CS 577: NATURAL LANGUAGE PROCESSING

Abulhair Saparov

Lecture 25: Semantics II

SEMANTICS

- We discussed how to represent meaning of natural language sentences.
- We looked at using logic to represent truth-functional meaning:
  - Propositional logic
  - First-order logic
- Today, let's talk about whether FOL is a good meaning representation.
  - Are there better meaning representations?
  - What makes one representation better than another?
- How do we model the conversion from natural language to logical form?
  - Semantic parsing

IS FOL A GOOD REPRESENTATION OF MEANING?

- Consider the sentence '`Mark drives`'.
  - We could represent it in logic as `drive(mark)`.
- But what about '`Mark drove`'?
  - Or '`Mark drives quickly`'?
- Maybe `drive_past(mark)` and `drive_quickly(mark)`?
  - But `drive`, `drive_past`, and `drive_quickly` are distinct concepts. (they are different symbols)
  - The meaning representation doesn't tell us that they're related.
- What about '`Mark drives more slowly than Alice`'?

IS FOL A GOOD REPRESENTATION OF MEANING?

- In first-order logic, we can only have predicates that take objects as arguments (i.e., variables or constants).
  - `drive(mark)` is allowed.
  - `quickly(drive)` is not allowed since drive is a predicate.
- In second-order logic, there are two kinds of predicates:
  - First-order predicates that only take objects as arguments (as in FOL).
  - Second-order predicates that take objects or 1st order predicates as arguments.
- Possible logical form assignments in 2nd order logic:
  - `'Mark drove' -> past(drive)(mark)`
  - `'Mark drives quickly' -> quickly(drive)(mark)`

SECOND-ORDER LOGIC

- ‘Mark drives more slowly than Alice’
  - `slower_than(drive,alice)(mark)`
- 2^{nd} order logic also enables quantification over first-order predicates.
  - ‘Mark and Alice are siblings.’
    - `siblings(mark,alice)`
  - ‘Mark and Alice are related.’
    - `∃R(R(mark,alice))`

HIGHER-ORDER LOGICS

- Why stop at second-order predicates?
- Why not allow second-order predicates to be arguments to other predicates?
  - I.e., third-order predicates?
- This is third-order logic.
- In general, n^{th} order logic is one where $(n-1)^{\text{th}}$-order predicates can be arguments of other predicates,
  - And we can quantify over $(n-1)^{\text{th}}$-order predicates.
- Taking the limit as n -> ∞, we obtain higher-order logic.

HIGHER-ORDER LOGIC

- In higher-order logic (HOL), there is no fundamental distinction between predicates (of any order) and constants/objects.
  - Everything is a term.
- In, HOL we can define new functions using lambda abstraction:
  - $\lambda x(2*x)$ is a function that takes one argument x, and returns $2*x$.
  - We can use it like any other function:
    - $(\lambda x(2*x))(4) = 8$
  - $\lambda x.\lambda y(x + y)$ just adds its two inputs (it has arity 2).
  - $\lambda x.\lambda y(x = y)$ returns true if and only if x and y are the same.
  - $\lambda f.\forall \mathrm{x}.\forall \mathrm{y}(f(\mathrm{x},\mathrm{y}) \texttt{->} f(\mathrm{y},\mathrm{x}))$ is true iff f is a symmetric function.
- This language is also called lambda calculus.

HIGHER-ORDER LOGIC

- Lambda calculus is very expressive.
- Any computable function can be written as a lambda calculus expression.
  - For any Turing machine, there is a lambda calculus expression that computes the same thing.
  - For any lambda calculus expression, there is a Turing machine that computes the same thing.

    (Church-Turing thesis)
- There are richer logics beyond HOL, such as type theories.

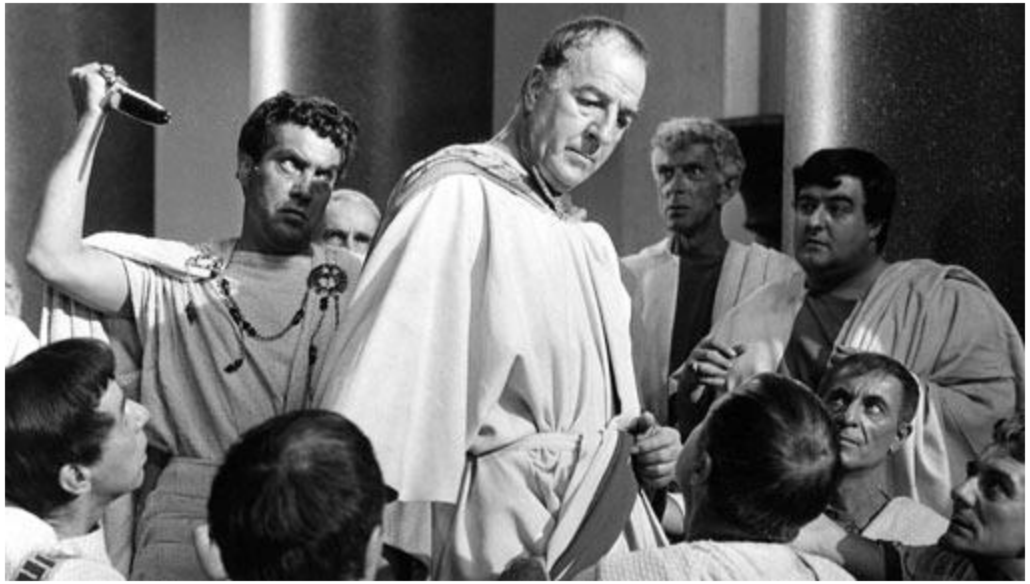
IS HIGHER-ORDER LOGIC NECESSARY FOR NATURAL LANGUAGE?

- Are there other ways we can represent the meaning of tense or adverbs using only first-order logic?
- We'll work with the following running example:
  - '`Brutus stabs Caesar`'
- Try first-order logic:
  - `stab(brutus,caesar)`

[Sayeed, Neo-Davidsonian semantics, 2012; F. H. Fuger, La muerte de Julio César, 1818]

REPRESENTING THE MEANING OF SENTENCES

- That was easy.
- Let’s make it more complicated:
  - ‘`Brutus stabs Caesar with a knife`’



[Sayeed, Neo-Davidsonian semantics, 2012; Shakespeare’s Julius Caesar, dir. Joseph L. Mankiewicz, 1953]

REPRESENTING THE MEANING OF SENTENCES

- That was easy.
- Let's make it more complicated:
  - '`Brutus stabs Caesar with a knife`'
- The knife is a participant in the action.
  - It's an instrument.
- Maybe: Make it an argument of the `stab` predicate?
  - `stab(brutus,caesar,knife)`
- Well, '`knife`' itself is not a singular concept.
  - Rather, it's an instance of an object that has type `knife`.
  - `∃k(knife(k) & stab(brutus,caesar,k))`

[Sayeed, Neo-Davidsonian semantics, 2012]

REPRESENTING THE MEANING OF SENTENCES

- 'Brutus stabs Caesar in the agora'

[Sayeed, Neo-Davidsonian semantics, 2012; Gérôme, The Death of Caesar, 1867]

REPRESENTING THE MEANING OF SENTENCES

- ‘`Brutus stabs Caesar in the agora`’
- The agora is also a participant in the action.
- So maybe: `stab(brutus,caesar,agora)`?
- But the agora is not participating in the same way as the knife.
  - The agora is a location, whereas the knife is an instrument.
  - So the third argument of `stab` is ambiguous.
- Or add a fourth argument:
  - `stab(brutus,caesar,_,agora)`

[Sayeed, Neo-Davidsonian semantics, 2012]

REPRESENTING THE MEANING OF SENTENCES

- What if we split up the predicate?
  - `stab(brutus,caesar) & in(agora)`
  - This is a bit better.
- ‘Brutus stabs Caesar with a knife in the agora.’
  - `stab(brutus,caesar) & with(knife) & in(agora)`
  - `∃k(knife(k) & stab(brutus,caesar) & with(k) & in(agora))`
- ‘Brutus stabs Caesar with a knife in the agora and twists it hard.’
  - `∃k(knife(k) & stab(brutus,caesar) & with(k) & in(agora) & twists(brutus,k) & hard)`
  - It’s difficult to represent the adverb `hard`.

