Causal abstractions of neural natural language inference models

Christopher Potts
Joint work with Atticus Geiger, Josh Rozner, Hanson Lu, Thomas Icard, and Noah Goodman

Stanford Linguistics and the Stanford NLP Group

ILFC Seminar, October 12, 2021

My ACL talk engaging with ‘NLP for Social Good’

YouTube

My ACL talk engaging with 'NLP for Social Good'

Reliable characterizations of NLP systems as a social responsibility

YouTube

My ACL talk engaging with ‘NLP for Social Good’

Reliable characterizations of NLP systems as a social responsibility

1. Benchmark datasets: Delimit responsible use
2. System assessment: Connect with real-world concerns

YouTube

My ACL talk engaging with ‘NLP for Social Good’

Reliable characterizations of NLP systems as a social responsibility

1. Benchmark datasets: Delimit responsible use
2. System assessment: Connect with real-world concerns

Do exactly what you said you would do.

YouTube

My ACL talk engaging with ‘NLP for Social Good’

Reliable characterizations of NLP systems as a social responsibility

1. Benchmark datasets: Delimit responsible use
2. System assessment: Connect with real-world concerns
3. Structural evaluation methods: Seek guarantees

Do exactly what you said you would do.

YouTube

Overview: Structural evaluation methods

Overview: Structural evaluation methods

Motivations

Overview: Structural evaluation methods

Motivations

	Characterize representations	Causal inference	Improved training

Overview: Structural evaluation methods

Motivations

	Characterize representations	Causal inference	Improved training
Probing	😊		

Overview: Structural evaluation methods

Motivations

	Characterize representations	Causal inference	Improved training
Probing	😊		
Feature attribution	🤔	😊	

Overview: Structural evaluation methods

Motivations

	Characterize representations	Causal inference	Improved training
Probing	😊		
Feature attribution	🤔	😊	
Causal abstraction	😊	😊	😊

Overview: Structural evaluation methods

Motivations

	Characterize representations	Causal inference	Improved training
Probing	😊		
Feature attribution	🤔	😊	
Causal abstraction	😊	😊	😊

Case study: Monotonicity NLI

Motivations

Systematicity

Systematicity

Fodor and Pylyshyn (1988:37):

"What we mean when we say that linguistic capacities are *systematic* is that the ability to produce/understand some sentences is *intrinsically* connected to the ability to produce/understand certain others."

Systematicity

Fodor and Pylyshyn (1988:37):

"What we mean when we say that linguistic capacities are *systematic* is that the ability to produce/understand some sentences is *intrinsically* connected to the ability to produce/understand certain others."

1. Sandy loves the puppy.

Systematicity

Fodor and Pylyshyn (1988:37):

"What we mean when we say that linguistic capacities are *systematic* is that the ability to produce/understand some sentences is *intrinsically* connected to the ability to produce/understand certain others."

1. Sandy loves the puppy.
2. The puppy loves Sandy.

Systematicity

Fodor and Pylyshyn (1988:37):

"What we mean when we say that linguistic capacities are *systematic* is that the ability to produce/understand some sentences is *intrinsically* connected to the ability to produce/understand certain others."

1. Sandy loves the puppy.
2. The puppy loves Sandy.
3. the turtle $\sim$ the puppy

Systematicity

Fodor and Pylyshyn (1988:37):

"What we mean when we say that linguistic capacities are *systematic* is that the ability to produce/understand some sentences is *intrinsically* connected to the ability to produce/understand certain others."

1. Sandy loves the puppy.
2. The puppy loves Sandy.
3. the turtle $\sim$ the puppy
4. The turtle loves the puppy.

Systematicity

Fodor and Pylyshyn (1988:37):

"What we mean when we say that linguistic capacities are *systematic* is that the ability to produce/understand some sentences is *intrinsically* connected to the ability to produce/understand certain others."

1. Sandy loves the puppy.
2. The puppy loves Sandy.
3. the turtle $\sim$ the puppy
4. The turtle loves the puppy.
5. The puppy loves the turtle.

Systematicity

Fodor and Pylyshyn (1988:37):

"What we mean when we say that linguistic capacities are *systematic* is that the ability to produce/understand some sentences is *intrinsically* connected to the ability to produce/understand certain others."

1. Sandy loves the puppy.
2. The puppy loves Sandy.
3. the turtle $\sim$ the puppy
4. The turtle loves the puppy.
5. The puppy loves the turtle.
6. The turtle loves Sandy.

Systematicity

Fodor and Pylyshyn (1988:37):

“What we mean when we say that linguistic capacities are *systematic* is that the ability to produce/understand some sentences is *intrinsically* connected to the ability to produce/understand certain others.”

1. Sandy loves the puppy.
2. The puppy loves Sandy.
3. the turtle $\sim$ the puppy
4. The turtle loves the puppy.
5. The puppy loves the turtle.
6. The turtle loves Sandy.
7. ...

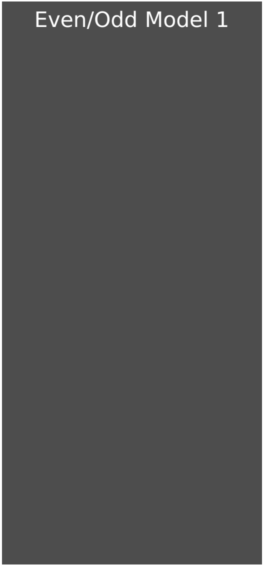
Systematicity

Fodor and Pylyshyn (1988:37):

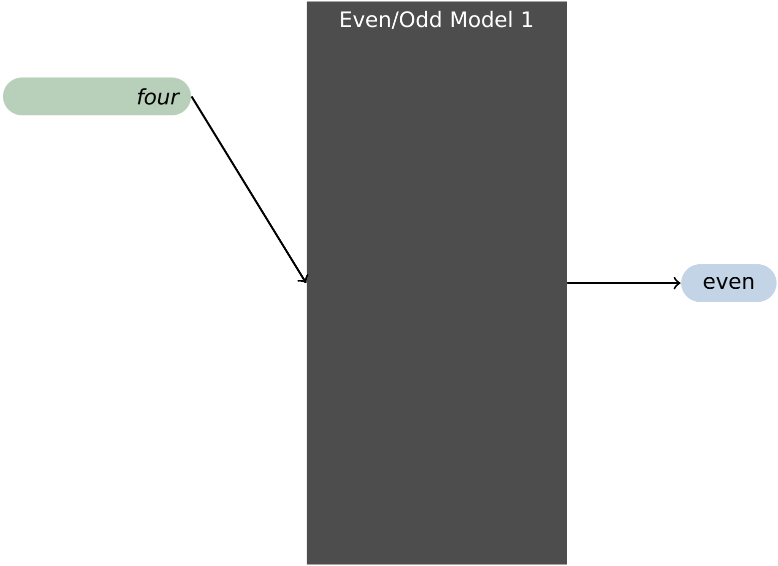
"What we mean when we say that linguistic capacities are *systematic* is that the ability to produce/understand some sentences is *intrinsically* connected to the ability to produce/understand certain others."

Example		Gold	Prediction
The bakery	sells a mean apple pie.	pos	pos
They	sell a mean apple pie.	pos	pos
She	sells a mean apple pie.	pos	neg
He	sells a mean apple pie.	pos	neg

