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’
My ACL talk engaging with ‘NLP for Social Good’

Reliable characterizations of NLP systems as a social responsibility
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
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.
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.
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

<table>
<thead>
<tr>
<th>Characterize representations</th>
<th>Causal inference</th>
<th>Improved training</th>
</tr>
</thead>
</table>

### Probing

😊
Overview: Structural evaluation methods

Motivations

<table>
<thead>
<tr>
<th>Characterize representations</th>
<th>Causal inference</th>
<th>Improved training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probing</td>
<td>😊</td>
<td></td>
</tr>
<tr>
<td>Feature attribution</td>
<td>😐</td>
<td>😊</td>
</tr>
</tbody>
</table>

Overview: Structural evaluation methods

Motivations

<table>
<thead>
<tr>
<th>Characterize representations</th>
<th>Causal inference</th>
<th>Improved training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probing</td>
<td>😊</td>
<td></td>
</tr>
<tr>
<td>Feature attribution</td>
<td>😐</td>
<td>😊</td>
</tr>
</tbody>
</table>

Overview: Structural evaluation methods

Motivations

<table>
<thead>
<tr>
<th>Characterize representations</th>
<th>Causal inference</th>
<th>Improved training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probing</td>
<td>😊</td>
<td></td>
</tr>
<tr>
<td>Feature attribution</td>
<td>😐</td>
<td>😊</td>
</tr>
</tbody>
</table>

Overview: Structural evaluation methods

Motivations

<table>
<thead>
<tr>
<th>Characterize representations</th>
<th>Causal inference</th>
<th>Improved training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probing</td>
<td>😊</td>
<td></td>
</tr>
<tr>
<td>Feature attribution</td>
<td>😐</td>
<td>😊</td>
</tr>
</tbody>
</table>

Overview: Structural evaluation methods

Motivations

<table>
<thead>
<tr>
<th>Characterize representations</th>
<th>Causal inference</th>
<th>Improved training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probing</td>
<td>😊</td>
<td></td>
</tr>
<tr>
<td>Feature attribution</td>
<td>😐</td>
<td>😊</td>
</tr>
</tbody>
</table>
## Overview: Structural evaluation methods

### Motivations

<table>
<thead>
<tr>
<th>Characterize representations</th>
<th>Causal inference</th>
<th>Improved training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probing</td>
<td>😊</td>
<td>😊</td>
</tr>
<tr>
<td>Feature attribution</td>
<td>😐</td>
<td>😊</td>
</tr>
<tr>
<td>Causal abstraction</td>
<td>😊</td>
<td>😊</td>
</tr>
</tbody>
</table>
Overview: Structural evaluation methods

Motivations

<table>
<thead>
<tr>
<th>Characterize representations</th>
<th>Causal inference</th>
<th>Improved training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probing</td>
<td>😊</td>
<td>☹</td>
</tr>
<tr>
<td>Feature attribution</td>
<td>😕</td>
<td>😊</td>
</tr>
<tr>
<td>Causal abstraction</td>
<td>😊</td>
<td>😊</td>
</tr>
</tbody>
</table>

Case study: Monotonicity NLI
Motivations
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.”
Systematicity

Fodor and Pylyshyn (1988:37):
“What we mean when we say that linguistic capacities are \textit{systematic} is that the ability to produce/understand some sentences is \textit{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 ∼ 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 ~ 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 ~ 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 ~ 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 ~ 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.”

<table>
<thead>
<tr>
<th>Example</th>
<th>Gold</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>The bakery sells a mean apple pie.</td>
<td>pos</td>
<td>pos</td>
</tr>
<tr>
<td>They sell a mean apple pie.</td>
<td>pos</td>
<td>pos</td>
</tr>
<tr>
<td>She sells a mean apple pie.</td>
<td>pos</td>
<td>neg</td>
</tr>
<tr>
<td>He sells a mean apple pie.</td>
<td>pos</td>
<td>neg</td>
</tr>
</tbody>
</table>
Limits of behavioral testing
Limits of behavioral testing