[Sayeed, Neo-Davidsonian semantics, 2012]

REPRESENTING THE MEANING OF SENTENCES

- The main difficulty is referring to predicates.
  - First-order logic does not allow us to refer to predicates.
  - In natural language, there is very rich variety of ways to refer to predicates.
    - Adverbial modifiers, relative clauses, etc.
- It's easy to represent modifiers of nouns in FOL.
  - '`some clever driver in America`'
  - `∃x(driver(x) & clever(x) & location(x,america))`
- Can we borrow this idea for verbs?
  - '`Bob drives cleverly in America`'

[Sayeed, Neo-Davidsonian semantics, 2012]

DAVIDSON'S SOLUTION

- Davidson (1989) quoted in Maienborn (2010):
  "Adverbial modification is thus seen to be logically on a par with adjectival modification: what adverbial clauses modify is not verbs but the events that certain verbs introduce."
- A verb (e.g., '`stab`') is actually a description of an event.
  - An event is an object rather than a predicate.
  - We can quantify over events,
  - Just as we can quantify over any other object.

[Sayeed, Neo-Davidsonian semantics, 2012]

DAVIDSON'S SOLUTION

- 'Brutus stabs Caesar with a knife in the agora.'
  - ∃k(knife(k)
    & ∃e(stab(e,brutus,caesar) & with(e,k) & location(e,agora))
)
- We now have a flexible way to refer to the event of stabbing.
- 'Brutus stabs Caesar with a knife in the agora and twists it hard.'
  - ∃k(knife(k)
    & ∃e(stab(e,brutus,caesar) & with(e,k) & location(e,agora)
    & ∃t(twist(t,brutus,k) & hard(t))
)
)

[Sayeed, Neo-Davidsonian semantics, 2012]

EVENT SEMANTICS

- This idea is called event semantics (or Davidsonian semantics; Davidson, 1967).
  - ‘`Mark drove`’ -> `∃d(drive(d,mark) & past(d))`
  - ‘`Mark drives quickly`’ -> `∃d(drive(d,mark) & quickly(d))`
- Parsons (1990) proposed an updated form where thematic roles (or semantic roles) are explicit.
- In ‘`Brutus stabbed Caesar`’ (active voice)
  - The subject is Brutus, and the object is Caesar
- In ‘`Caesar was stabbed by Brutus`’ (passive voice)
  - The subject is Caesar, and the object is Brutus
- Even if the grammatical roles have switched, the semantic roles have not!

SEMANTIC ROLES

- Every event has a set of semantic or thematic roles.
- The exact set of roles is debated, but here are some candidates:
  - Agent: the performer of the action (e.g., '`Brutus stabs`')
  - Patient: the thing undergoing the action that changes state (e.g., '`Brutus stabs Caesar`')
  - Theme: the thing undergoing the action but does not change state (e.g., '`I gave them the food`')
  - Instrument
  - Location
  - Time
  - etc…

SEMANTIC ROLE LABELING

- Semantic role labeling is the task of identifying the semantic roles for a given sentence and verb.
  - Sometimes the verb is not specified.
- Example input:

  ‘`The batter hit the ball yesterday.`’

- Example output:

  ‘$[_{\texttt{agent}}\texttt{The batter}]$ `hit` $[_{\texttt{patient}}\texttt{the ball}]$ $[_{\texttt{time}}\texttt{yesterday}]$.’

- Not really full compositional semantic parsing,
  - But a “shallow” form of it.
  - Can be thought of as something between syntactic and semantic parsing.

EVENT SEMANTICS

- '`Caesar was stabbed by Brutus`'
  - Caesar is the patient (or theme)
  - Brutus is the agent
- In Davidsonian semantics, we would write this as:
  - `∃e.stab(e,brutus,caesar)`
  - But how would we write the meaning of '`Caesar was stabbed`'?
- In neo-Davidsonian semantics, we separate the semantic roles explicitly:
  - `∃e(stab(e) & agent(e,brutus) & patient(e,caesar))`
  - `∃e(stab(e) & arg1(e,brutus) & arg2(e,caesar))`
  - `∃e(stab(e) & arg1(e)=brutus & arg2(e)=caesar)`

LOGICAL FORMALISMS

- First-order logic is not just one monolithic logical formalism.
  - There are different ways we can use FOL to represent meaning.
  - Each with advantages and disadvantages.
- What makes a good logical formalism/meaning representation?
  - Coverage:
    - If there are two sentences with different meanings, they should have different logical forms.
    - If there are multiple readings/interpretations of the same sentence, there should be a logical form for each reading.

SEMANTIC AMBIGUITY

- We have seen ambiguity in syntax:
  - ‘`Sally caught a butterfly with a net.`’
  - ‘`Sally caught a butterfly with a stripe.`’
- We can also have ambiguity in semantics:
  - ‘`The trophy didn’t fit in the suitcase because it’s too big.`’
  - ‘`The trophy didn’t fit in the suitcase because it’s too small.`’
  - These sentences have the same syntactic structure.
  - Coreference resolution:
    - Does ‘`it`’ refer to the same thing as ‘`trophy`’?
    - Or to the same thing as ‘`suitcase`’?
- A good logical formalism will have 2 LFs for each sentence.

