Modular Computation
Geiger et al. 2020 & Parte 1984

Carina
Matthew
Yixuan
Hang
Outline

1. Monotonicity Reasoning (Hang)  11:35-11:50
2. Discussion  11:55-12:10
3. Geiger et al. 2020 (Yixuan)  12:10-12:30
4. Breakout Room + Discussion  12:40-12:55
5. **10-minute Break**  12:55-1:10
6. Compositionality + MCP (Carina)  1:10-1:25
7. Challenges (Matthew)  11:55-12:10
8. Breakout Room + Discussion  12:10-12:30
Question

How can we know the model is doing the linguistic task vs. learning linguistic knowledge/reasoning?
Monotonicity Reasoning

What is monotonicity?

Entailment

Negation
Paper Outline

1. Challenge Test Sets
2. Systematic Generalization Task
3. Probing
4. Intervention
MoNLI Dataset

Procedure

- Ensure the_hypernym_ / hyponym occurs in SNLI
- Ensure substitution generates a grammatically coherent sentence
- Generate one entailment and one neutral example

NMoNLI (1,202 examples)

PMoNLI (1,476 examples)

(A) The three children are not holding plants.

(B) The three children are not holding flowers.

This leads to two new MoNLI examples:

(A) entailment  (B)
(B) neutral  (A)
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Input pretraining</th>
<th>NLI train data</th>
<th>No MoNLI fine-tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>SNLI</td>
</tr>
<tr>
<td>CBOW</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>78.9</td>
</tr>
<tr>
<td>BiLSTM</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>81.6</td>
</tr>
<tr>
<td>ESIM</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>87.9</td>
</tr>
<tr>
<td>ESIM</td>
<td>GloVe</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>BERT</td>
<td>BERT</td>
<td>SNLI train</td>
<td>90.8</td>
</tr>
<tr>
<td>BERT</td>
<td>BERT</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>BERT</td>
<td></td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1: The results of our behavioral analysis. The columns labeled *No MoNLI fine-tuning* display the challenge test set results (Section 5.1), and the columns labeled *With MoNLI fine-tuning* display systematic generalization task results (Section 5.2). The numbers are accuracy values; all the datasets have balanced label distributions. Dashes mark experiments that would involve untrained NLI parameters due to training/fine-tuning set-up.
Observations on the Challenge Test Set

- No MoNLI fine-tuning,
  - Comparable results on PMoNLI
  - All models consistently fail on NMoNLI
  - 38 data points (ish) +++
- Combining MNLI + SNLI to have more negation examples yields a similar results
  - ~4% (18K) negation examples
A Systematic Generalization Task

Can models learn the general theory of entailment and negation beyond lexical relationship?

Experiment Design

1. train/test split: substitution words must be in disjoint
2. Inoculation on NMoNLI
Train/Test data split -- disjoint

<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>
</tbody>
</table>

Make sure there is no overlapping

Otherwise, models just memorize negation
Inoculation

Two stage fine-tuning on both SNLI and NMoNLI datasets respectively

A pre-trained model is further fine-tuned on different small amounts of adversarial data while performance on the original dataset and the adversarial dataset is tracked

- choose the highest average accuracy between both datasets
Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Input pretraining</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</td>
<td>PMoNLI</td>
</tr>
<tr>
<td>CBOW</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>78.9</td>
<td>64.6</td>
</tr>
<tr>
<td>BiLSTM</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>81.6</td>
<td>73.2</td>
</tr>
<tr>
<td>ESIM</td>
<td>GloVe</td>
<td>SNLI train</td>
<td>87.9</td>
<td>86.6</td>
</tr>
<tr>
<td>ESIM</td>
<td>GloVe</td>
<td></td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>ESIM</td>
<td></td>
<td></td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>BERT</td>
<td>BERT</td>
<td>SNLI train</td>
<td>90.8</td>
<td>94.4</td>
</tr>
<tr>
<td>BERT</td>
<td>BERT</td>
<td></td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>BERT</td>
<td></td>
<td></td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 1: The results of our behavioral analysis. The columns labeled No MoNLI fine-tuning display the challenge test set results (Section 5.1), and the columns labeled With MoNLI fine-tuning display systematic generalization task results (Section 5.2). The numbers are accuracy values; all the datasets have balanced label distributions. Dashes mark experiments that would involve untrained NLI parameters due to training/fine-tuning set-up.
Observations on systematic generalization