Even/Odd Model 1

four

even
Limits of behavioral testing

Even/Odd Model 1

- four
- twenty one
- odd
Limits of behavioral testing
Limits of behavioral testing

Even/Odd Model 1

- four
- twenty one
- thirty two
- thirty six

even
Limits of behavioral testing

Even/Odd Model 1

four

twenty one

thirty two

thirty six

sixty three

odd
Limits of behavioral testing

Even/Odd Model 1

- four: even
- twenty one: odd
- thirty two: even
- thirty six: even
- sixty three: odd
- else: odd

odd
Limits of behavioral testing

Even/Odd Model 1

- four: even
- twenty one: odd
- thirty two: even
- thirty six: even
- sixty three: odd
- else: odd

odd
Limits of behavioral testing

Even/Odd Model 2

- four
- twenty one
- thirty two
- thirty six
- sixty three
- twenty two

even
Limits of behavioral testing

Even/Odd Model 2

- four
- twenty one
- thirty two
- thirty six
- sixty three
- twenty two
- five
- odd
Limits of behavioral testing

Even/Odd Model 2

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

odd
Limits of behavioral testing

Even/Odd Model 2

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

Even
Limits of behavioral testing

Even/Odd Model 2

\[d = \begin{cases} 
\text{one}: & \text{odd} \\
\text{two}: & \text{even} \\
\text{three}: & \text{odd} \\
\text{four}: & \text{even} \\
\text{five}: & \text{odd} \\
\text{six}: & \text{even} \\
\text{seven}: & \text{odd} \\
\text{eight}: & \text{even} \\
\text{nine}: & \text{odd} \\
\text{else}: & \text{odd} 
\end{cases}\]

return \(d[\text{input final token}]\)

even
Limits of behavioral testing

Even/Odd Model 2

\[ d = \begin{align*}
\text{one: odd} \\
\text{two: even} \\
\text{three: odd} \\
\text{four: even} \\
\text{five: odd} \\
\text{six: even} \\
\text{seven: odd} \\
\text{eight: even} \\
\text{nine: odd} \\
\text{else: odd}
\end{align*} \]

return \[ d[\text{input final token}] \]

\[ \text{odd} \]
Limits of behavioral testing

Even/Odd Model 3

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

\[ h = \text{SmallLinearModel}(X, y) \]
Core method

\[
\text{SmallLinearModel}(h) = \text{task}_1 \ X \ y \ h_1 \ \text{task}_1 \ y \ h_2 \ \text{task}_2 \ y \ h_3 \ \text{task}_3
\]
Core method

\[
\text{SmallLinearModel}(h) = \text{task}
\]
Core method

\[
\text{SmallLinearModel}(h) = \text{task}_1 \quad \text{task}_2 \quad \text{task}_3
\]

\[
h(X, y)
\]

\[
h_1 \quad \text{task}_y
\]
Core method

\[
\text{SmallLinearModel}(h) = \text{task}_y
\]

\[
h = \text{task}_y
\]

\[
w r r t m t w
\]

\[
X y
\]

\[
h_1 \quad \text{task}_y^1
\]

\[
h_2 \quad \text{task}_y^2
\]
Core method

\[
\text{SmallLinearModel}(h) = \text{task}_y
\]

\[
X, \quad y
\]

\[
h_1, \quad \text{task}_y^1
\]

\[
h_2, \quad \text{task}_y^2
\]

\[
h_3, \quad \text{task}_y^3
\]
Core method

\[
\text{SmallLinearModel}(h) = \text{task}_y
\]

\[
h_1 \quad \text{task}^1_y \\
\]

\[
h_2 \quad \text{task}^2_y \quad \text{SmallLinearModel}(X, y)
\]

\[
h_3 \quad \text{task}^3_y
\]
## Core method

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>t</th>
<th>w</th>
<th>w</th>
<th>w</th>
</tr>
</thead>
</table>
Probing BERT

Figure 1: Summary statistics on BERT-large. Columns on left show F1 dev-set scores for the baseline (\(P(0)\)) and full-model (\(P(L)\)) probes. Dark (blue) are the mixing weight center of gravity (Eq. 2); light (purple) are the expected layer from the cumulative scores (Eq. 4).