SEMANTIC AMBIGUITY

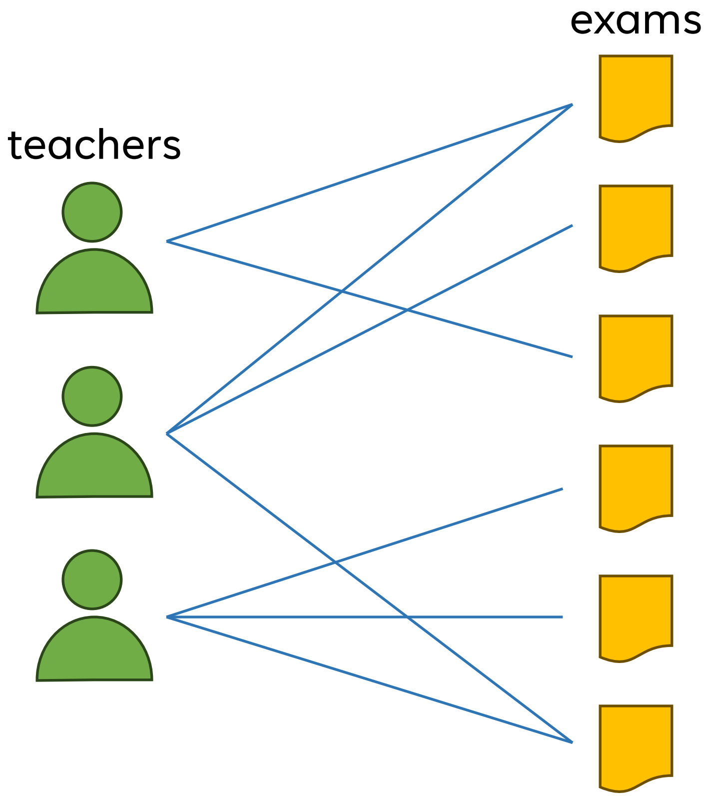
- Lexical ambiguity is another source of ambiguity in semantics:
  - ‘`I went to the bank to withdraw cash.`’
  - ‘`I went to the bank to catch some fish.`’
- The word ‘`bank`’ refers to the financial institution sense in the first sentence.
  - Whereas it refers to a riverbank in the second.
- A good logical formalism will produce two logical forms for each of the above two sentences:
  - 1 LF for the financial institution sense,
  - And 1 LF for the riverbank sense.
- The logical formalism itself has no background knowledge of the world.
  - Maybe there’s a possible world where you withdraw money near rivers.

SEMANTIC AMBIGUITY

- Quantifier scope ambiguity is another source of ambiguity in semantics:
  - '`Every dog chases a cat.`'
- There are two valid readings:
  - `∀d(dog(d) -> ∃c(cat(c) & chase(d,c)))`
  - `∃c(cat(c) & ∀d(dog(d) -> chase(d,c)))`
- The only difference is the order of the quantifiers,
  - But there is a stark difference in meaning.
- Negation scope ambiguity: '`All that glitters is not gold.`'
  - `∀g(glitter(g) -> ¬gold(g))`
  - `¬∀g(glitter(g) -> gold(g))`

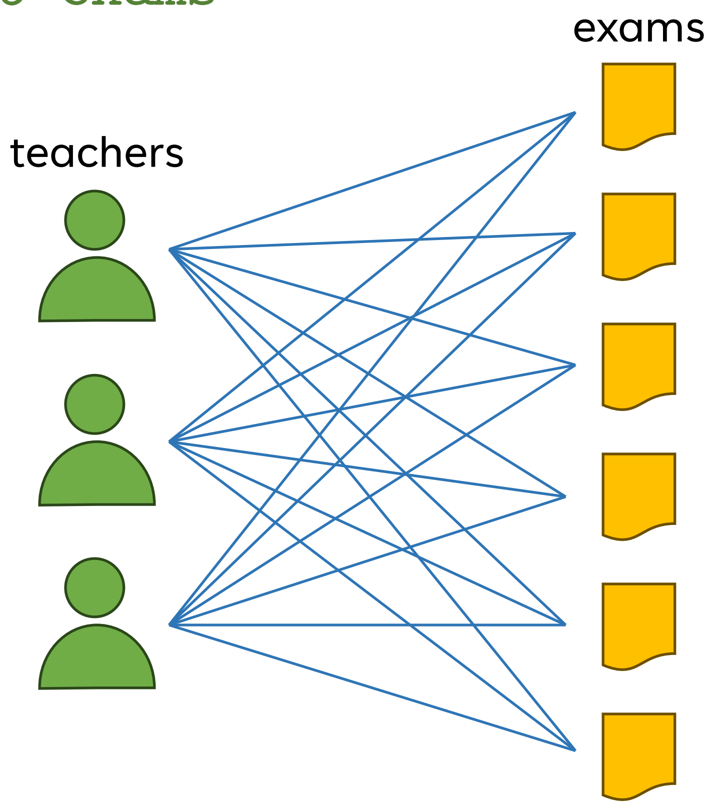
SEMANTIC AMBIGUITY

- More extreme example of quantifier scope ambiguity:
  - ‘`3 teachers graded 6 exams`’



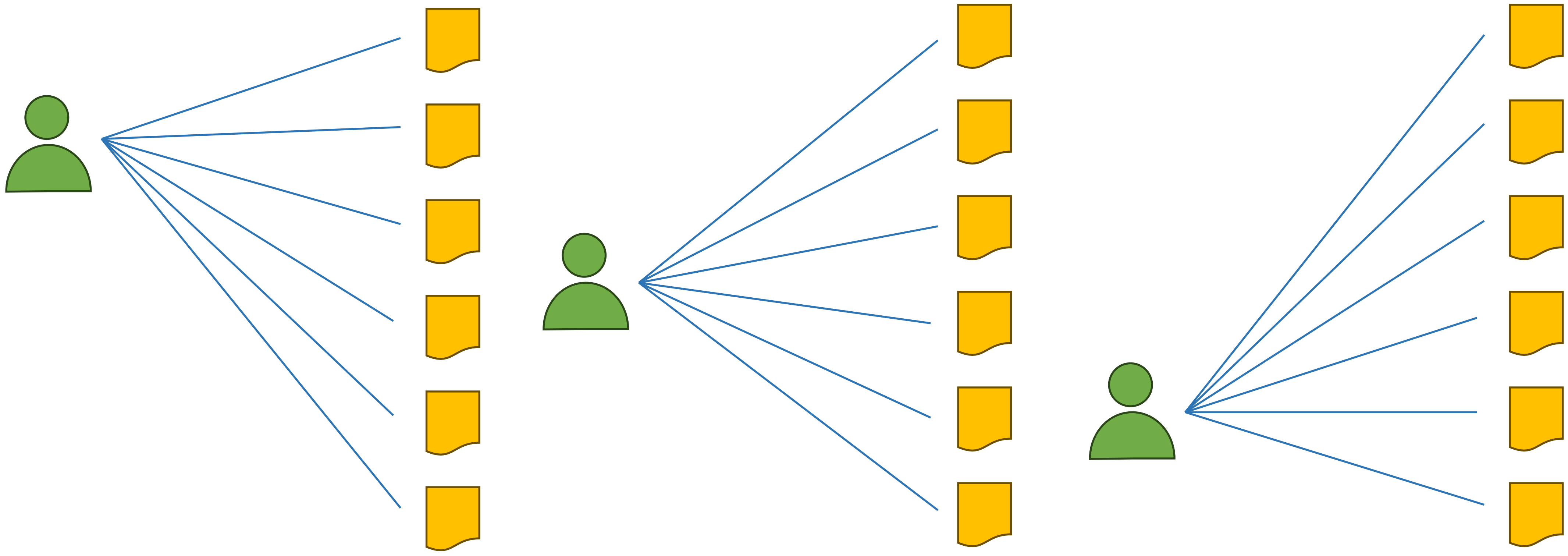
SEMANTIC AMBIGUITY

- More extreme example of quantifier scope ambiguity:
  - ‘3 teachers graded 6 exams’



SEMANTIC AMBIGUITY

- More extreme example of quantifier scope ambiguity:
  - `'3 teachers graded 6 exams'`



LOGICAL FORMALISMS

- What makes a good logical formalism/meaning representation?
  - Coverage
  - Language-independence:
    - This depends on the application.
    - If we wish to model translation from one natural language to another, Where the meaning is language-independent,
    - Ideally, two sentences in different languages but with the same meaning should map to the same logical form.