1. All models solved the task
2. Only BERT maintain high accuracy on SNLI
3. Removing pre-training on SNLI has little influence on results for BERT and ESIM
4. Removing pre-training for BERT and ESIM make them fail the task
   a. Note: BERT’s score is double that of ESIM with random initialization
5. Weak evidence from behavioral evaluation
1. Why does combining SNLI + MNLI **NOT** improve the model’s generalization on NMoNLI?
2. What would happen if we combine MoNLI and SNLI instead of doing the two-stage fine-tuning?
3. Do we need to create a specific adversarial dataset for each linguistic phenomenon of interest?
Structural Evaluation

Trying to determine internal dynamics to ‘conclusively evaluate systematicity’

- Probing & Intervention
  - Not well understood methodologies
  - Have to be tailored to the model

- BERT
  - Fine tuned on NMoNLI
  - Chosen because it does well without sacrificing SNLI performance
**INFER and Intuition**

- Question is if BERT (at the algorithmic level) implements lexical entailment and negation

  - **INFER**
    - Algorithmic description of entailment
    - `lexrel`: The lexical entailment relation between the substituted *words* in the MoNLI example

- Intuition behind storing and using `lexrel`
  - If BERT implements algorithm (loosely) then it will *store* a rep and *use* it
  - Storing $\rightarrow$ probe
  - Using $\rightarrow$ Intervention

```
INFER(MoNLlExample)
1  lexrel ← GET-LEX-REL(MoNLlExample)
2  if CONTAINS-NOT(MoNLlExample)
3      return REVERSE(lexrel)
4  return lexrel
```
Probing

- Idea: We want to see if lexrel (entailment relationship between words) is represented, and where
- BERT structure (12 layers of transformer encoders), get 1 vector rep/word per layer as a contextual embedding
  - Per word, this vector is not just info on the word like it would be for word2vec, heavily contextualized as BERT uses the words around it to inform
- **Assumption:** lexrel is stored in one of these vectors
- Specifically, one of the vectors for CLS, w_p, and w_h
- Try to find the vector which most likely stores this linguistic information
- Train the probe on all MoNLI
Probing and Selectivity

Takeaway (Hewitt and Manning 2019):

- Probes: use representations to predict linguistic properties
- Good probe: need high accuracy and high selectivity
- Probe design: use linear probes with fewer units

[CLS] I dance [SEP] I move [SEP]
Experiment

- Simple model with 4 Hidden Units
- Predict the value of $lexrel$ from the contextual embedding as the only input
  - Accuracy and selectivity are both plotted
Probe Results
Interpretation

- Why do the first couple of vectors for the [CLS] token not perform great?
- Essentially all vectors not 1-4 for the [CLS] token perform well for the task
  - *Lexrel* info is encapsulated in all of these places
Interventions

- Verifying whether the lexrel rep is used and where it is
- Want to show that the causal dynamics of INFER are mimicked by BERT
  - Not enough to show output of INFER and BERT match
  - lexrel is the only variable
  - Causal role can be determined with counterfactuals on how changing value of lexrel causes output to change

Example:

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

lexrel: tree is hypernym of elm

negation: true

INFER: entailment

Idea: if you flip lexrel, the output of INFER will change

INFER(MoNLI Example)

1 lexrel ← GET-LEX-REL(MoNLI Example)
2 if CONTAINS-NOT(MoNLI Example)
3 return REVERSE(lexrel)
4 return lexrel
Intervention Cont.

How would this work with BERT?