Evidence that the corresponding layer contains more information related to that particular task.

Center-of-Gravity. As a summary statistic, we define the mixing weight center of gravity as:

\[
\bar{E} = \frac{\sum_{\ell} E(s) \cdot \Delta(s)}{\sum_{\ell} \Delta(s)}
\]

This reflects the average layer attended to for each task; intuitively, we can interpret a higher value to mean that the information needed for that task is captured by higher layers.

3.2 Cumulative Scoring

We would like to estimate at which layer in the encoder a target \((s_1, s_2, \text{label})\) can be correctly predicted. Mixing weights cannot tell us this directly, because they are learned as parameters and do not correspond to a distribution over data. A naive classifier at a single layer cannot either, because information about a particular span may be spread out across several layers, and as observed in Peters et al. (2018b) the encoder may choose to discard information at higher layers.

To address this, we train a series of classifiers \(\{P(\ell)\}\) which use scalar mixing (Eq. 1) to attend to layer \(\ell\) as well as all previous layers. \(P(0)\) corresponds to a non-contextual baseline that uses only a bag of word(piece) embeddings, while \(P(L)\) corresponds to probing all layers of the BERT model. These classifiers are cumulative, in the sense that \(P(\ell+1)\) has a similar number of parameters but with access to strictly more information than \(P(\ell)\), and we see intuitively that performance (F1 score) generally increases as more layers are added.

We can then compute a differential score \(\Delta(s)\), which measures how much better we do on the probing task if we observe one additional encoder layer \(\ell\):

\[
\Delta(s) = \frac{\text{Score}(P(\ell+1)) - \text{Score}(P(\ell))}{\text{Score}(P(L)) - \text{Score}(P(0))}
\]

Expected Layer. Again, we compute a (pseudo) expectation over the differential scores as a summary statistic. To focus on the behavior of the contextual encoder layers, we omit the contribution of both the "trivial" examples resolved at layer 0, as well as the remaining headroom from 3.

Note that if a new layer provides distracting features, the probing model can overfit and performance can drop. We see this in particular in the last 1-2 layers (Figure 2).

This is not a true expectation because the F1 score is not an expectation over examples.
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!

$w = (x W_1; x W_2; x W_3)$

$W_1 = 1$

$W_2 = 1$

$W_3 = 0$

$x W_1 = 1$

$y W_2 = 1$

$w = 0$

$z W_3 = 1$
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!

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

$$w = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad (xW_1; xW_2; xW_3)w$$
## Summary

<table>
<thead>
<tr>
<th></th>
<th>Characterize representations</th>
<th>Causal inference</th>
<th>Improved training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probing</td>
<td>🤔</td>
<td>😊</td>
<td>😊</td>
</tr>
<tr>
<td>Feature attribution</td>
<td>🤔</td>
<td>😊</td>
<td>😊</td>
</tr>
<tr>
<td>Causal abstraction</td>
<td>😊</td>
<td>😊</td>
<td>😊</td>
</tr>
</tbody>
</table>
Feature attribution
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. . . .
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

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.

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.
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

\[ W_1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad W_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad W_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \]

\[ w = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \quad (xW_1; xW_2; xW_3)w \]

Sundararajan et al. 2017
Reliable insights about causal structure

\[ IG = 7 \quad IG = 8 \]

\[
W_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix} \quad W_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \quad W_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}
\]

\[
w = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \quad (xW_1; xW_2; xW_3)w
\]

Sundararajan et al. 2017
## Summary

<table>
<thead>
<tr>
<th></th>
<th>Characterize representations</th>
<th>Causal inference</th>
<th>Improved training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probing</td>
<td>😊</td>
<td>😊</td>
<td>😊</td>
</tr>
<tr>
<td>Feature attribution</td>
<td>😐</td>
<td>😊</td>
<td>😊</td>
</tr>
<tr>
<td>Causal abstraction</td>
<td>😊</td>
<td>😊</td>
<td>😊</td>
</tr>
</tbody>
</table>
Causal abstraction
## Recipe