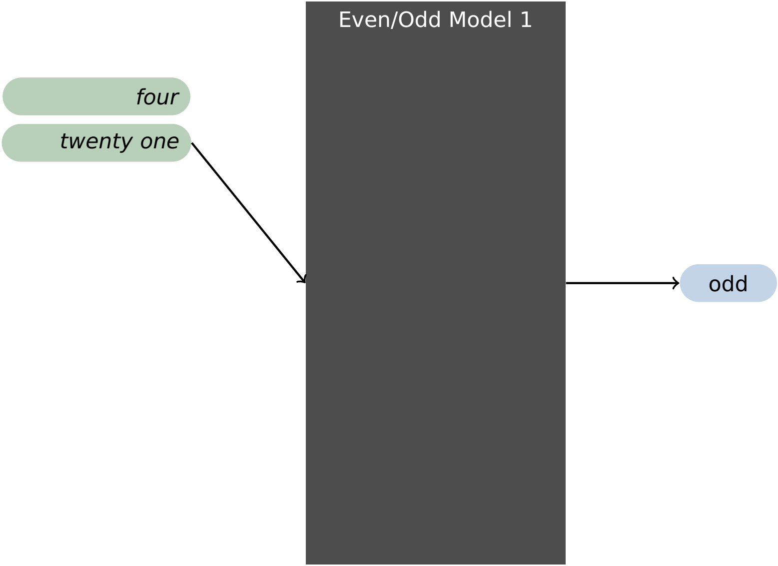
Limits of behavioral testing



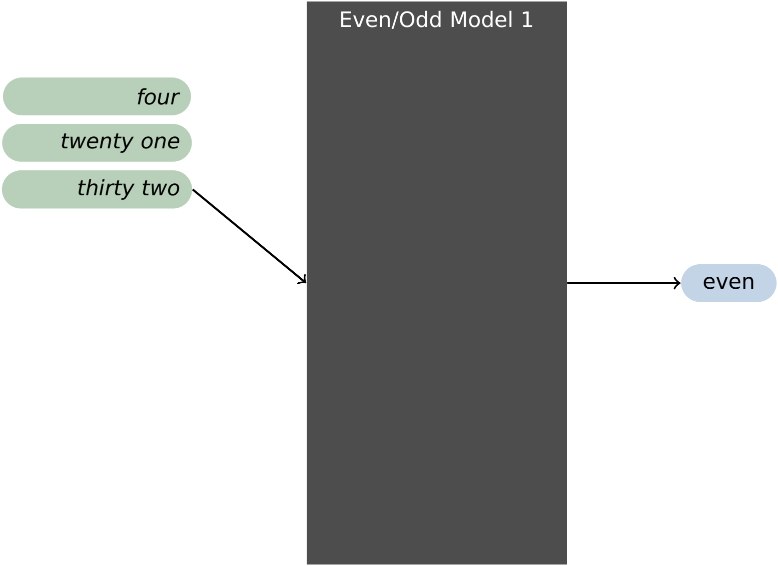
Limits of behavioral testing



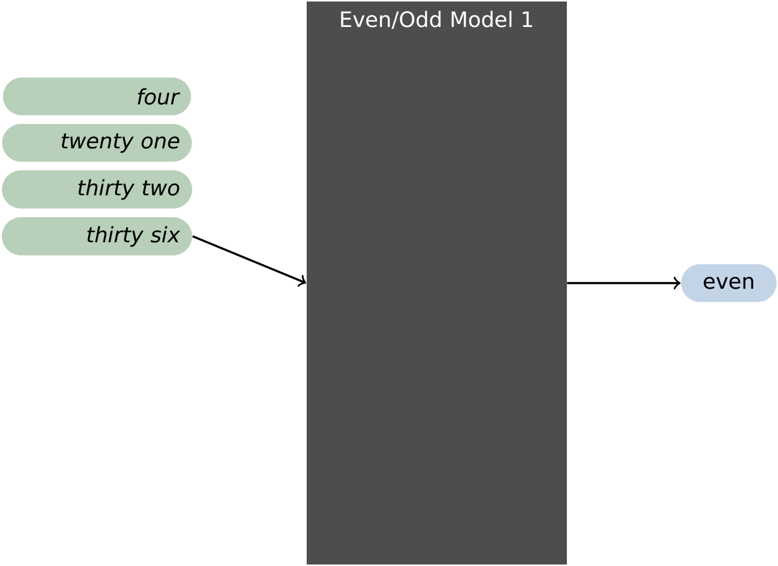
Limits of behavioral testing



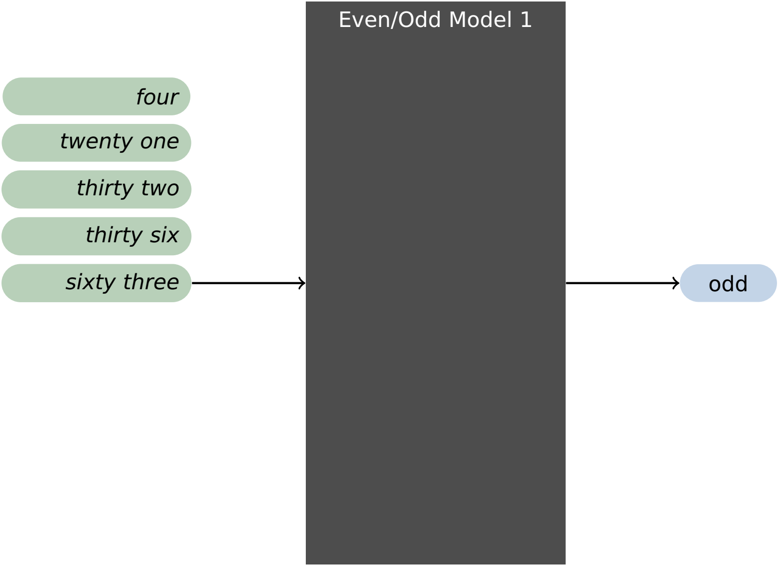
Limits of behavioral testing



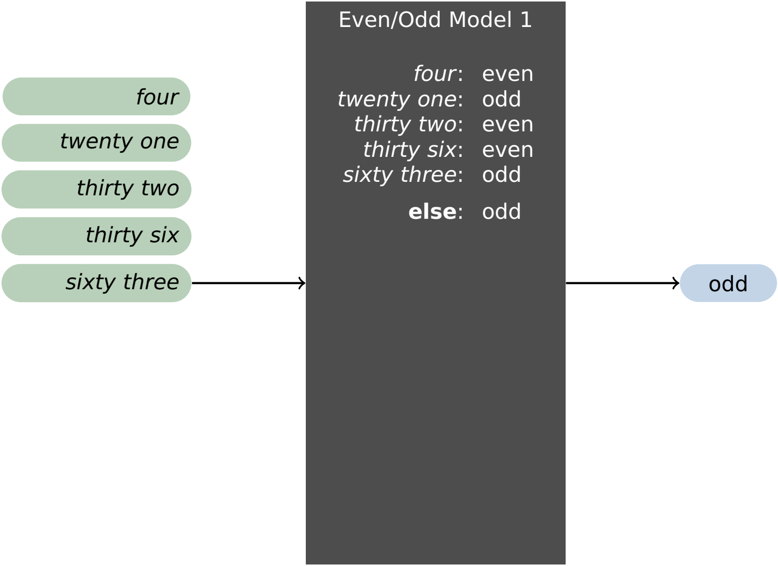
Limits of behavioral testing



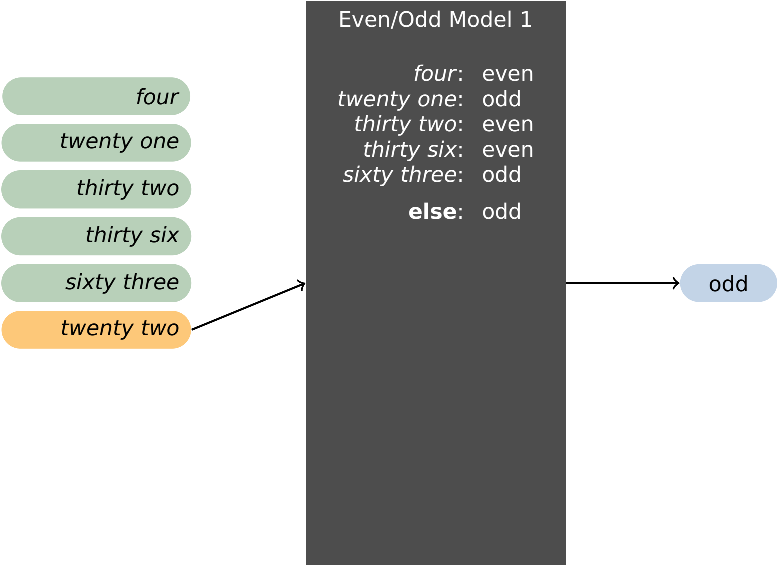
Limits of behavioral testing



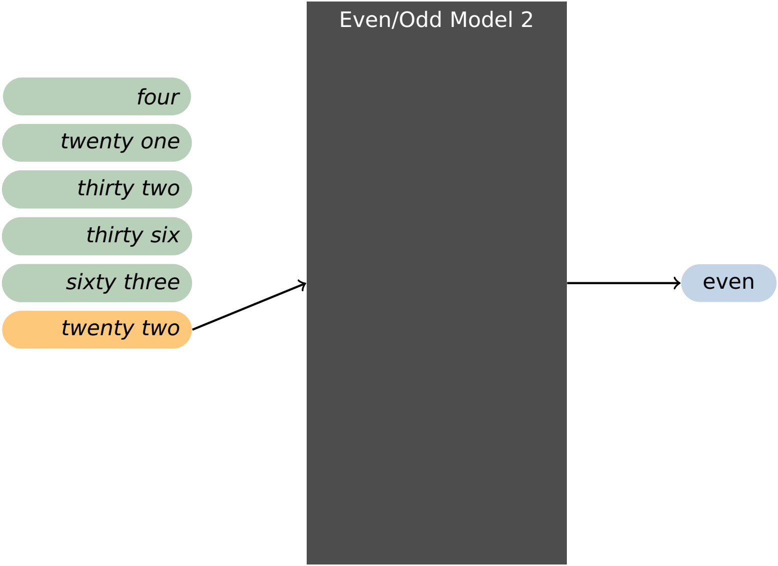
Limits of behavioral testing



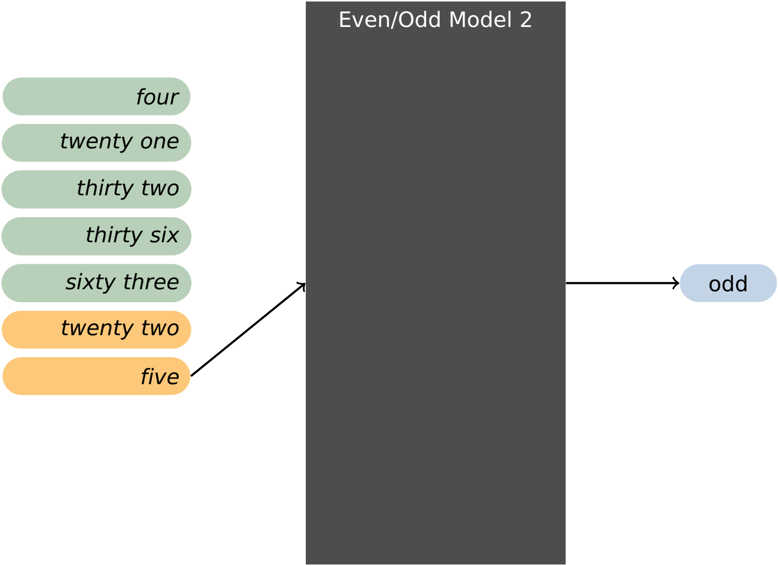
Limits of behavioral testing



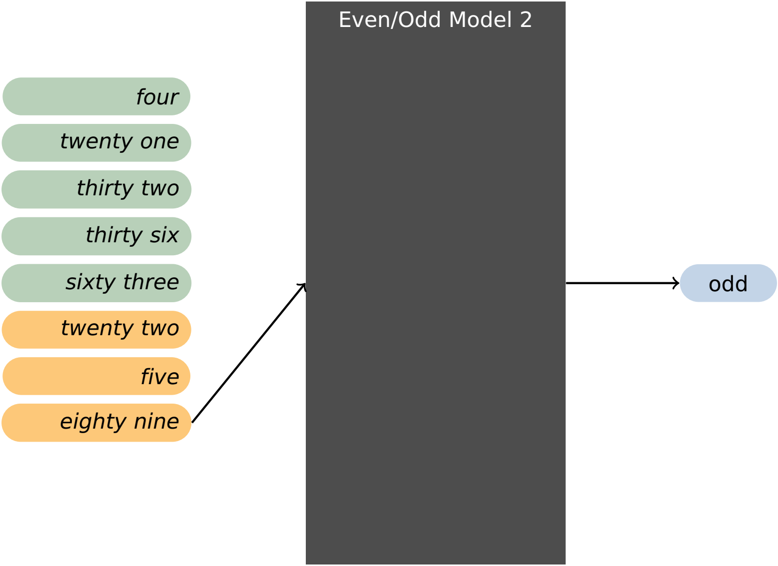
Limits of behavioral testing



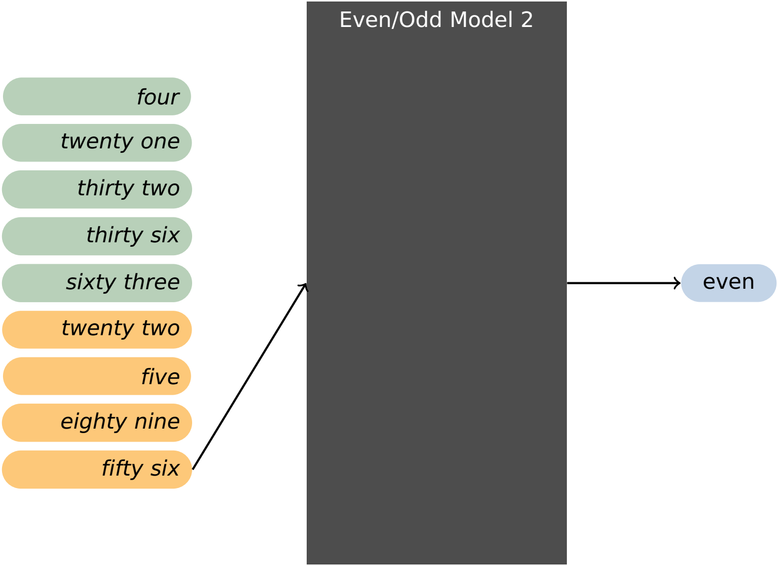
Limits of behavioral testing



Limits of behavioral testing



Limits of behavioral testing



Limits of behavioral testing

four
twenty one
thirty two
thirty six
sixty three
twenty two
five
eighty nine
fifty six

Even/Odd Model 2

$$d = \begin{bmatrix} \textit{one}\text{:} & \text{odd} \\ \textit{two}\text{:} & \text{even} \\ \textit{three}\text{:} & \text{odd} \\ \textit{four}\text{:} & \text{even} \\ \textit{five}\text{:} & \text{odd} \\ \textit{six}\text{:} & \text{even} \\ \textit{seven}\text{:} & \text{odd} \\ \textit{eight}\text{:} & \text{even} \\ \textit{nine}\text{:} & \text{odd} \\ \textbf{else}\text{:} & \text{odd} \end{bmatrix}$$

return
d[input final token]

even