For a guess, $L$, of where the vector is and 2 examples, we can say that BERT mimics INFER on those 2 examples if the interchange behaves as expected.
Formalization and Experiment

Let $L$ be the hypothesis that $lexrel$ is stored at a specific location of 36, suppose $L$ with input $i$ is replaced with $L$ with input $j$, and feed $i$ into this modified bert. We call this

$$\text{BERT}_{L(i) \rightarrow L(j)}(i)$$

For some subset of MoNLI, if we believe BERT is storing value of $lexrel$ at $L$ and using info to make final prediction, then for all $i,j$ in $S$ we should have

$$\text{INFER}_{lexrel(i) \rightarrow lexrel(j)}(i) = \text{BERT}_{L(i) \rightarrow L(j)}(i)$$
Experiment

- For any pair of examples $i,j$, draw an edge between $i$ and $j$ if the interchange of the lexrel vector leads to the expected behavior
- Conducted interchange experiments at 36 different locations and chose most promising after partial graph
  - BERT^3 _wh
- 7 Million interchanges at this location
  - One for every pair of examples in MoNLI
- Greedy algorithm to discover large subsets of MoNLI where BERT mimics causal dynamics of INFER
Graph Visualization
Results

- Found large subsets of 98, 63, 47, and 37
- Expected number of subsets larger than 20 with this property if interchange had random effect is $10^{-8}$
- Same causal dynamics on 4 large subsets of MoNLI

Takeaway?

- Seems promising!
  - Interventions seem to show that the probability that BERT isn’t at some level implementing this algorithm is extremely low
- A lot of assumptions and shortcuts taken for the sake of reducing computation though
Breakout Rooms

10 min

- Did this approach show whether the model is able to just pass the entailment reasoning task or whether it was able to implement entailment reasoning?
- Does the probing/intervention approach seem promising to understand other linguistic tasks?
- Why weren't the clusters bigger? What assumptions made by the authors do you think were more/less valid or had bigger effects?
Compositionality

Partee 1984
Principle of Compositionality

The meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined.

> theory-dependent as highlighted terms can have different interpretations.
Montague's strong version of the compositionality principle (MCP)

Compositionality as a homomorphism between the syntactic and semantic algebra
What is an Algebra?

An algebra is a tuple $< A, f_1, \ldots, f_n >$ consisting of

- a set $A$
- one or more operations (functions) $f_1, \ldots, f_n$,

where $A$ is closed under each of $f_1, \ldots, f_n$.
What is an Algebra?

a. $< \mathbb{N}, +, \times >$
   - The natural numbers are \textit{closed} under addition and multiplication

Note: This is not an algebra: $< \mathbb{N}, +, \times, - >$, since $\mathbb{N}$ isn’t closed under subtraction

b. $< \mathbb{Z}, +, \times, - >$
   - The integers are \textit{closed} under addition, multiplication, and subtraction

Note: This is not an algebra: $< \mathbb{Z}, +, \times, -, \div >$, since $\mathbb{Z}$ isn’t closed under division.
Different Algebras Can Be Similar!

Key Observation
The algebras \(< \{1,0\}, \text{Conj}, \text{Disj} > \) and \(< \{\{a\}, \emptyset\}, \cap, \cup > \) are intuitively ‘similar’

\(< \{1,0\}, \text{Conj}, \text{Disj} > \)
- Conj(1,1) = 1
- Conj(1,0) = 0
- Conj(0,1) = 0
- Conj(0,0) = 0

\(< \{\{a\}, \emptyset\}, \cap, \cup > \)
- \(\cap(\{a\},\{a\}) = \{a\}\)
- \(\cap(\{a\}, \emptyset) = \emptyset\)
- \(\cap(\emptyset, \{a\}) = \emptyset\)
- \(\cap(\emptyset, \emptyset) = \emptyset\)

- \(\cup(\{a\},\{a\}) = \{a\}\)
- \(\cup(\{a\}, \emptyset) = \{a\}\)
- \(\cup(\emptyset, \{a\}) = \{a\}\)
- \(\cup(\emptyset, \emptyset) = \emptyset\)
Different Algebras Can Be Similar!