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

*Geiger et al. 2020, 2021*
Recipe

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

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

<table>
<thead>
<tr>
<th>Characterize representations</th>
<th>Causal inference</th>
<th>Improved training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probing</td>
<td>😊</td>
<td></td>
</tr>
<tr>
<td>Feature attribution</td>
<td>😐</td>
<td>😊</td>
</tr>
<tr>
<td>Causal abstraction</td>
<td>😊</td>
<td>😊</td>
</tr>
</tbody>
</table>
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 ⊑ food
MoNLI dataset construction

Positive MoNLI (PMoNLI; 1,476 examples)

SNLI hypothesis (A) Food was served.
WordNet pizza ⊆ food
New example (B) Pizza was served.
MoNLI dataset construction

Positive MoNLI (PMoNLI; 1,476 examples)

SNLI hypothesis (A) Food was served.
WordNet pizza ⊑ 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 ⊏ 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 ⊏ plants
New example (B) The children are not holding flowers.

Negative MoNLI (A) entailment (B)
Negative MoNLI (B) neutral (A)
MoNLI monotonicity algorithm

<table>
<thead>
<tr>
<th>Monotonicity NLI</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoNLI Pizza was served. entailment Food was served. lexrel Pizza entailment Food</td>
<td>reverse lexrel neutral</td>
</tr>
<tr>
<td>MoNLI Pizza was not served. neutral Food was not served. lexrel Pizza entailment Food reverse lexrel neutral</td>
<td></td>
</tr>
</tbody>
</table>
MoNLI monotonicity algorithm

Infer(example)
1  \textit{lexrel} \leftarrow \text{get-lexrel}(example)
2  \textbf{if} \ \text{contains-not}(example)
3  \quad \textbf{return} \ \text{reverse}(\textit{lexrel})
4  \textbf{return} \ \textit{lexrel}
MoNLI monotonicity algorithm

Infer(example)

1. \texttt{lexrel} \leftarrow \texttt{get-lexrel(example)}
2. \textbf{if} \texttt{contains-not(example)}
3. \textbf{return} \texttt{reverse(lexrel)}
4. \textbf{return} \texttt{lexrel}

MoNLI
\texttt{lexrel}

Pizza was served. \hspace{1cm} \textbf{entailment} \hspace{1cm} Food was served.

Pizza \hspace{1cm} \textbf{entailment} \hspace{1cm} Food
MoNLI monotonicity algorithm

Infer(example)

1. $\text{lexrel} \leftarrow \text{get-lexrel(example)}$
2. if contains-not(example)
3. \text{return} reverse(lexrel)
4. \text{return} lexrel

MoNLI

\begin{align*}
\text{Pizza} \text{ was served.} & \quad \text{entailment} \quad \text{Food} \text{ was served.} \\
\text{Pizza} & \quad \text{entailment} \\
\text{Food} & \quad \text{Food}
\end{align*}

\text{Pizza} \text{ was not served.} & \quad \text{neutral} \quad \text{Food} \text{ was not served.} \\
\text{Pizza} & \quad \text{neutral} \\
\text{Food} & \quad \text{Food}
\end{align*}