Limits of behavioral testing

four
twenty one
thirty two
thirty six
sixty three
twenty two
five
eighty nine
fifty six
sixteen

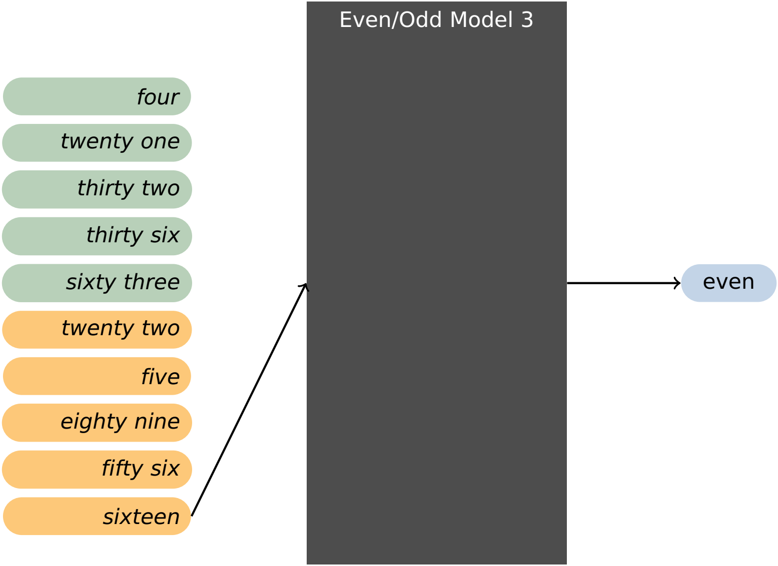
Even/Odd Model 2

$$d = \begin{bmatrix} \textit{one}: & \text{odd} \\ \textit{two}: & \text{even} \\ \textit{three}: & \text{odd} \\ \textit{four}: & \text{even} \\ \textit{five}: & \text{odd} \\ \textit{six}: & \text{even} \\ \textit{seven}: & \text{odd} \\ \textit{eight}: & \text{even} \\ \textit{nine}: & \text{odd} \\ \textbf{else}: & \text{odd} \end{bmatrix}$$

return
d[*input final token*]

odd

Limits of behavioral testing



Seeking generalization guarantees

Seeking generalization guarantees

- Goal: causal analysis of a model's structure, to obtain guarantees about how it will behave.

Seeking generalization guarantees

- Goal: causal analysis of a model's structure, to obtain guarantees about how it will behave.
- Further questions of
  - fairness
  - bias
  - reliability
  - robustness

  are hard to address without such guarantees.

Improving networks

Improving networks

Structural analysis as the first step towards training networks to have the properties we want.

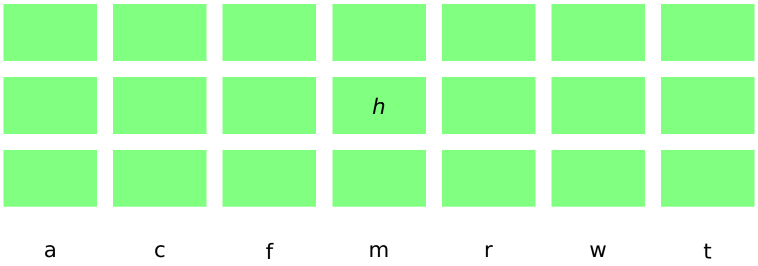
Probing

Core idea behind probing

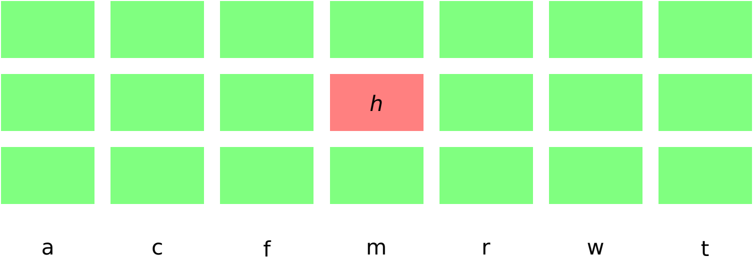
Use a supervised model (the probe) to determine what is latently encoded in the hidden representations of a target models.

