution Model



Query: What cities are in Georgia?

Module: lookup



(c)

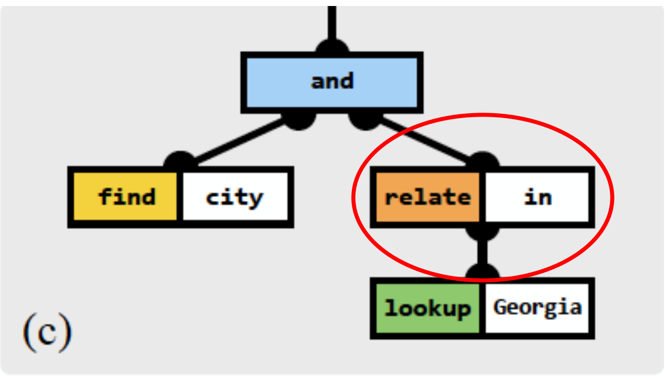
Lookup (→ Attention)

`lookup[i]` produces an attention focused entirely at the index $f(i)$, where the relationship f between words and positions in the input map is known ahead of time (e.g. string matches on database fields).

$$[\![\texttt{lookup}[i]]\!] = e_{f(i)} \qquad (2)$$

where e_i is the basis vector that is 1 in the ith position and 0 elsewhere.

Module: relate

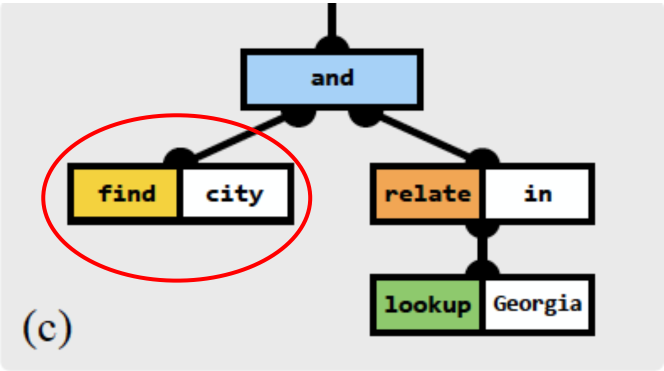


Relate (<u>Attention</u> $\rightarrow$ <u>Attention</u>)

`relate` directs focus from one region of the input to another. It behaves much like the `find` module, but also conditions its behavior on the current region of attention h. Let $\bar{w}(h) = \sum_k h_k w^k$, where h_k is the k^{th} element of h. Then,

$$[\![\texttt{relate[}i\texttt{]}(h)]\!] = \text{softmax}(a \odot \sigma(Bv^i \oplus CW \oplus D\bar{w}(h) \oplus e)) \tag{4}$$

Module: find



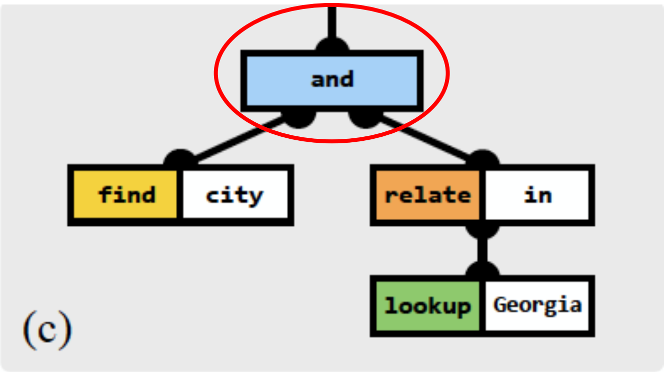
(c)

Find (→ Attention)

`find[i]` computes a distribution over indices by concatenating the parameter argument with each position of the input feature map, and passing the concatenated vector through a MLP:

$$[\![\texttt{find}[i]]\!] = \mathrm{softmax}(a \odot \sigma(Bv^i \oplus CW \oplus d)) \quad (3)$$

Module: and



And (Attention* → Attention)

`and` performs an operation analogous to set intersection for attentions. The analogy to probabilistic logic suggests multiplying probabilities:

$$[\![\texttt{and}(h^1, h^2, \ldots)]\!] = h^1 \odot h^2 \odot \cdots \qquad (5)$$

Train an Execution Model

- Maximize $\sum_{(w,y,z)} \log p_z(y|w;\theta_e)$

1. w a world representation
2. x a question
3. y an answer
4. z a network layout
5. θ a collection of model parameters

State-of-the-art Performance: VQA

State-of-the-art Performance: VQA

	test-dev				test-std
	Yes/No	Number	Other	All	All
Zhou (2015)	76.6	35.0	42.6	55.7	55.9
Noh (2015)	80.7	37.2	41.7	57.2	57.4
Yang (2015)	79.3	36.6	46.1	58.7	58.9
NMN	81.2	38.0	44.0	58.6	58.7
D-NMN	81.1	38.6	45.5	59.4	**59.4**

State-of-the-art Performance: GeoQA

Is Key Largo an island?

```
(exists (and lookup[key-largo] find[island]))
```

yes: correct

What national parks are in Florida?

```
(and find[park] (relate[in] lookup[florida]))
```

everglades: correct

What are some beaches in Florida?

```
(exists (and lookup[beach]
             (relate[in] lookup[florida])))
```

yes (daytona-beach): wrong parse

What beach city is there in Florida?

```
(and lookup[beach] lookup[city]
     (relate[in] lookup[florida]))
```

[none] (daytona-beach): wrong module behavior

	Accuracy	
Model	GeoQA	GeoQA+Q
LSP-F	48	–
LSP-W	51	–
NMN	51.7	35.7
D-NMN	**54.3**	**42.9**