LOGICAL FORMALISMS

- What makes a good logical formalism/meaning representation?
  - Coverage
  - Language-independence
  - Uniqueness:
    - Two sentences with the same meaning should map to the same LF.
    - Not always needed.
  - Amenable for reasoning:
    - If we want to reason over the logical forms,
    - It would help to have a set of inference rules with which we can construct proofs.

LOGICAL FORMALISMS

- What makes a good logical formalism/meaning representation?
  - Coverage
  - Language-independence
  - Uniqueness
  - Amenable for reasoning
  - Easy to parse:
    - It shouldn't be too difficult to parse from natural language into LF.
    - If the logical form is more similar to the natural language utterance, semantic parsing is easier.
    - We will discuss semantic parsing in more detail later.

LOGICAL FORMALISMS

- What makes a good logical formalism/meaning representation?
  - Coverage
  - Language-independence
  - Uniqueness
  - Amenable for reasoning
  - Easy to parse
  - Compositional:
    - The logical form of larger phrases should be composed of the logical forms of subphrases.

COMPOSITIONALITY

- Compositionality is also relevant in syntactic analysis.
  - The syntax tree of a larger phrase is composed of the syntax trees of its constituents.
  - Grammars with recursion enable syntactic compositionality.
  - E.g., context-free grammars.
- Semantic compositionality:
  - ‘`a barn`’ -> `∃b.barn(b)`
  - ‘`a dog`’ -> `∃d.dog(d)`
  - ‘`in a barn`’ -> `∃b(barn(b) & location(_,b))`
  - ‘`a dog in a barn`’ -> `∃d(dog(d) & ∃b(barn(b) & location(d,b)))`
- Exceptions: Idioms such as ‘`break a leg`’, ‘`on the other hand`’, etc.

COMPOSITIONALITY

- Compositionality is very useful for efficient parsing.
  - We can parse a sentence by first parsing smaller phrases.
  - Enables dynamic programming approaches.
- Compositionality is also important for generalization.
  - We can predict the syntactic/semantic structure of larger phrases/sentences based on its smaller constituent phrases.
  - Compositional generalization :
    - If a model correctly "understands" some phrases/sentences,

      How well does it "understand" larger phrases/sentences that contain those phrases?

COMPOSITIONAL GENERALIZATION

- Kim and Linzen (2020) introduced a dataset called COGS to test the semantic compositional generalization ability of NLP models.
- Some examples from COGS:

TRAINING

[[The girl]] = $\iota x.\, girl'(x)$, [[The cat]] = $\boldsymbol{\iota x.\, cat'(x)}$, [[The boy]] = $\iota x.\, boy'(x)$

[[The cat loves the girl]] = $love'(\boldsymbol{\iota x.\, cat(x)}, \iota x.\, girl'(x))$

[[The hedgehog sees the cat]] = $see'(\iota x.\, hedgehog'(x), \boldsymbol{\iota x.\, cat'(x)})$

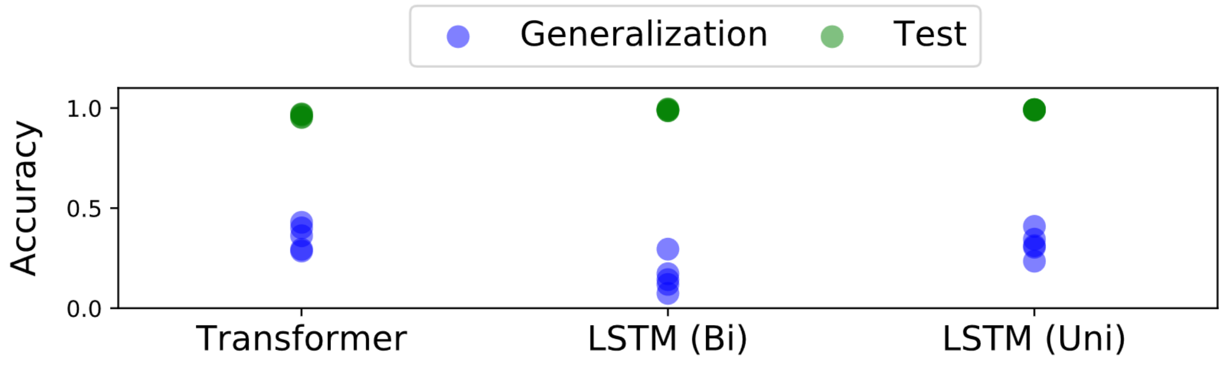
GENERALIZATION

[[The boy loves the hedgehog]] = $love'(\iota x.\, boy'(x), \boldsymbol{\iota x.\, hedgehog(x)})$

COMPOSITIONAL GENERALIZATION

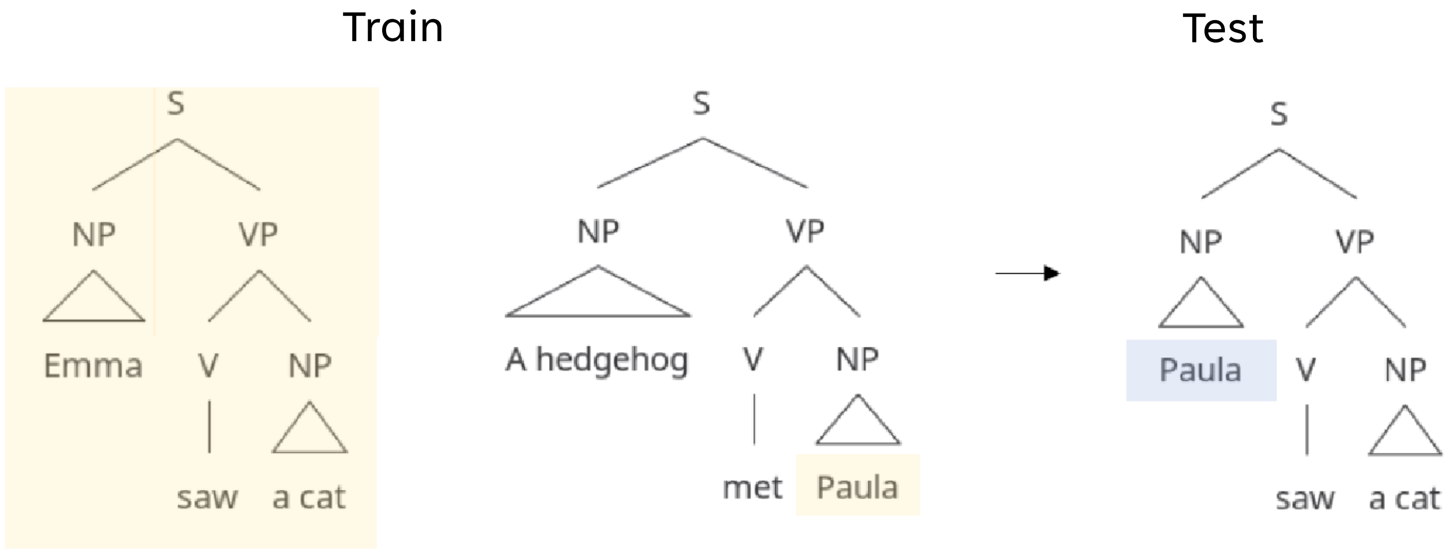
- Kim and Linzen (2020) introduced a dataset called COGS to test the semantic compositional generalization ability of NLP models.
- They evaluated transformers, unidirectional LSTMs, and bidirectional LSTMs.