Conneau et al. 2018; Tenney et al. 2019

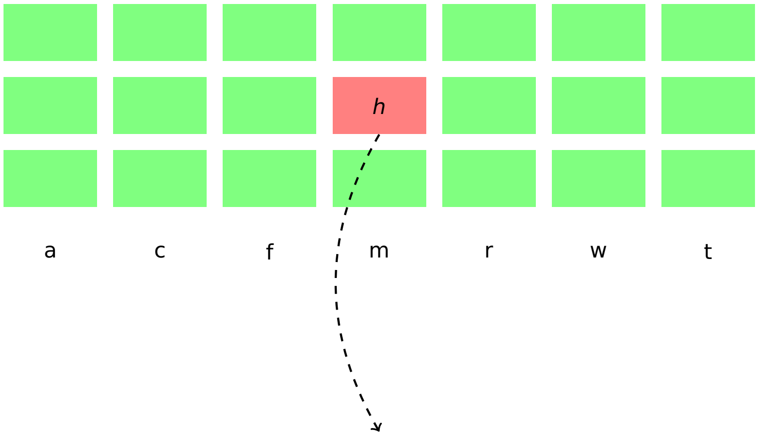
Core method



Core method

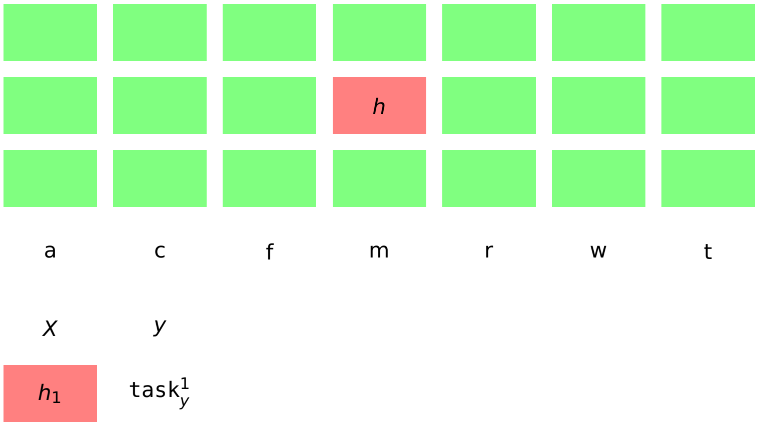


Core method

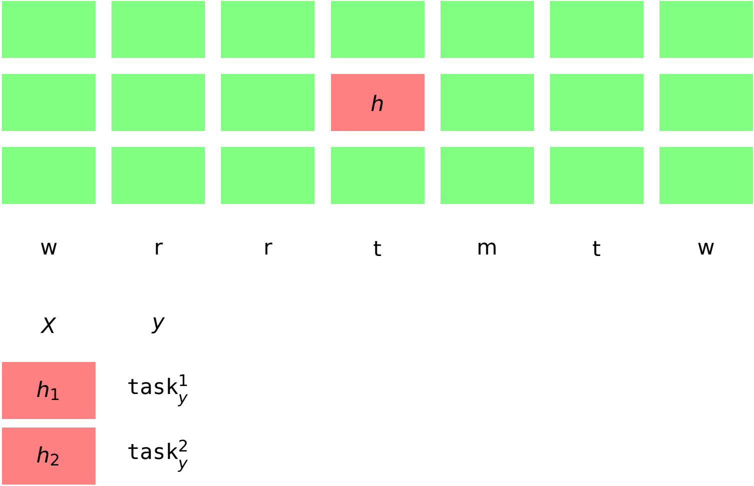


$\text{SmallLinearModel}(h) = \textit{task}$

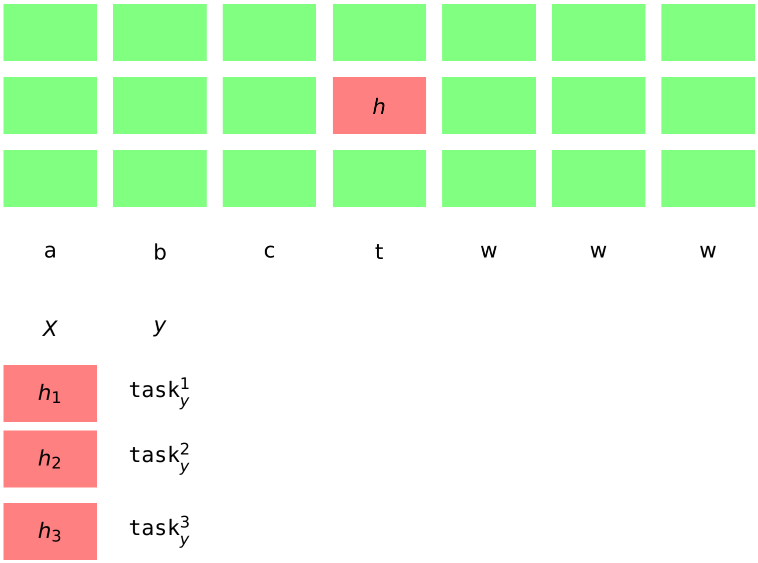
Core method



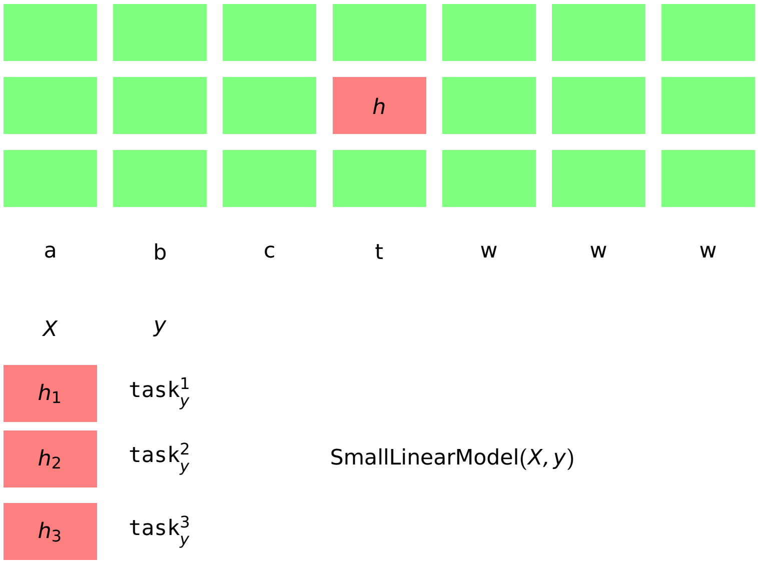
Core method



Core method

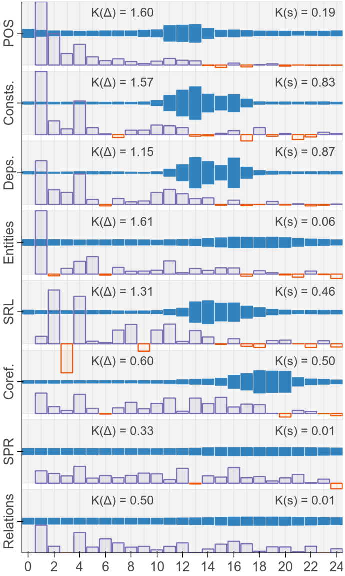

h
a b c t w w w
X y
h_1 $\texttt{task}_y^1$
h_2 $\texttt{task}_y^2$
h_3 $\texttt{task}_y^3$

Core method


h
a b c t w w w
X
y
h_1
$\mathtt{task}_y^1$
h_2
$\mathtt{task}_y^2$
SmallLinearModel(X, y)
h_3
$\mathtt{task}_y^3$

Core method

Probing BERT



Tenney et al. 2019

Central limitations

Central limitations

Probing or learning a new model?

Central limitations

Probing or learning a new model?

1. A probe is a supervised model with a particular featurization choice.

Central limitations

Probing or learning a new model?

1. A probe is a supervised model with a particular featurization choice.
2. At least some of the information that we identify is likely to be stored in the probe model.

Central limitations

Probing or learning a new model?

1. A probe is a supervised model with a particular featurization choice.
2. At least some of the information that we identify is likely to be stored in the probe model.
3. Responses:
   - Unsupervised probes (Saphra and Lopez 2019; Clark et al. 2019; Hewitt and Manning 2019)
   - Control tasks (Hewitt and Liang 2019)

Central limitations

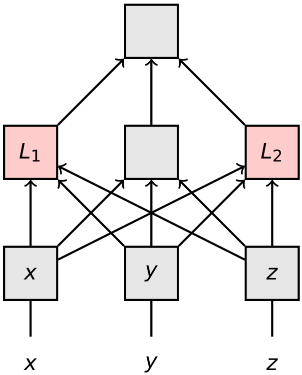
Probing or learning a new model?

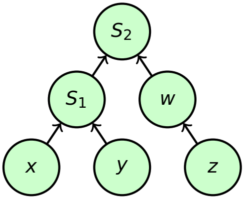
1. A probe is a supervised model with a particular featurization choice.
2. At least some of the information that we identify is likely to be stored in the probe model.
3. Responses:
   - Unsupervised probes (Saphra and Lopez 2019; Clark et al. 2019; Hewitt and Manning 2019)
   - Control tasks (Hewitt and Liang 2019)

No causal inference

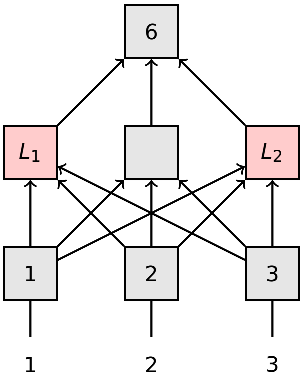
Probes cannot tell us about whether the information that we identify has any *causal* relationship with the target model's behavior (Belinkov and Glass 2019; Geiger et al. 2020, 2021).

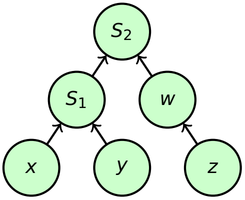
Simple running example



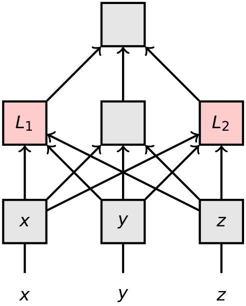


Simple running example

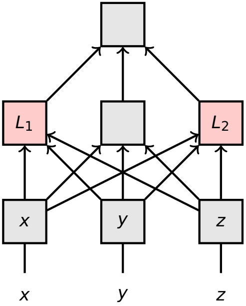




No causal inferences

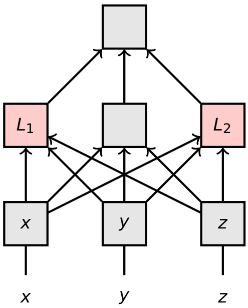


No causal inferences



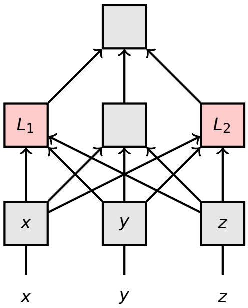
1. Probe L_1: it computes $x + y$

No causal inferences

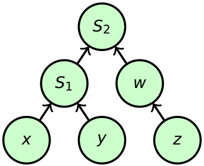


1. Probe L_1: it computes $x + y$
2. Probe L_2: it computes z

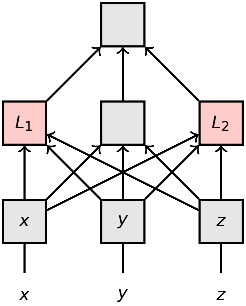
No causal inferences



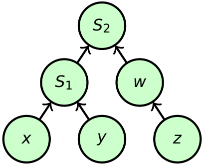
1. Probe L_1: it computes $x + y$
2. Probe L_2: it computes z
3. Aha!



No causal inferences



1. Probe L_1: it computes $x + y$
2. Probe L_2: it computes z
3. Aha!



4. But neither has any impact on the output!

$$W_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \quad W_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad W_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\mathbf{w} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad (\mathbf{x}W_1; \mathbf{x}W_2; \mathbf{x}W_3)\,\mathbf{w}$$

Summary

	Characterize representations	Causal inference	Improved training
Probing	😊		
Feature attribution	🤔	😊	
Causal abstraction	😊	😊	😊

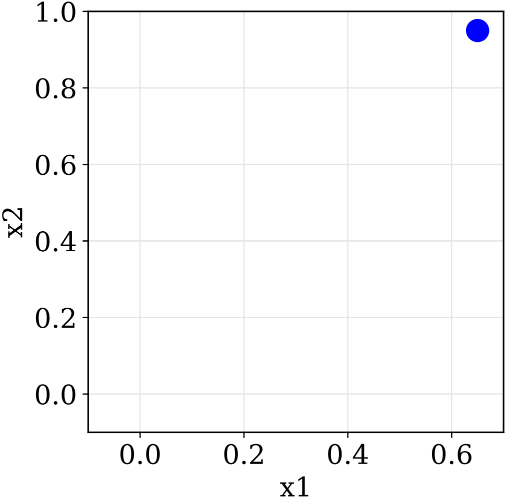
Feature attribution

captum.ai

1. Integrated gradients (Sundararajan et al. 2017)
2. Gradients
3. Saliency Maps (Simonyan et al. 2013)
4. DeepLift (Shrikumar et al. 2017)
5. Deconvolution (Zeiler and Fergus 2014)
6. LIME (Ribeiro et al. 2016)
7. Feature ablation
8. Feature permutation
9. ...