Key Observation
The algebras $<\{1,0\},\mathrm{Conj},\mathrm{Disj}>$ and $<\{\{a\},\emptyset\},\cap,\cup>$

<table>
<thead>
<tr>
<th></th>
<th>${1,0}$, $\mathrm{Conj}$, $\mathrm{Disj}$</th>
<th>${{a},\emptyset}$, $\cap$, $\cup$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathrm{Conj}(1,1)$</td>
<td>1</td>
<td>$\cap({a},{a}) = {a}$</td>
</tr>
<tr>
<td>$\mathrm{Conj}(1,0)$</td>
<td>0</td>
<td>$\cap({a},\emptyset) = \emptyset$</td>
</tr>
<tr>
<td>$\mathrm{Conj}(0,1)$</td>
<td>0</td>
<td>$\cap(\emptyset,{a}) = \emptyset$</td>
</tr>
<tr>
<td>$\mathrm{Conj}(0,0)$</td>
<td>0</td>
<td>$\cap(\emptyset,\emptyset) = \emptyset$</td>
</tr>
<tr>
<td>$\mathrm{Disj}(1,1)$</td>
<td>1</td>
<td>$\cup({a},{a}) = {a}$</td>
</tr>
<tr>
<td>$\mathrm{Disj}(1,0)$</td>
<td>1</td>
<td>$\cup({a},\emptyset) = {a}$</td>
</tr>
<tr>
<td>$\mathrm{Disj}(0,1)$</td>
<td>1</td>
<td>$\cup(\emptyset,{a}) = {a}$</td>
</tr>
<tr>
<td>$\mathrm{Disj}(0,0)$</td>
<td>0</td>
<td>$\cup(\emptyset,\emptyset) = \emptyset$</td>
</tr>
</tbody>
</table>

Intuitive similarity can be formalized as homomorphism between algebras!

$$h = \begin{pmatrix} 1 & \rightarrow & \{a\} \\ 0 & \rightarrow & \emptyset \end{pmatrix}$$

$$\mathrm{Conj} \approx \cap$$

$$h(\mathrm{Conj}(1,1)) = h(1) = \{a\} = \cap(\{a\},\{a\}) = \cap(h(1),h(1))$$
MCP Compositionality: Homomorphism Between Syntactic and Semantic Algebra

Syntax:
Arrangement of words and phrases into well-formed sentences

Semantics
Meaning of words, phrases, and sentences

\[
\left[ \text{a carrot} \right] = \text{carrot}
\]
MCP Compositionality: Homomorphism Between Syntactic and Semantic Algebra

Syntax:
\[ \varphi = \text{SynOp}(\psi, \chi) \]

Semantics
\[ [[\varphi]] = \text{SemOp}([[\psi]], [[\chi]]) \]
Building Blocks

[[Bill]] = \( \frac{\text{Bill}}{} \)

[[walks]] = function that takes one argument, x, and yields 1 iff x walks

[[Bill walks]] = 1 iff \( \frac{\text{Bill}}{\text{walks}} \) walks
Building Blocks

[[Bill]] = \overset{\text{NP}}{\overset{\text{VP}}{\overset{\text{S}}{\overset{\bigcirc}{\overset{\text{Bill}}{\text{walks}}}}}} = \text{function that takes one argument, } x, \text{ and yields 1 iff } x \text{ walks}

[[Bill walks]] = 1 \text{ iff } \overset{\text{walks}}{\overset{\text{Bill}}{\text{walks}}}

\[ S = \overset{\text{NP}}{\overset{\text{VP}}{\overset{\text{S}}{\overset{\bigcirc}{\overset{\text{Bill}}{\text{walks}}}}}} \]

\[ [\text{Bill}] = \overset{\text{NP}}{\overset{\text{VP}}{\overset{\text{S}}{\overset{\bigcirc}{\overset{\text{Bill}}{\text{walks}}}}}} \]

\[ [\text{walks}] = \text{function that takes one argument, } x, \text{ and yields 1 iff } x \text{ walks} \]

\[ [[\text{Bill walks}]] = 1 \text{ iff } \overset{\text{walks}}{\overset{\text{Bill}}{\text{walks}}} \]

\[ S_s. \quad [\text{A}] = [B] ( [C] ) \]

\[ G_s. \quad \text{Merge } (X,Y) \rightarrow \langle z, X, Y \rangle \text{ or } \overset{\text{Z}}{\overset{X}{\overset{Y}{\text{X Y}}}} \]
Montague's Paradise: Perfect Homomorphism

Syntax

\[
\langle \text{Pat likes Peter}_S, \text{Merge} \rangle \\
\langle \text{Pat}_N, \varnothing \rangle \ \langle \text{likes Peter}_P, \varnothing \rangle \\
\langle \text{likes}_V, \varnothing \rangle \\
\langle \text{Peter}_N, \varnothing \rangle
\]

Simplified semantics

\[
\langle [[\text{Pat likes Peter}]] = 1 \text{ iff } \text{Pat likes Peter}, \text{=} \rangle \\
\langle [[\text{likes} \text{Peter}]] = \lambda x. x \text{ likes Peter}, \text{=} \rangle \\
\langle [[\text{Pat}]] = \lambda y. \lambda x. x \text{ likes } y \rangle \\
\langle [[\text{Peter}]] = \text{Peter} \rangle
\]

Key features: Bottom-up!
Meanings of leaves are independent!
This Seems Familiar!