Model	Dev.	Test	Gen.
Transformer	0.96	0.96	**0.35** (± 0.06)
LSTM (Bi)	0.99	0.99	0.16 (± 0.08)
LSTM (Uni)	0.99	0.99	0.32 (± 0.06)



COMPOSITIONAL GENERALIZATION

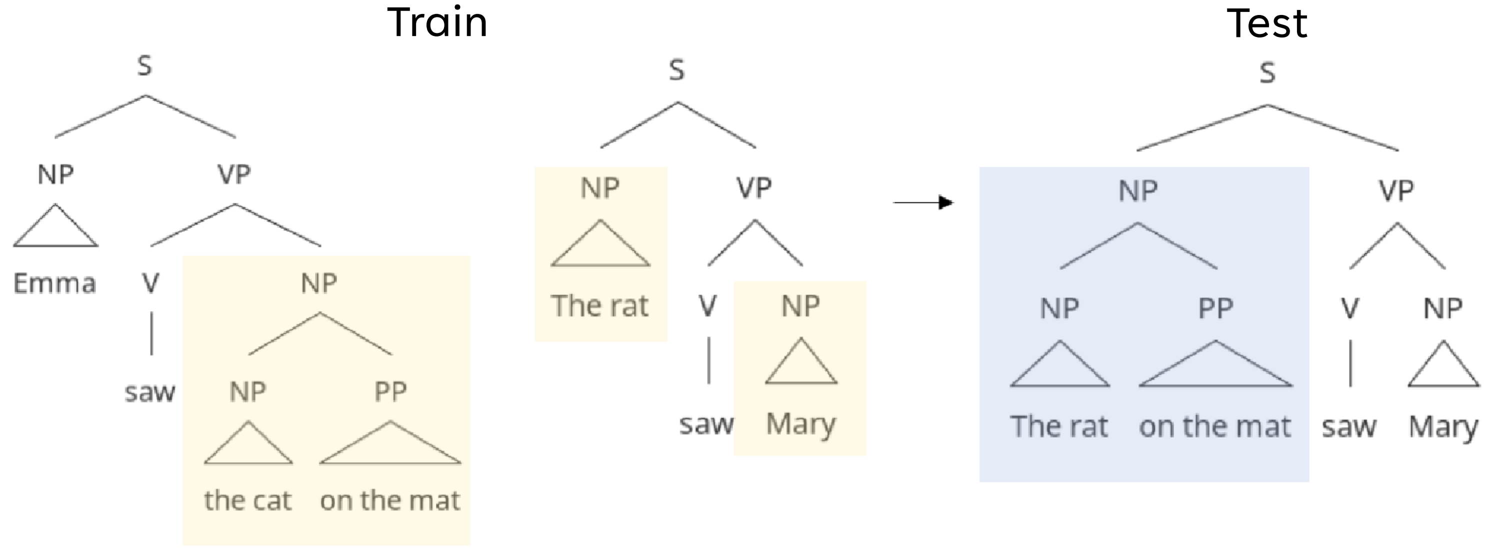
- They considered two types of generalization:
  - Lexical generalization: A word is used in a previously-unseen context.



[Kim and Linzen, 2020]

COMPOSITIONAL GENERALIZATION

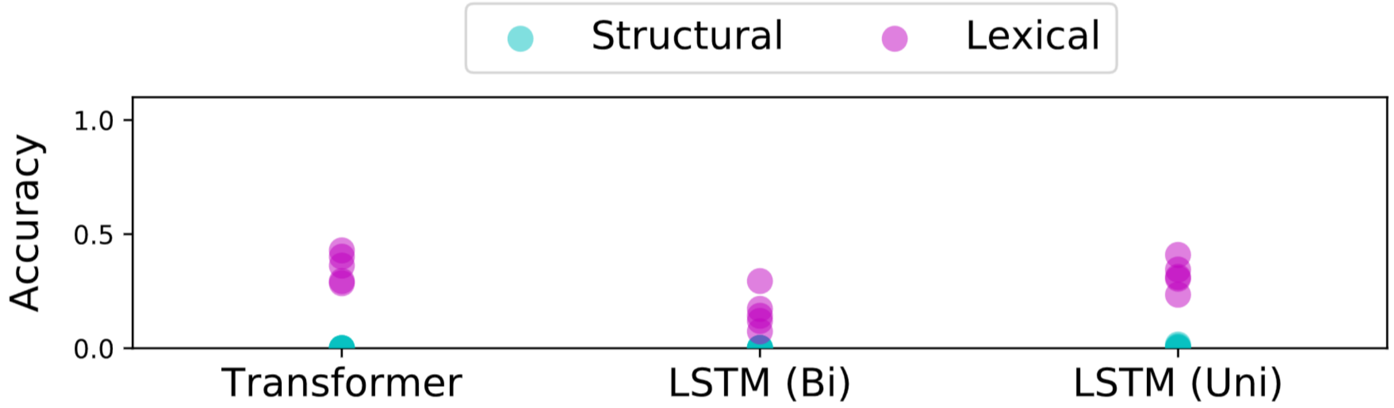
- They considered two types of generalization:
  - Lexical generalization: A word is used in a previously-unseen context.
  - Structural generalization: Familiar structures are combined into a novel larger structure.



[Kim and Linzen, 2020]

COMPOSITIONAL GENERALIZATION

- They considered two types of generalization:
  - Lexical generalization: A word is used in a previously-unseen context.
  - Structural generalization: Familiar structures are combined into a novel larger structure.



- They found that the tested models are incapable of structural generalization.
  - Newer models may do better?

[Kim and Linzen, 2020]