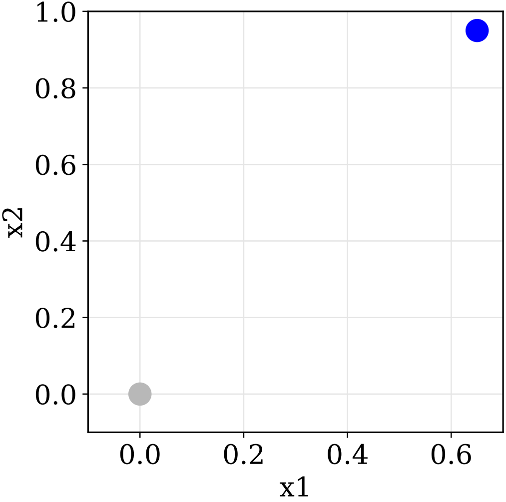
https://captum.ai

Integrated gradients: Intuition



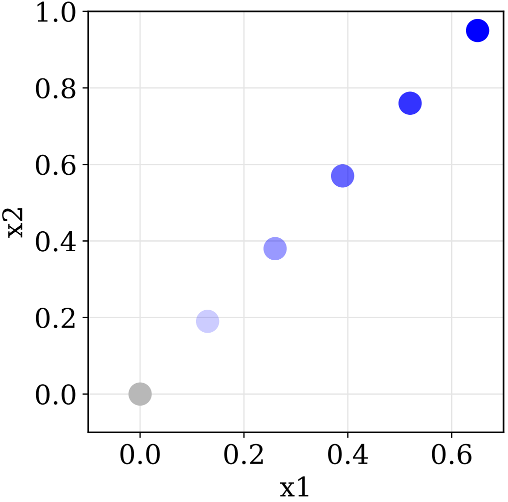
Sundararajan et al. 2017; slide with IG definition

Integrated gradients: Intuition



Sundararajan et al. 2017; slide with IG definition

Integrated gradients: Intuition



Sundararajan et al. 2017; slide with IG definition

Central properties

Central properties

Sensitivity

If two inputs x and x' differ only at dimension i and lead to different predictions, then feature f_i has non-zero attribution.

$$M([1,0,1]) = \text{positive}$$
$$M([1,1,1]) = \text{negative}$$

Central properties

Sensitivity

If two inputs x and x' differ only at dimension i and lead to different predictions, then feature f_i has non-zero attribution.

$$M([1,0,1]) = \text{positive}$$
$$M([1,1,1]) = \text{negative}$$

Completeness

For input x and baseline x', the sum of attributions for x is equal to $M(x) - M(x')$.

Central properties

Sensitivity

If two inputs x and x' differ only at dimension i and lead to different predictions, then feature f_i has non-zero attribution.

$$M([1,0,1]) = \text{positive}$$
$$M([1,1,1]) = \text{negative}$$

Completeness

For input x and baseline x', the sum of attributions for x is equal to $M(x) - M(x')$.

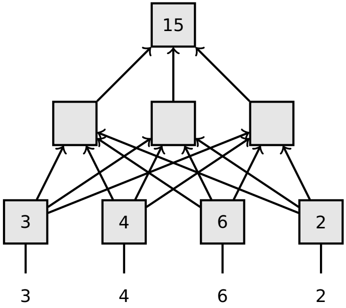
Implementation invariance

If two models M and M' have identical input/output behavior, then the attributions for M and M' are identical.

Reliable insights about causal structure

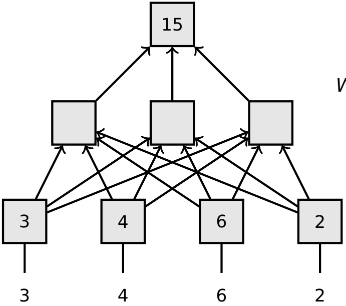
Sundararajan et al. 2017

Reliable insights about causal structure



Sundararajan et al. 2017

Reliable insights about causal structure

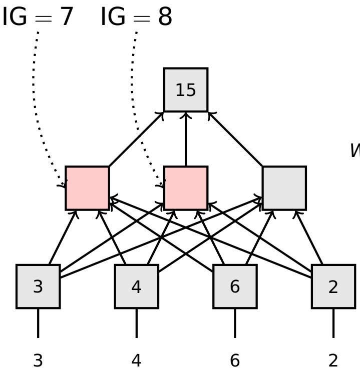


$$W_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad W_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad W_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{w} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \quad (\mathbf{x}W_1; \mathbf{x}W_2; \mathbf{x}W_3)\,\mathbf{w}$$

Sundararajan et al. 2017

Reliable insights about causal structure



$$W_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad W_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad W_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{w} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \qquad (\mathbf{x}W_1; \mathbf{x}W_2; \mathbf{x}W_3)\,\mathbf{w}$$

Sundararajan et al. 2017

Summary

	Characterize representations	Causal inference	Improved training
Probing	😊		
Feature attribution	🤔	😊	
Causal abstraction	😊	😊	😊

Causal abstraction

Recipe

Geiger et al. 2020, 2021

Recipe

1. State a hypothesis about (an aspect of) the target model's causal structure.

Geiger et al. 2020, 2021

Recipe

1. State a hypothesis about (an aspect of) the target model's causal structure.
2. Search for an alignment betewen the causal model and target model.

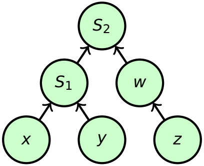
Geiger et al. 2020, 2021

Recipe

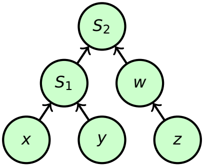
1. State a hypothesis about (an aspect of) the target model's causal structure.
2. Search for an alignment betewen the causal model and target model.
3. Perform *interchange interventions*.

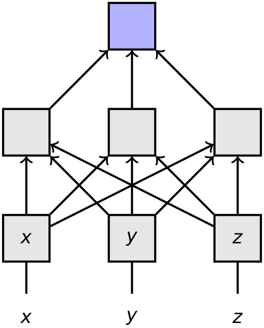
Geiger et al. 2020, 2021

Interchange intervention analysis

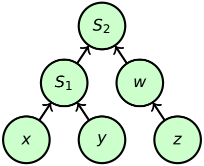


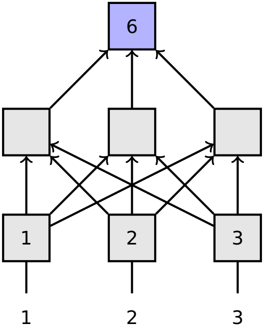
Interchange intervention analysis



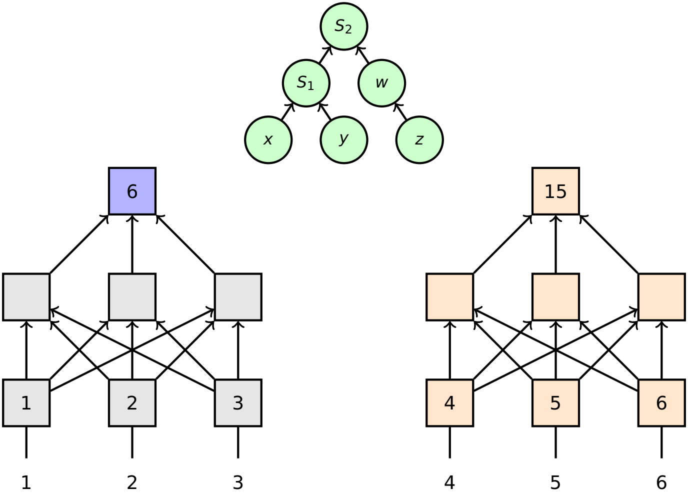


Interchange intervention analysis

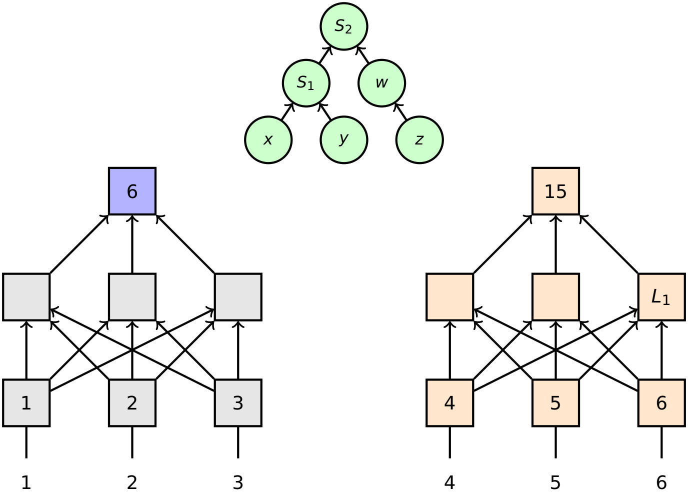




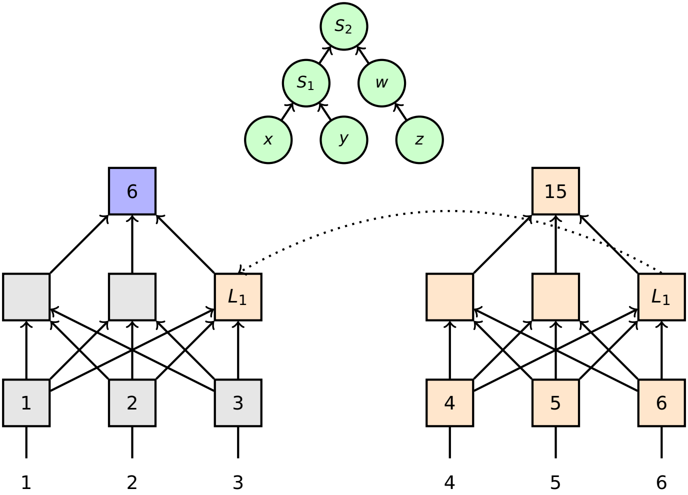
Interchange intervention analysis



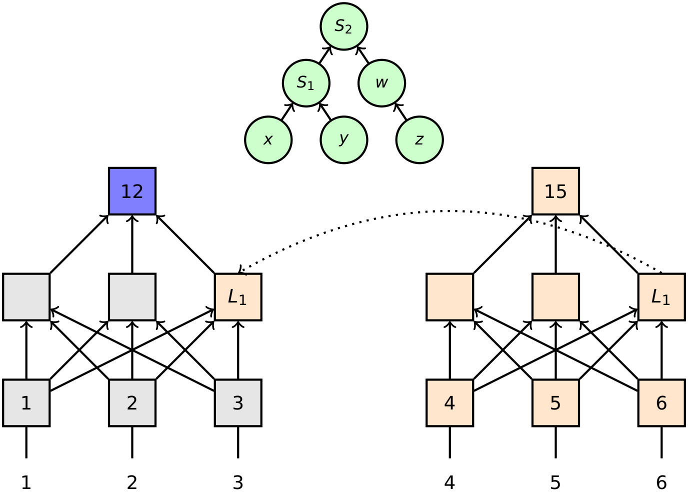
Interchange intervention analysis



Interchange intervention analysis



Interchange intervention analysis



Connections to the literature

- Constructive abstraction (Beckers et al. 2020)
- Causal mediation analysis (Vig et al. 2020)
- Role Learning Networks (Soulos et al. 2020)
- CausaLM (Feder et al. 2021)
- Amnesic Probing (Elazar et al. 2021)

Summary

	Characterize representations	Causal inference	Improved training
Probing	😊		
Feature attribution	🤔	😊	
Causal abstraction	😊	😊	😊

Monotonicity NLI (MoNLI)

MoNLI dataset construction

MoNLI dataset construction

Positive MoNLI (PMoNLI; 1,476 examples)

MoNLI dataset construction

Positive MoNLI (PMoNLI; 1,476 examples)

SNLI hypothesis (A)	Food was served.