reverse(lexrel)
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

<table>
<thead>
<tr>
<th>Model</th>
<th>Input pretrain</th>
<th>NLI train data</th>
<th>No MoNLI fine-tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiLSTM</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>SNLI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>81.6</td>
</tr>
</tbody>
</table>
MoNLI as challenge dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>Input pretrain</th>
<th>NLI train data</th>
<th>No MoNLI fine-tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiLSTM</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>81.6 73.2 37.9</td>
</tr>
<tr>
<td>ESIM</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>87.9 86.6 39.4</td>
</tr>
</tbody>
</table>
MoNLI as challenge dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>Input pretrain</th>
<th>NLI train data</th>
<th>No MoNLI fine-tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiLSTM</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>81.6 73.2 37.9</td>
</tr>
<tr>
<td>ESIM</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>87.9 86.6 39.4</td>
</tr>
<tr>
<td>BERT</td>
<td>BERT</td>
<td>SNLI train</td>
<td>90.8 94.4 2.2</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>NMoNLI Train</th>
<th>NMoNLI Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>person</td>
<td>dog</td>
</tr>
<tr>
<td>instrument</td>
<td>building</td>
</tr>
<tr>
<td>food</td>
<td>ball</td>
</tr>
<tr>
<td>machine</td>
<td>car</td>
</tr>
<tr>
<td>woman</td>
<td>mammal</td>
</tr>
<tr>
<td>music</td>
<td>animal</td>
</tr>
<tr>
<td>tree</td>
<td></td>
</tr>
<tr>
<td>boat</td>
<td></td>
</tr>
<tr>
<td>fruit</td>
<td></td>
</tr>
<tr>
<td>produce</td>
<td></td>
</tr>
<tr>
<td>fish</td>
<td></td>
</tr>
<tr>
<td>plant</td>
<td></td>
</tr>
<tr>
<td>jewelry</td>
<td></td>
</tr>
<tr>
<td>anything</td>
<td></td>
</tr>
<tr>
<td>hat</td>
<td></td>
</tr>
<tr>
<td>man</td>
<td></td>
</tr>
<tr>
<td>horse</td>
<td></td>
</tr>
<tr>
<td>gun</td>
<td></td>
</tr>
<tr>
<td>adult</td>
<td></td>
</tr>
<tr>
<td>shirt</td>
<td></td>
</tr>
<tr>
<td>shoe</td>
<td></td>
</tr>
<tr>
<td>store</td>
<td></td>
</tr>
<tr>
<td>cake</td>
<td></td>
</tr>
<tr>
<td>individual</td>
<td></td>
</tr>
<tr>
<td>clothe</td>
<td></td>
</tr>
<tr>
<td>weapon</td>
<td></td>
</tr>
<tr>
<td>creature</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
A systematic generalization task

<table>
<thead>
<tr>
<th>NMoNLI Train</th>
<th>NMoNLI Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>person</td>
<td>198</td>
</tr>
<tr>
<td>instrument</td>
<td>100</td>
</tr>
<tr>
<td>food</td>
<td>94</td>
</tr>
<tr>
<td>machine</td>
<td>60</td>
</tr>
<tr>
<td>woman</td>
<td>58</td>
</tr>
<tr>
<td>music</td>
<td>52</td>
</tr>
<tr>
<td>tree</td>
<td>52</td>
</tr>
<tr>
<td>boat</td>
<td>46</td>
</tr>
<tr>
<td>fruit</td>
<td>42</td>
</tr>
<tr>
<td>produce</td>
<td>40</td>
</tr>
<tr>
<td>fish</td>
<td>40</td>
</tr>
<tr>
<td>plant</td>
<td>38</td>
</tr>
<tr>
<td>jewelry</td>
<td>36</td>
</tr>
<tr>
<td>anything</td>
<td>34</td>
</tr>
<tr>
<td>hat</td>
<td>20</td>
</tr>
<tr>
<td>man</td>
<td>20</td>
</tr>
<tr>
<td>horse</td>
<td>16</td>
</tr>
<tr>
<td>gun</td>
<td>12</td>
</tr>
<tr>
<td>adult</td>
<td>10</td>
</tr>
<tr>
<td>shirt</td>
<td>8</td>
</tr>
<tr>
<td>shoe</td>
<td>6</td>
</tr>
<tr>
<td>store</td>
<td>6</td>
</tr>
<tr>
<td>cake</td>
<td>4</td>
</tr>
<tr>
<td>individual</td>
<td>4</td>
</tr>
<tr>
<td>clothe</td>
<td>2</td>
</tr>
<tr>
<td>weapon</td>
<td>2</td>
</tr>
<tr>
<td>creature</td>
<td>2</td>
</tr>
</tbody>
</table>

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

BERT trained on SNLI

ESIM trained on SNLI
Fine-tuning on Negative MoNLI

![Graph comparing the accuracy of BERT and ESIM trained on SNLI to SNLI Test and NMoNLI Test across varying number of examples.](image)
## Fine-tuning results