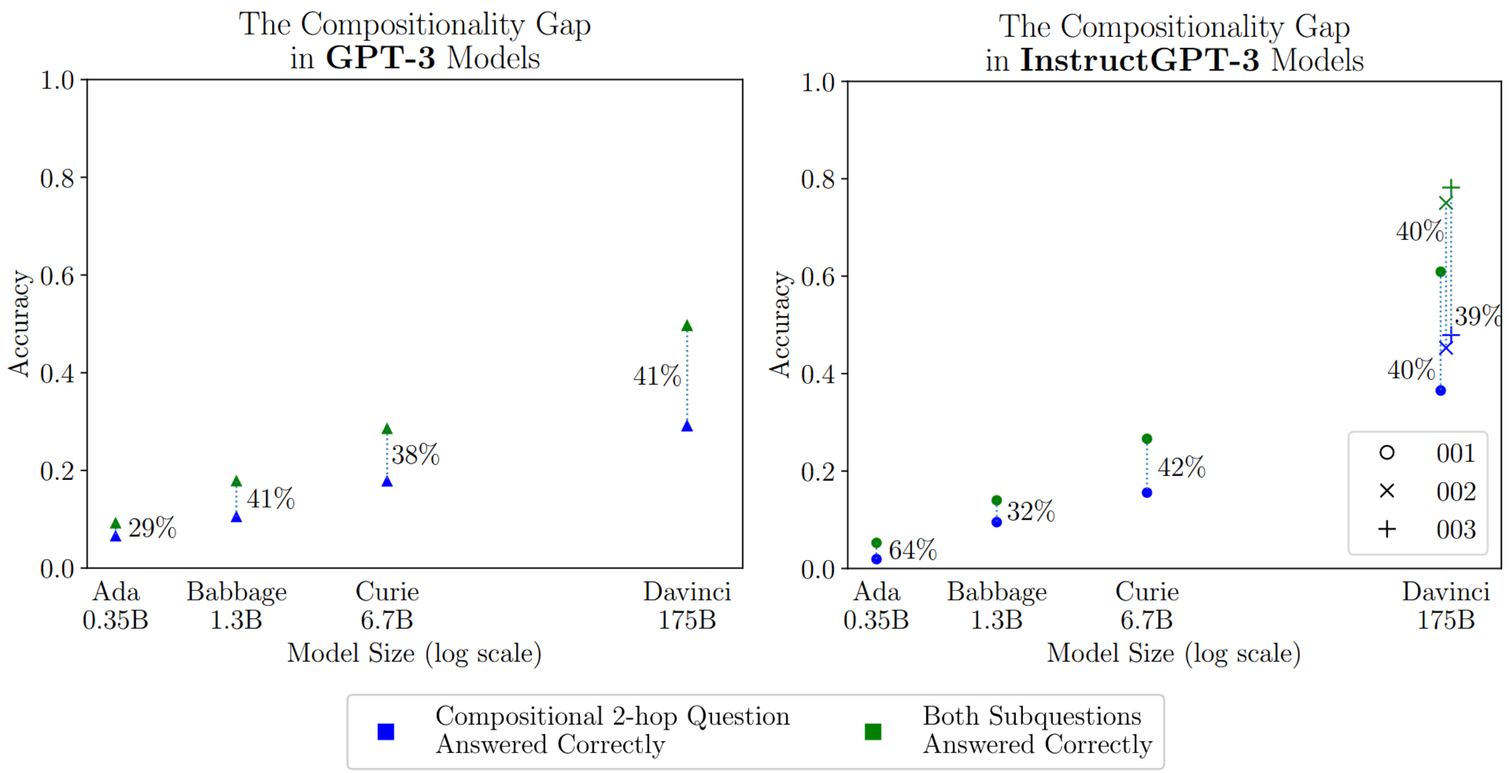
COMPOSITIONAL GENERALIZATION

- Compositionality is also a feature of reasoning.
- A proof can consist of simpler subproofs.
  - In order to find the full proof, an intelligent system must first find the subproofs,
  - Then combine them to produce the full proof.
- E.g., 2-hop question answering:

  ‘`Who won the Master’s Tournament the year Justin Bieber was born?`’
- To answer this, you must prove:
  - The year Justin Bieber was born is 1994.
  - Jose Maria Olazabal won the Master’s Tournament in 1994.

COMPOSITIONAL GENERALIZATION

- Press et al. (2023) measured the difference in accuracy of LM responses to the two individual questions and the compositional question.



LOGICAL FORMALISMS

- What makes a good logical formalism/meaning representation?
  - Coverage
  - Language-independence
  - Uniqueness
  - Amenable for reasoning
  - Easy to parse
  - Compositional
  - Human-readable:
    - Logical forms should be easy to understand by humans.
    - Ideally, they shouldn't require extensive special training (e.g., a university course on formal semantics).

LOGICAL FORMALISMS

- However, there is a tradeoff between coverage and human-readability.
- At one extreme, we have domain-specific logical formalisms.
  - Since the domain is restricted, we don't have to worry about coverage.
  - We can ignore sentences outside the domain.
- As we extend the coverage of the formalism and make it more domain-independent,
  - We have to make sure the formalism can capture the semantic expressivity of natural language.
    - E.g., natural language can refer to actions described as verbs,
    - The correct representation of adverbial modifiers, etc.
- As a result, they become more difficult to read. (perhaps unavoidably)

LMS USE LATENT MEANING REPRESENTATIONS?

- Do LLMs have a latent meaning representation?
- Or do they use natural language?
- Do they "think in English"?
- Wendler, Veselovsky, Monea, and West (2024) inspected the intermediate activations of Llama-2 models.
  - They applied a mechanistic interpretability technique called logit lens.
  - Instead of waiting until the last layer to decode the activations into vocabulary space (via unembedding and softmax),
  - Apply the unembedding+softmax to activations of intermediate layers.

LMS USE LATENT MEANING REPRESENTATIONS?

- They tested three tasks:
  - Translation

```
Français: "vertu" - 中文: "德"
Français: "siège" - 中文: "座"
Français: "neige" - 中文: "雪"
Français: "montagne" - 中文: "山"
Français: "fleur" - 中文: "
```

[Wendler, Veselovsky, Monea, and West, 2024]

LMS USE LATENT MEANING REPRESENTATIONS?

- They tested three tasks:
  - Translation
  - Repetition

```
中文: "德" - 中文: "德"
中文: "座" - 中文: "座"
中文: "雪" - 中文: "雪"
中文: "山" - 中文: "山"
中文: "花" - 中文: "
```

[Wendler, Veselovsky, Monea, and West, 2024]

LMS USE LATENT MEANING REPRESENTATIONS?

- They tested three tasks:
  - Translation
  - Repetition
  - Cloze

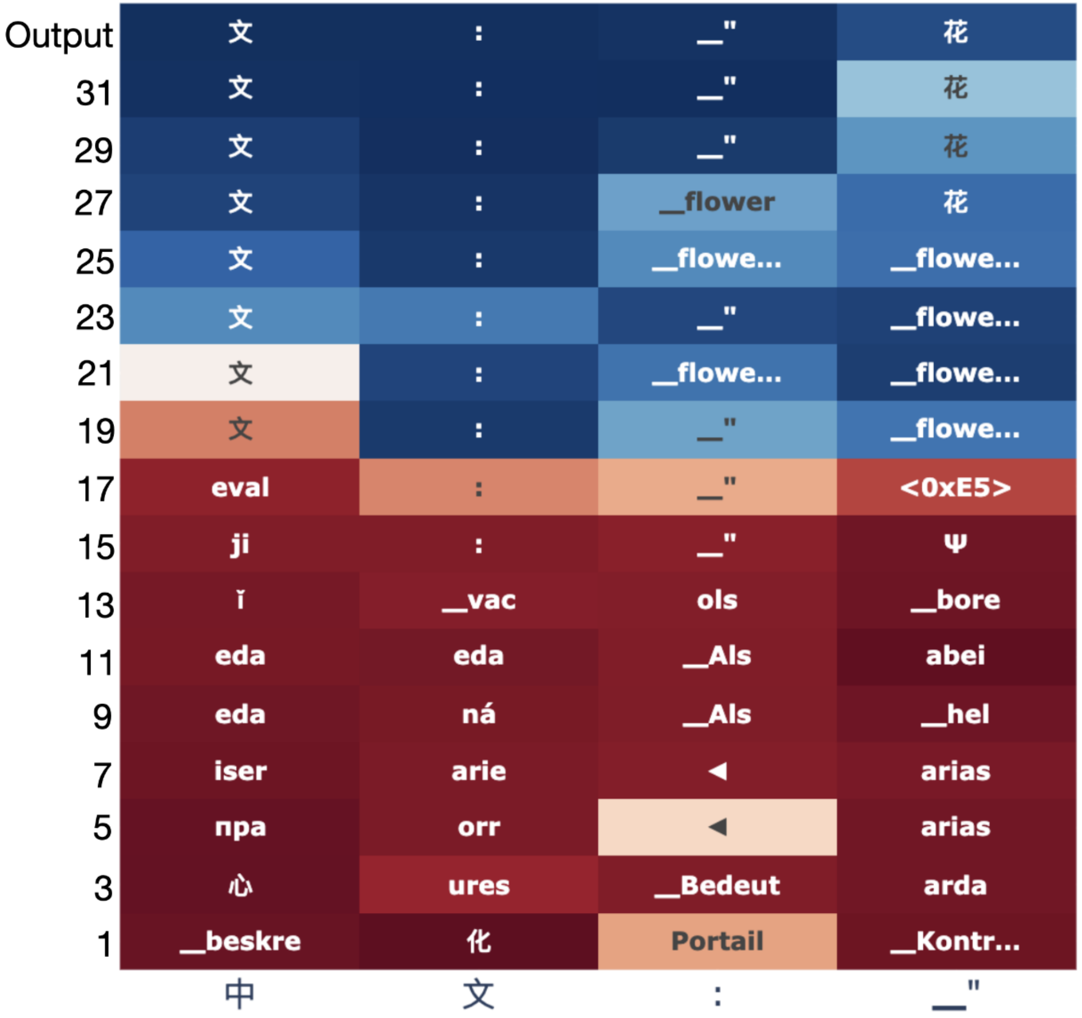
A "___" is used to play sports like soccer and basketball. Answer: "ball".
A "___" is a solid mineral material forming part of the surface of the earth. Answer: "rock".
A "___" is often given as a gift and can be found in gardens. Answer: "