MoNLI dataset construction

Positive MoNLI (PMoNLI; 1,476 examples)

SNLI hypothesis (A)	Food was served.
WordNet	pizza $\sqsubset$ food

MoNLI dataset construction

Positive MoNLI (PMoNLI; 1,476 examples)

SNLI hypothesis (A)	Food was served.
WordNet	pizza $\sqsubset$ food
New example (B)	Pizza was served.

MoNLI dataset construction

Positive MoNLI (PMoNLI; 1,476 examples)

SNLI hypothesis (A)	Food was served.
WordNet	pizza $\sqsubset$ food
New example (B)	Pizza was served.
Positive MoNLI	(A) **neutral** (B)
Positive MoNLI	(B) **entailment** (A)

MoNLI dataset construction

Positive MoNLI (PMoNLI; 1,476 examples)

SNLI hypothesis (A)	Food was served.
WordNet	pizza $\sqsubset$ food
New example (B)	Pizza was served.
Positive MoNLI	(A) **neutral** (B)
Positive MoNLI	(B) **entailment** (A)

Negative MoNLI (PMoNLI; 1,202 examples)

SNLI hypothesis (A)	The children are **not** holding plants.
WordNet	flowers $\sqsubset$ plants
New example (B)	The children are **not** holding flowers.
Negative MoNLI	(A) **entailment** (B)
Negative MoNLI	(B) **neutral** (A)

MoNLI monotonicity algorithm

MoNLI monotonicity algorithm

Infer(*example*)

lexrel ← get-lexrel(*example*)
if contains-not(*example*)
        return reverse(*lexrel*)
return *lexrel*

MoNLI monotonicity algorithm

```
Infer(example)
lexrel ← get-lexrel(example)
if contains-not(example)
        return reverse(lexrel)
return lexrel
```

MoNLI	Pizza was served.	**entailment**	Food was served.
lexrel	Pizza	**entailment**	Food

MoNLI monotonicity algorithm

Infer(*example*)

```
lexrel ← get-lexrel(example)
if contains-not(example)
        return reverse(lexrel)
return lexrel
```

MoNLI	Pizza was served.	**entailment**	Food was served.
lexrel	Pizza	**entailment**	Food
MoNLI	Pizza was not served.	**neutral**	Food was not served.
lexrel	Pizza	**entailment**	Food
reverse(*lexrel*)		**neutral**	

Models

Models

BiLSTM The bidirectional LSTM baseline from Williams et al. (2018).

Models

BiLSTM The bidirectional LSTM baseline from Williams et al. (2018).

ESIM The Enhanced Sequential Inference Model (Chen et al. 2016) is a hybrid TreeLSTM-based and biLSTM-based model that uses an inter-sentence attention mechanism to align words across sentences.

Models

BiLSTM The bidirectional LSTM baseline from Williams et al. (2018).

ESIM The Enhanced Sequential Inference Model (Chen et al. 2016) is a hybrid TreeLSTM-based and biLSTM-based model that uses an inter-sentence attention mechanism to align words across sentences.

BERT A Transformer model trained to do masked language modeling and next-sentence prediction (Devlin et al. 2019).

MoNLI as challenge dataset

MoNLI as challenge dataset

			No MoNLI fine-tuning		
Model	Input pretrain	NLI train data	SNLI	PMoNLI	NMoNLI
BiLSTM	GloVe	SNLI train	81.6	73.2	37.9

MoNLI as challenge dataset

			No MoNLI fine-tuning		
Model	Input pretrain	NLI train data	SNLI	PMoNLI	NMoNLI
BiLSTM	GloVe	SNLI train	81.6	73.2	37.9
ESIM	GloVe	SNLI train	87.9	86.6	39.4

MoNLI as challenge dataset

			No MoNLI fine-tuning		
Model	Input pretrain	NLI train data	SNLI	PMoNLI	NMoNLI
BiLSTM	GloVe	SNLI train	81.6	73.2	37.9
ESIM	GloVe	SNLI train	87.9	86.6	39.4
BERT	BERT	SNLI train	90.8	94.4	2.2

Model failure or dataset failure?

Liu et al. (2019)

"What should we conclude when a system fails on a challenge dataset? In some cases, a challenge might exploit blind spots in the design of the original dataset (*dataset weakness*). In others, the challenge might expose an inherent inability of a particular model family to handle certain natural language phenomena (*model weakness*). These are, of course, not mutually exclusive."

Negation coverage in SNLI and MultiNLI

1. SNLI: Only 38 examples have negated premise and hypothesis.
2. MultiNLI: 18K examples (≈4%) have negated premise and hypothesis, but few have the properties we are after.

A systematic generalization task

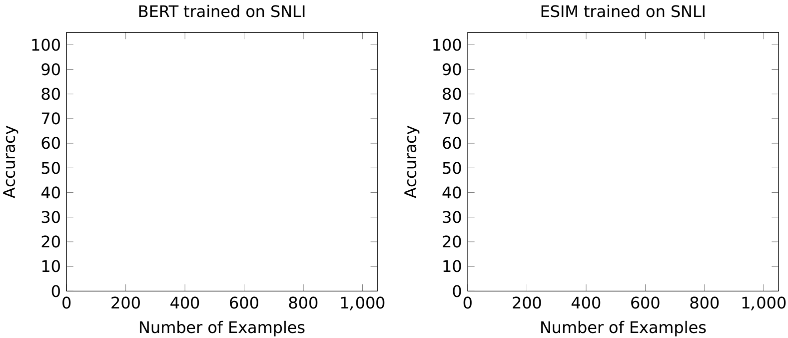
NMoNLI Train		NMoNLI Test	
person	198	dog	88
instrument	100	building	64
food	94	ball	28
machine	60	car	12
woman	58	mammal	4
music	52	animal	4
tree	52		
boat	46		
fruit	42		
produce	40		
fish	40		
plant	38		
jewelry	36		
anything	34		
hat	20		
man	20		
horse	16		
gun	12		
adult	10		
shirt	8		
shoe	6		
store	6		
cake	4		
individual	4		
clothe	2		
weapon	2		
creature	2		

A systematic generalization task

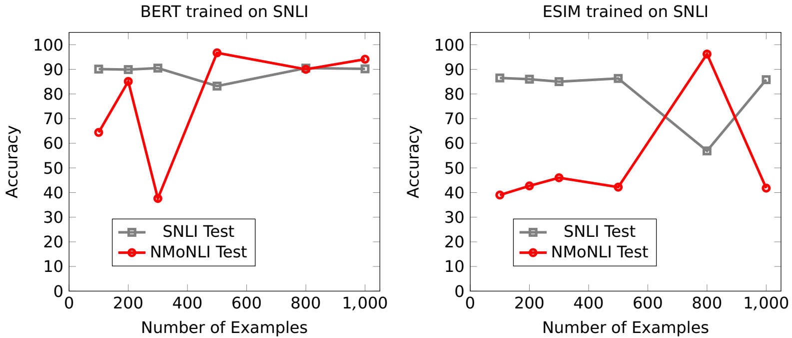
NMoNLI Train		NMoNLI Test	
person	198	dog	88
instrument	100	building	64
food	94	ball	28
machine	60	car	12
woman	58	mammal	4
music	52	animal	4
tree	52		
boat	46		
fruit	42		
produce	40		
fish	40		
plant	38		
jewelry	36		
anything	34		
hat	20		
man	20		
horse	16		
gun	12		
adult	10		
shirt	8		
shoe	6		
store	6		
cake	4		
individual	4		
clothe	2		
weapon	2		
creature	2		

Our models know these lexical relations (high Positive MoNLI accuracy) and will be compelled to combine this knowledge with what they learn about negation during Negative MoNLI fine-tuning.

Fine-tuning on Negative MoNLI



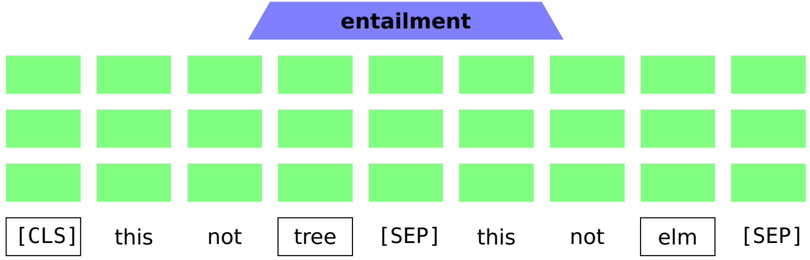
Fine-tuning on Negative MoNLI



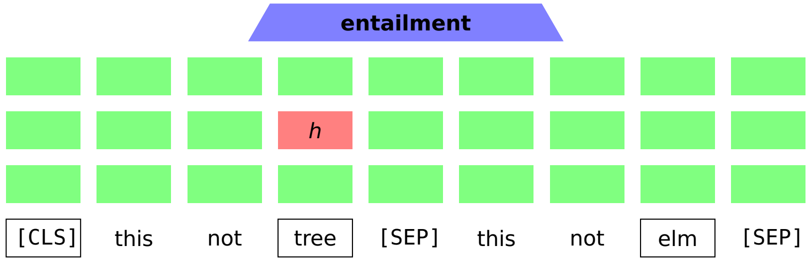
Fine-tuning results

Model	Input pretrain	NLI train data	No MoNLI fine-tuning SNLI	No MoNLI fine-tuning PMoNLI	No MoNLI fine-tuning NMoNLI	With NMoNLI fine-tuning SNLI	With NMoNLI fine-tuning NMoNLI
BiLSTM	GloVe	SNLI train	81.6	73.2	37.9	74.6	93.5
ESIM	GloVe	SNLI train	87.9	86.6	39.4	56.9	96.2
BERT	BERT	SNLI train	90.8	94.4	2.2	90.5	90.0