**Derivations:** Assumption of prior knowledge (oracle on derivation primitives)

**Compositionality:** homomorphism from inputs to representations

For any x with $D(x) = <D(x_a), D(x_b)> : f(x) = f(x_a) * f(x_b)$
Challenges overcome by Montague

- Structural ambiguity
  - Syntactic structure vs. Phonological Form (=spell-out)

- Context-(in)dependent meanings
  - Intensions (Senses) vs. Extensions (Denotations)
Structural Ambiguity

It’s not the case that Pat likes Peter and Megan smokes.

When is this sentence true?
Syntactically Ambiguous Languages

- Disambiguated expressions are the analysis trees themselves
- Ambiguation Relation $R$ maps analysis tree to string in the tree root

Example: "Pat likes Peter" is the phonological spell-out of the structure:

```
R

⟨Pat$_{NP}$, ∅⟩  ⟨likes Peter$_{VPiv}$, Merge⟩

   ⟨likes$_{VTv}$, ∅⟩  ⟨Peter$_{NP}$, ∅⟩
```

= Pat likes Peter

("Phonological spell-out" of the structure)
Disambiguation structures

Same spell-out, but different meaning!

It’s not the case that Pat likes Peter and Megan smokes.
Context (In)dependent Meaning: Why We Need Intensions I

*The president of the United States is blonde* = S

Truth of statement evaluated on 23-10-2020: [[S]] = True
Truth of statement evaluated on 23-10-2021: [[S]] = ？

How can there be different meanings?

**Intension/Sense:** \([\text{the president of the US}]^w = \text{the president of the US at } w \text{ (type } <s,e>)\)
> the presidential concept (function: context → president in that context)

**Extension/Denotation:** \([\text{the president of the US}]^{w_0} = \text{Donald Trump} \text{ (type } <e>)\)
> the presidential referent (current person picked out by that function)
Why We Need Intensions II

Toy context $w_0$: $[[\text{surgeon}]]^{w_0} = [[\text{violinist}]]^{w_0}$

(1) Jon is a skillful surgeon.
(2) Jon is a skillful violinist.
BUT: One can be true without the other! Why?

Solution:
- adjective denotes a function that applies to the \textit{intension} of the common noun phrase.
- $[[\text{surgeon}]]^w \neq [[\text{violinist}]]^w$ >> Intensions are clearly different!
- So (1) and (2) can have different truth values even if the \textit{extensions} pick out the same people!

Caveat: Implemented via more complex types of functions in Montague grammar! ($<s,<e,t>$, etc.)
Challenges to MCP
Generic Interpretation of Noun Phrases

A. The horse is widespread  
   - Generic

B. The horse is in the barn  
   - Non-generic

C. The horse is growing stronger  
   - Ambiguous
Where is the disambiguating information?

The horse is in the barn
Genericness as Local Ambiguity

The teacher was explaining the diesel engine.

**Non-Generic**

**Generic**
Things in the Wrong Place

An occasional sailor walked by
Things in the Wrong Place

An occasional sailor walked by occasionally
Constructions with Extra Meanings

A. Being a master of disguise, Bill would fool anyone

B. Wearing his new outfit, Bill would fool anyone

(Since Bill is)

Single Event

Two Events
Implicit Argument Differences

A. Every man in this room is a father  
Father of his own child

B. Every man in this room is an enemy  
Enemy of the same entity
“the horse” has two distinct senses. What are the implications of this for our models, especially regarding word embeddings?

Do we still have a robust definition of compositionality after accounting for these examples?

To what extent do we want our algorithms to model this principle of compositionality? How can we best adapt existing models to handle it?