<table>
<thead>
<tr>
<th>Model</th>
<th>Input pretrain</th>
<th>NLI train data</th>
<th>No MoNLI fine-tuning</th>
<th>With NMoNLI fine-tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>SNLI  PMoNLI  NMoNLI</td>
<td>SNLI  NMoNLI</td>
</tr>
<tr>
<td>BiLSTM</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>81.6 73.2 37.9</td>
<td>74.6 93.5</td>
</tr>
<tr>
<td>ESIM</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>87.9 86.6 39.4</td>
<td>56.9 96.2</td>
</tr>
<tr>
<td>BERT</td>
<td>BERT</td>
<td>SNLI train</td>
<td>90.8 94.4 2.2</td>
<td>90.5 90.0</td>
</tr>
</tbody>
</table>
Focusing on the BERT model

entailment

[CLS] this not tree [SEP] this not elm [SEP]
Probes

entailment

[CLS]  this  not  tree  [SEP]  this  not  elm  [SEP]

Hewitt and Liang 2019
Probes

SmallLinearModel(h) = get-lexrel(tree, elm)

Hewitt and Liang 2019
Probe results for lexrel accuracy

entailment

[CLS] this not tree [SEP] this not elm [SEP]
Probe results for lexrel accuracy

entailment

[CLS] this not tree [SEP] this not elm [SEP]
BERT NLI interventions

### Entailment

- Input: `this not tree [SEP] this not elm`

- Intervention: `a pug runs [SEP] a dog runs`

### Neutral

- Input: `this not tree [SEP] this not elm`

- Intervention: `a pug runs [SEP] a dog runs`
BERT NLI interventions

entailment

this not tree [SEP] this not elm

entailment

a pug runs [SEP] a dog runs

neutral

this not tree [SEP] this not elm
BERT NLI interventions

**Overview**

- Motivations
- Probing
- Feature attribution
- Causal abstraction
- Monotonicity NLI
- Conclusion

**Probing**

- Feature attribution
- Causal abstraction
- Monotonicity NLI

**Feature attribution**

- BERT NLI interventions

**Causal abstraction**

- BERT NLI interventions

**Monotonicity NLI**

- BERT NLI interventions

**Conclusion**

- BERT NLI interventions

---

**Entailment**

```
this not tree [SEP] this not elm
```

**Entailment**

```
a pug runs [SEP] a dog runs
```

**Neutral**

```
this not tree [SEP] this not elm
```
BERT NLI interventions

**entailment**

```
this not tree [SEP] this not elm
```

**neutral**

```
a pug runs [SEP] a dog runs
```

This diagram illustrates the BERT NLI interventions for the entailment and neutral cases. The top part shows the entailment case with the original sentence 'a pug runs' being true under the revised sentence 'a dog runs'. The bottom part shows the neutral case with the original sentence 'this not tree' being false under the revised sentence 'this not elm'.
What it means for BERT to implement Infer
What it means for BERT to implement Infer

Infer(example)

1. \( \text{lexrel} \leftarrow \text{get-lexrel}(\text{example}) \)
2. \( \text{if contains-not}(\text{example}) \)
3. \( \text{return} \ \text{reverse}(\text{lexrel}) \)
4. \( \text{return} \ \text{lexrel} \)
What it means for BERT to implement Infer

Infer(example)

1. \[ \text{lexrel} \leftarrow \text{get-lexrel}(\text{example}) \]
2. \[ \text{if contains-not(}\text{example}) \]
3. \[ \text{return reverse(lexrel)} \]
4. \[ \text{return lexrel} \]

\[
\text{Infer}_{\text{lexrel}(i)\rightarrow\text{lexrel}(j)}(i) = \\
\begin{cases} 
\text{Infer}(i) & \text{lexrel}(i) = \text{lexrel}(j) \\
\text{reverse(}\text{Infer}(i)) & \text{lexrel}(i) \neq \text{lexrel}(j)
\end{cases}
\]
What it means for BERT to implement Infer

\( \text{Infer}(\text{example}) \)