Focusing on the BERT model

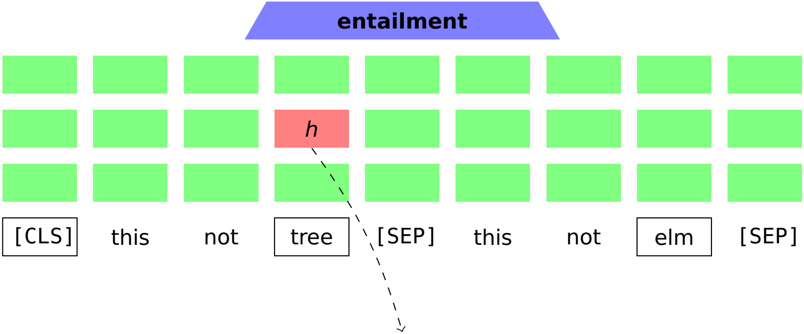


Probes



Hewitt and Liang 2019

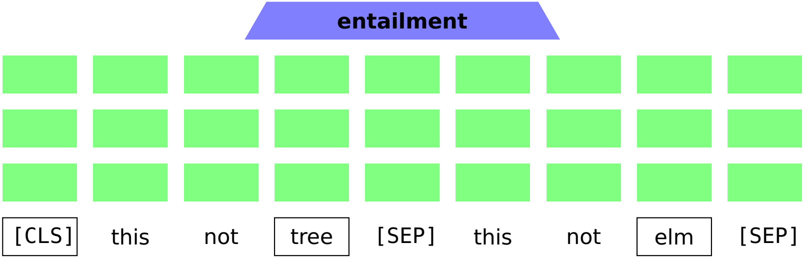
Probes



$$\text{SmallLinearModel}(h) = \text{get-lexrel}(\mathit{tree}, \mathit{elm})$$

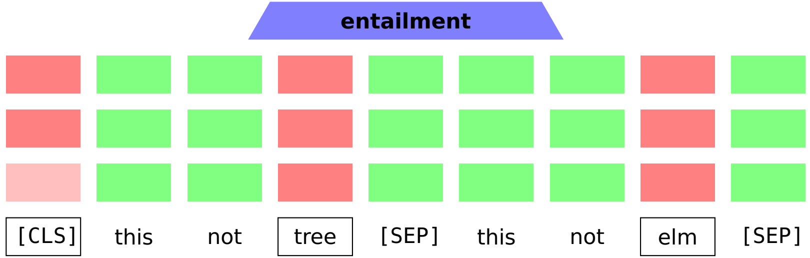
Hewitt and Liang 2019

Probe results for lexrel accuracy



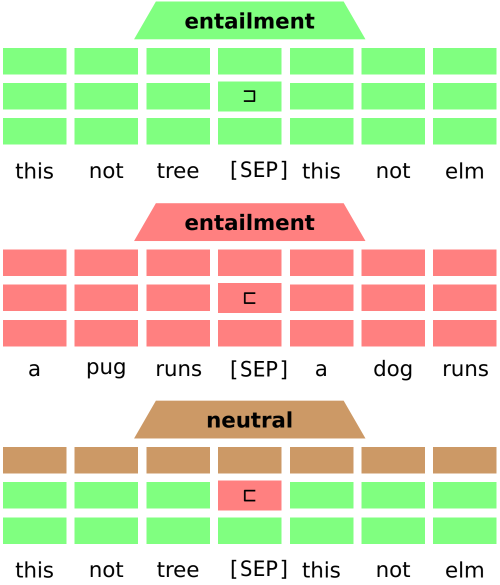
full probing results

Probe results for lexrel accuracy

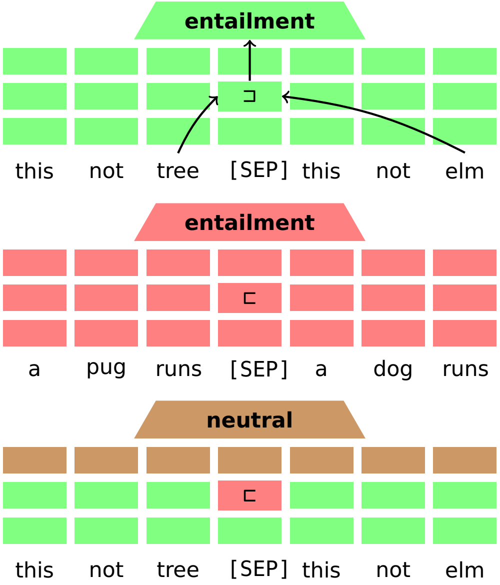


full probing results

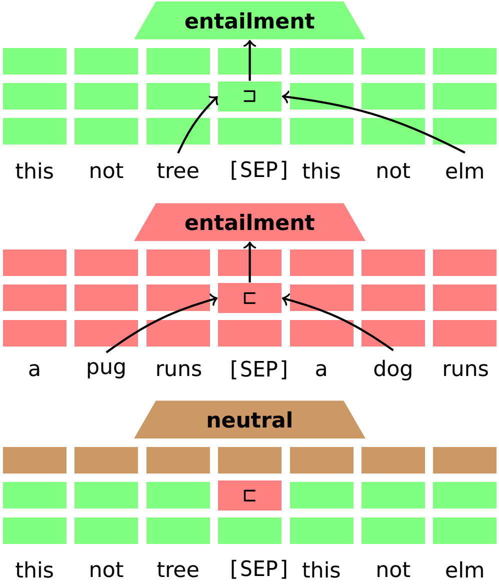
BERT NLI interventions



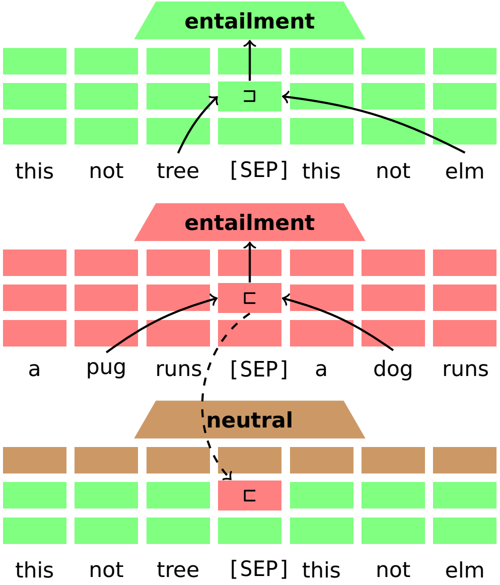
BERT NLI interventions



BERT NLI interventions



BERT NLI interventions



What it means for BERT to implement Infer

What it means for BERT to implement Infer

Infer(*example*)

lexrel ← get-lexrel(*example*)
if contains-not(*example*)
    return reverse(*lexrel*)
return *lexrel*

What it means for BERT to implement Infer

Infer(*example*)

1 *lexrel* ← get-lexrel(*example*)
2 **if** contains-not(*example*)
3 **return** reverse(*lexrel*)
4 **return** *lexrel*

$$\text{Infer}_{\mathit{lexrel}(i) \rightarrow \mathit{lexrel}(j)}(i) = \begin{cases} \text{Infer}(i) & \mathit{lexrel}(i) = \mathit{lexrel}(j) \\ \text{reverse}(\text{Infer}(i)) & \mathit{lexrel}(i) \neq \mathit{lexrel}(j) \end{cases}$$

What it means for BERT to implement Infer

Infer(*example*)

```
lexrel ← get-lexrel(example)
if contains-not(example)
        return reverse(lexrel)
return lexrel
```

$$\mathrm{Infer}_{lexrel(i) \rightarrow lexrel(j)}(i) =$$

$$\begin{cases} \mathrm{Infer}(i) & lexrel(i) = lexrel(j) \\ \mathrm{reverse}(\mathrm{Infer}(i)) & lexrel(i) \neq lexrel(j) \end{cases}$$

$$\mathrm{Infer}_{lexrel(i) \rightarrow lexrel(j)}(i) = \mathrm{BERT}_{L(i) \rightarrow L(j)}(i)$$

Methods and findings

Methods and findings

1. Find a useful intervention point.

Methods and findings

1. Find a useful intervention point.
2. Interchange interventions for every pair of examples at that site.

Methods and findings

1. Find a useful intervention point.
2. Interchange interventions for every pair of examples at that site.
3. Find clusters of examples in which BERT mimics the causal dynamics of Infer.

Methods and findings

1. Find a useful intervention point.
2. Interchange interventions for every pair of examples at that site.
3. Find clusters of examples in which BERT mimics the causal dynamics of Infer.
4. The largest subsets we found 98, 63, 47, and 37.

Methods and findings

1. Find a useful intervention point.
2. Interchange interventions for every pair of examples at that site.
3. Find clusters of examples in which BERT mimics the causal dynamics of Infer.
4. The largest subsets we found 98, 63, 47, and 37.
   a. For a random graph, the expected number of subsets larger than 20 is effectively 0.

Methods and findings

1. Find a useful intervention point.
2. Interchange interventions for every pair of examples at that site.
3. Find clusters of examples in which BERT mimics the causal dynamics of Infer.
4. The largest subsets we found 98, 63, 47, and 37.
   a. For a random graph, the expected number of subsets larger than 20 is effectively 0.
   b. If the site perfectly captured Infer, we would get a single huge cluster.

Largest exchangeable cluster

(cemetery,location)
(house,location) (den,location)
(ghetto,location) (backyard,location) (park,location)
(jungle,location) (meadow,location) (residence,location)
(laboratory,location) (playground,location) (studio,location)
(slum,location) (station,location) (farm,location)
(lab,location) (campsite,location)
(town,location) (lawn,location)

(saxophone,instrument) (flute,instrument)
(bass,instrument) (piano,instrument)
(violin,instrument) (tuba,instrument)
(harmonica,instrument)

(liquid,whiskey)
(liquid,margarita) (liquid,tequila)
(liquid,alcohol)

(woman,granny)
(woman,widow)

(dogs,huskies)
(dog,husky) (dog,chihuahua)
(dog,retriever) (dog,maltese)
(dog,terrier) (dog,pomeranian)

(beetle,insect)
(grasshopper,insect) (bee,insect)
(wasp,insect) (fly,insect) (cricket,insect)
(butterfly,insect) (bumblebee,insect)
(flea,insect) (roach,insect) (moth,insect)
(mosquito,insect)

(person,vegetarian) (person,lunatic)
(person,republican) (person,trooper)
(person,business)
(person,navigator)
(person,steward) (person,consultant)
(person,goalkeeper)
(person,farmer)
(person,sophomore) (person,housekeeper)
(person,cleaner) (person,physicist) (person,cop)
(person,cambodian) (person,detective)
(person,genius) (person,sergeant) (person,californian)
(person,doctor) (person,runner)

(hood,thing)
(nut,thing) (capsule,thing)
(pouch,thing) (structure,thing)
(root,thing) (nugget,thing)
(tube,thing)

(box,object)
(object,sweater) (hat,object)
(object,jacket) (toy,object)
(cane,object)

(water,rainwater)
(water,saltwater)

(sculptor,artist)

(berry,blueberry)

(tree,cypress)
(tree,magnolia) (trees,elms)
(tree,maple)

Which algorithm is BERT implementing then?