[Wendler, Veselovsky, Monea, and West, 2024]

LMS USE LATENT MEANING REPRESENTATIONS?

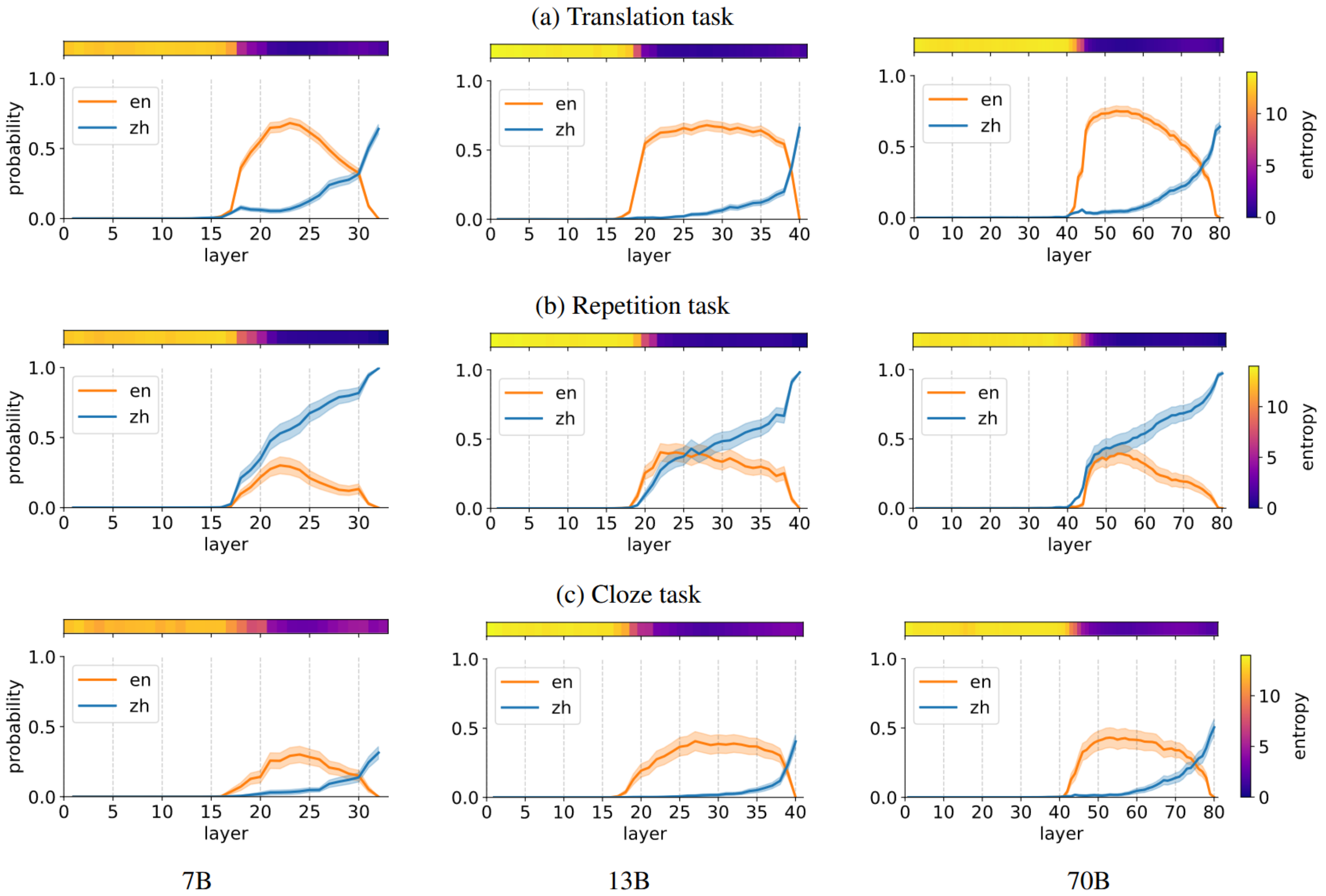
- Example of logit lens on translation:
  - Input: ‘`Français: "fleur" - 中文: `’
- Note that layers 19-27 contain the English word ‘`flower`’,

  Despite the translation being from French into Chinese.



[Wendler, Veselovsky, Monea, and West, 2024]

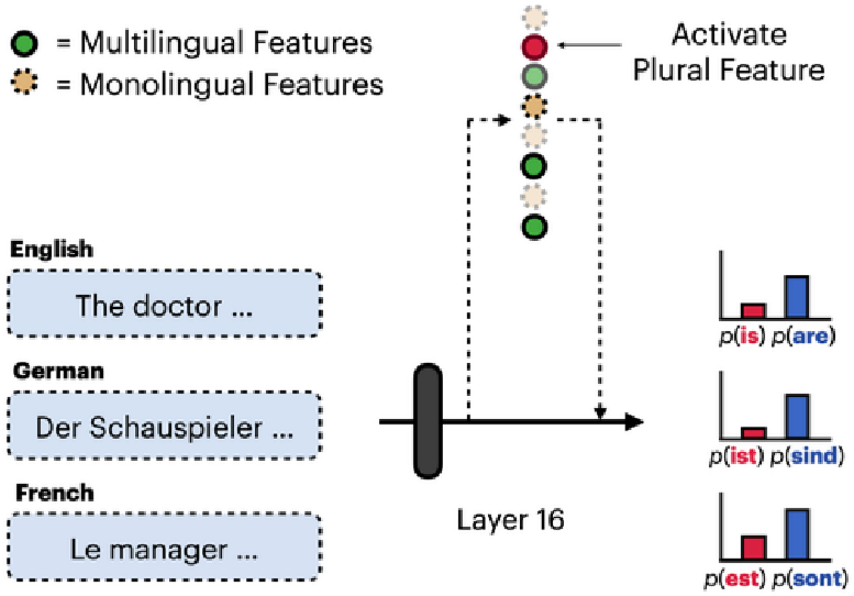
LMS USE LATENT MEANING REPRESENTATIONS?