1. \( \text{lexrel} \leftarrow \text{get-lexrel}(\text{example}) \)
2. \textbf{if} \ contains-not(\text{example})
3. \textbf{return} \ \text{reverse}(\text{lexrel})
4. \textbf{return} \ \text{lexrel}

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

\[
\text{Infer}_{\text{lexrel}(i) \rightarrow \text{lexrel}(j)}(i) = \text{BERT}_{L(i) \rightarrow L(j)}(i)
\]
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

Overview
Motivations
Probing
Feature attribution
Causal abstraction
Monotonicity NLI
Conclusion

- (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)
- (lab, location)
- (station, location)
- (farm, 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, steward)
- (person, consultant)
- (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)

1. \( \text{lexrel} \leftarrow \text{get-lexrel(example)} \)
2. \( \text{if contains-not(example)} \)
3. \( \text{return reverse(lexrel)} \)
4. \( \text{return lexrel} \)
Which algorithm is BERT implementing then?

\[
\text{Infer}(\text{example})
\]
1. \( \text{lexrel} \leftarrow \text{get-lexrel}(\text{example}) \)
2. \( \text{if} \ \text{contains-not}(\text{example}) \)
3. \( \text{return} \ \text{reverse}(\text{lexrel}) \)
4. \( \text{return} \ \text{lexrel} \)

\[
\text{Infer}(\text{example})
\]
1. \( \text{if} \ \text{inCluster}(C_1, \text{example}) \)
2. \( \text{lexrel}_1 \leftarrow \text{get-lexrel}(\text{example}) \)
3. \( \text{if} \ \text{contains-not}(\text{example}) \)
4. \( \text{return} \ \text{reverse}(\text{lexrel}_1) \)
5. \( \text{return} \ \text{lexrel}_1 \)
6. \( \text{if} \ \text{inCluster}(C_2, \text{example}) \)
7. \( \text{lexrel}_2 \leftarrow \text{get-lexrel}(\text{example}) \)
8. \( \text{if} \ \text{contains-not}(\text{example}) \)
9. \( \text{return} \ \text{reverse}(\text{lexrel}_2) \)
10. \( \text{return} \ \text{lexrel}_2 \)
11. \( \text{if} \ \text{inCluster}(C_3, \text{example}) \)
12. \( \text{lexrel}_3 \leftarrow \text{get-lexrel}(\text{example}) \)
13. \( \text{if} \ \text{contains-not}(\text{example}) \)
14. \( \text{return} \ \text{reverse}(\text{lexrel}_3) \)
15. \( \text{return} \ \text{lexrel}_3 \)
16. \( \ldots \)
Conclusion
Compositional complexity

Figure 2: The causal structure of the high-level natural logic causal model that performs inference on \textit{MQNLI}. The superscripts $P$ and $H$ stand for 'premise' and 'hypothesis' and the subscripts 'Obj' and 'Subj' stand for 'subject' and 'object'. The node labels are used to explain the experimental results in Section 5 and Section 5.2.

\textit{Geiger et al. 2021}
Training models to conform to a hypothesis
Training models to conform to a hypothesis

Diagram:

- Node 6
- Node 1
- Node 2
- Node 3

Connections:
- Node 1 to Node 6
- Node 2 to Node 6
- Node 3 to Node 6
Training models to conform to a hypothesis
Training models to conform to a hypothesis
Training models to conform to a hypothesis
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!
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


References

Integrated Gradients computation

\[ IG_i(M, x, x') = \left( x_i - x'_i \right) \cdot \sum_{k=1}^{m} \frac{\partial M(x' + \frac{k}{m} \cdot (x - x'))}{\partial x_i} \cdot \frac{1}{m} \]

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

Probes trained on representations of $w_P$

Accuracy/Selectivity vs. Representation Row #

- lexrel acc.
Probe results for lexrel accuracy

Probes trained on representations of $w_p$

Accuracy/Selectivity

Representation Row #
Probe results for lexrel accuracy

- **Probes trained on representations of** $w_p$
- **Probes trained on representations of** $w_h$
Probe results for lexrel accuracy

Probes trained on representations of [CLS]

Probes trained on representations of $w_p$

Probes trained on representations of $w_h$