Which algorithm is BERT implementing then?

```
Infer(example)
lexrel ← get-lexrel(example)
if contains-not(example)
        return reverse(lexrel)
return lexrel
```

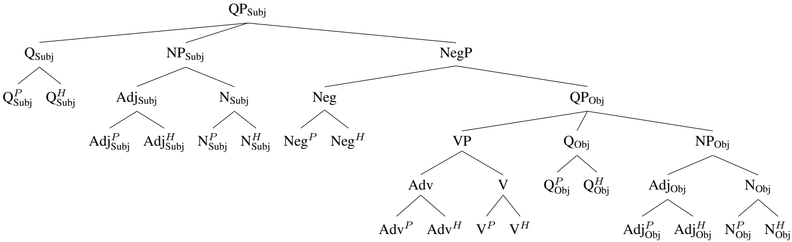
Which algorithm is BERT implementing then?

```
Infer(example)
lexrel ← get-lexrel(example)
if contains-not(example)
        return reverse(lexrel)
return lexrel
```

```
Infer(example)
if inCluster(C_1, example)
        lexrel_1 ← get-lexrel(example)
        if contains-not(example)
                return reverse(lexrel_1)
        return lexrel_1
if inCluster(C_2, example)
        lexrel_2 ← get-lexrel(example)
        if contains-not(example)
                return reverse(lexrel_2)
        return lexrel_2
if inCluster(C_3, example)
        lexrel_3 ← get-lexrel(example)
        if contains-not(example)
                return reverse(lexrel_3)
        return lexrel_3
...
```

Conclusion

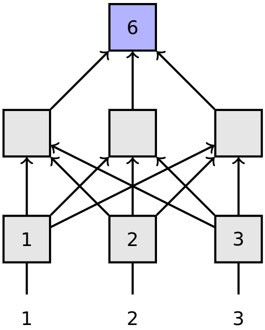
Compositional complexity



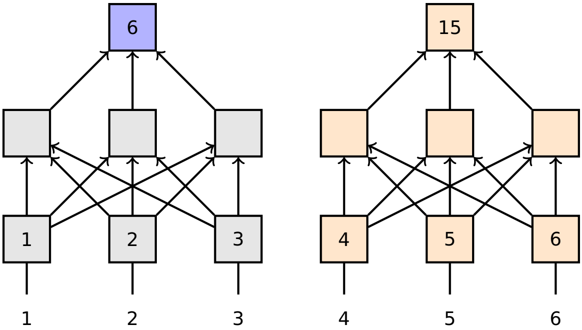
Geiger et al. 2021

Training models to conform to a hypothesis

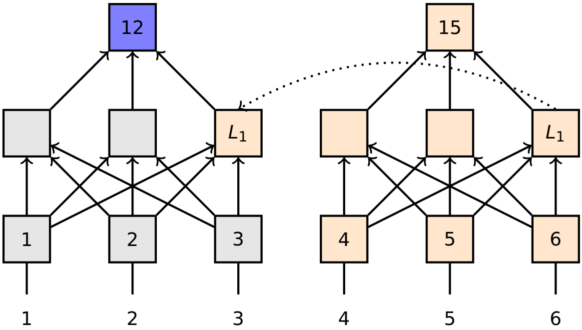
Training models to conform to a hypothesis



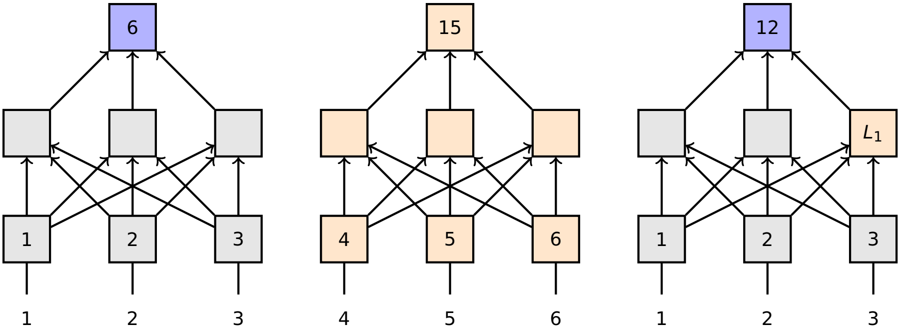
Training models to conform to a hypothesis



Training models to conform to a hypothesis



Training models to conform to a hypothesis



Open questions

Open questions

1. Can we more effectively leverage probes to find useful intervention points?

Open questions

1. Can we more effectively leverage probes to find useful intervention points?
2. What is the relationship between interchange interventions and integrated gradients?

Open questions

1. Can we more effectively leverage probes to find useful intervention points?
2. What is the relationship between interchange interventions and integrated gradients?
3. Can we characterize interchange interventions more generally so that they can be applied to more diverse models?

Open questions

1. Can we more effectively leverage probes to find useful intervention points?
2. What is the relationship between interchange interventions and integrated gradients?
3. Can we characterize interchange interventions more generally so that they can be applied to more diverse models?
4. Can interchanges be used to induce modularity during training?

Open questions

1. Can we more effectively leverage probes to find useful intervention points?
2. What is the relationship between interchange interventions and integrated gradients?
3. Can we characterize interchange interventions more generally so that they can be applied to more diverse models?
4. Can interchanges be used to induce modularity during training?

Thanks!

References I

Sander Beckers, Frederick Eberhardt, and Joseph Y. Halpern. 2020. Approximate causal abstractions. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 606–615. PMLR.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. 2016. Enhancing and combining sequential and tree LSTM for natural language inference. *CoRR*, abs/1609.06038.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9(0):160–175.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021. CausaLM: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386.

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

References II

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019. Inoculation by fine-tuning: A method for analyzing challenge datasets. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2171–2179, Minneapolis, Minnesota. Association for Computational Linguistics.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101. Association for Computational Linguistics.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Paul Soulos, R. Thomas McCoy, Tal Linzen, and Paul Smolensky. 2020. Discovering the compositional structure of vector representations with role learning networks. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 238–254, Online. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural nlp: The case of gender bias.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

References III

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

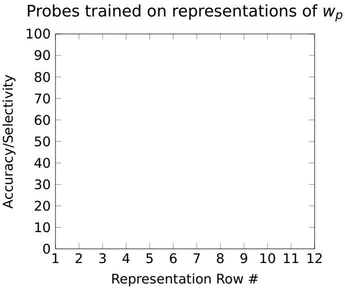
Integrated Gradients computation

$$\mathrm{IG}_i(M, x, x') = \overbrace{(x_i - x'_i)}^{5} \cdot \overbrace{\sum_{k=1}^{m}}^{4} \frac{\overbrace{\partial M(\overbrace{x' + \overbrace{\frac{k}{m}}^{1} \cdot (x - x')}^{2})}^{3}}{\partial x_i} \cdot \overbrace{\frac{1}{m}}^{4}$$

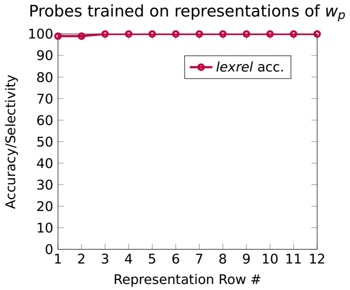
1. Generate $\alpha = [1, \ldots, m]$
2. Interpolate inputs between baseline x' and actual input x
3. Compute gradients for each interpolated input
4. Integral approximation through averaging
5. Scaling to remain in the space region as the original

Adapted from the TensorFlow integrated gradients tutorial

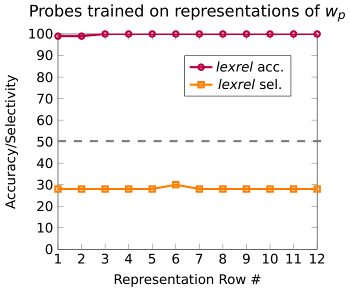
Probe results for lexrel accuracy



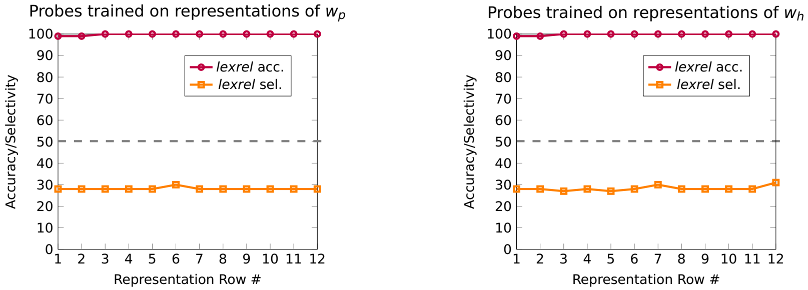
Probe results for lexrel accuracy



Probe results for lexrel accuracy

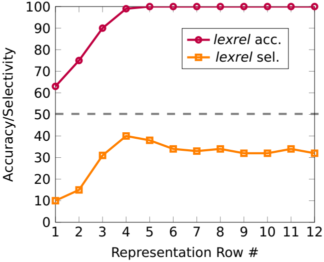


Probe results for lexrel accuracy

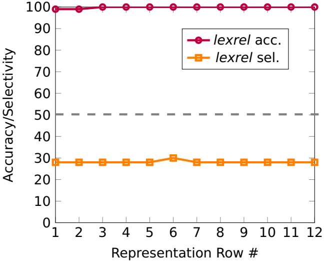


Probe results for lexrel accuracy

Probes trained on representations of [CLS]



Probes trained on representations of w_p



Probes trained on representations of w_h