[Wendler, Veselovsky, Monea, and West, 2024]

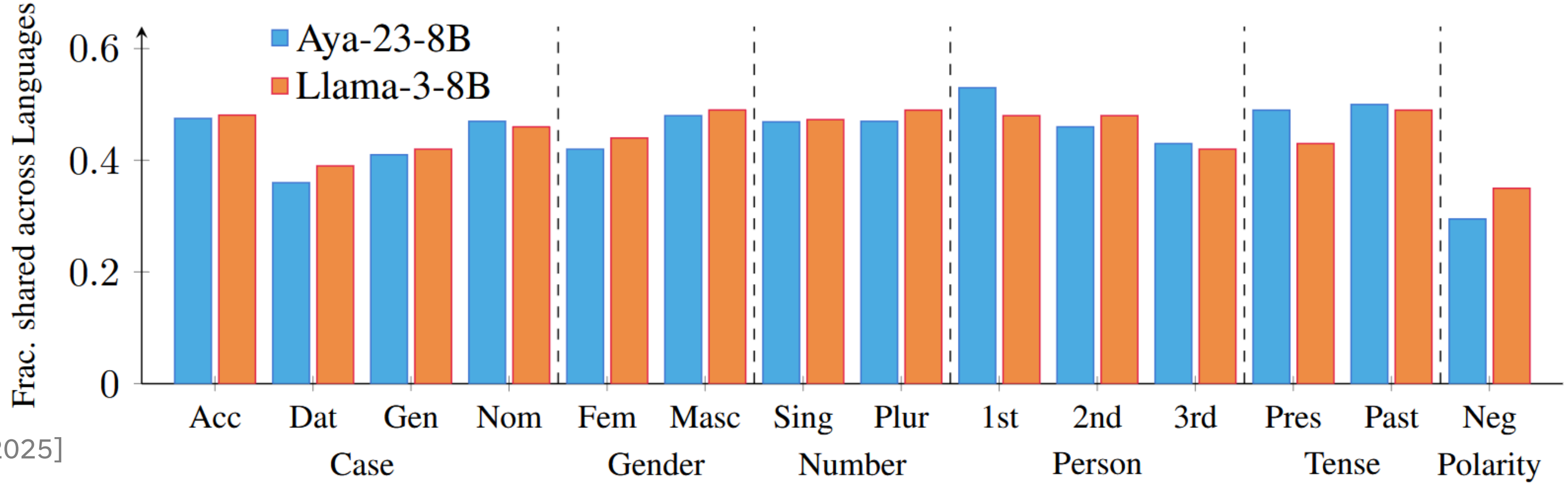
LMS USE LATENT MEANING REPRESENTATIONS?

- But perhaps LLMs do use a meaning representation,
  - Except this representation is “close” to English.
- Brinkmann et al (2025) used sparse autoencoders (another tool for mechanistic interpretability analysis) to identify grammatical features.
  - E.g., plural, singular, 1st person, 2nd person, 3rd person, past tense, etc.
  - They analyzed Llama-3-8B and Aya-23-8B.



LMS USE LATENT MEANING REPRESENTATIONS?

- They tested the LM on inputs from 23 languages.
- For each grammatical aspect (e.g., plural), they found the top 32 features for each language.
- They then computed the number of features in the intersection over all languages, and divided by the number of features in the union.



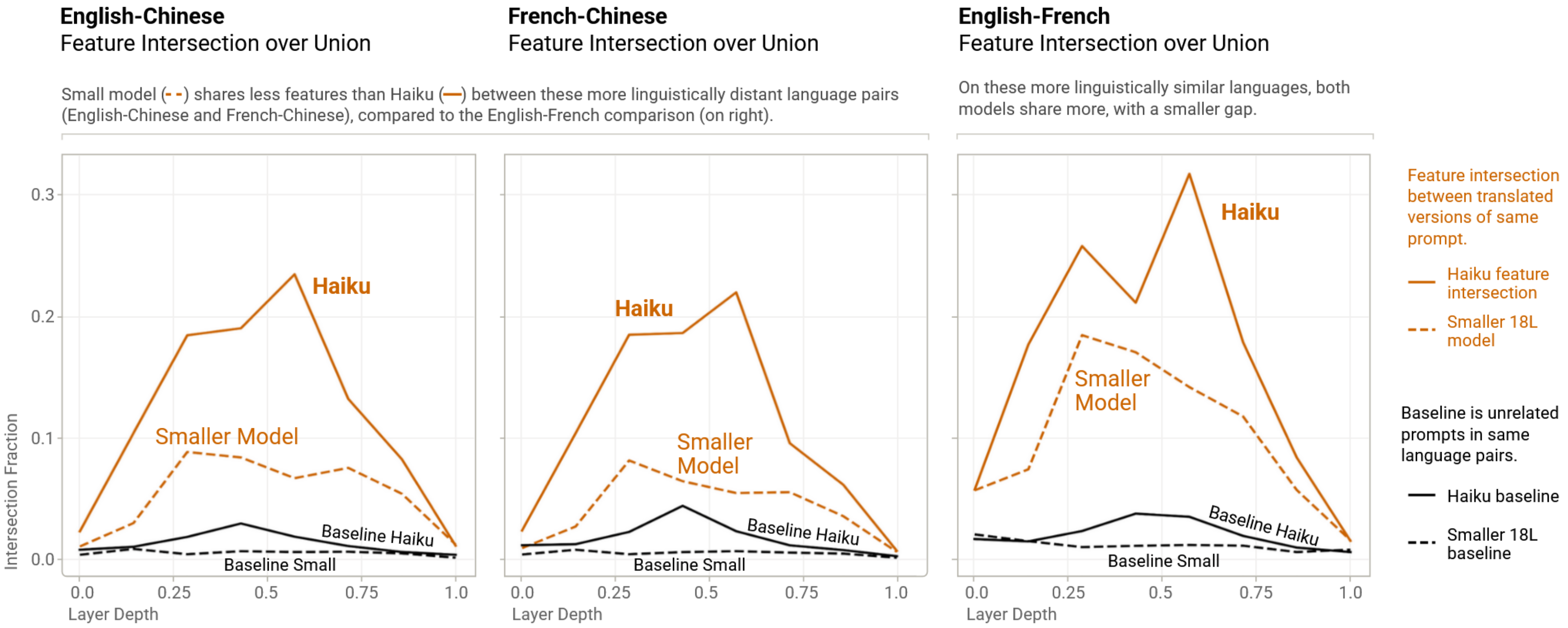
[Brinkmann et al., 2025]

LMS USE LATENT MEANING REPRESENTATIONS?

- Lindsey et al. (2025) performed a similar analysis on Claude 3.5 Haiku.
- They show the LM a paragraph in one language and the translated paragraph in another language.
- They record which features activate.
- They compute the number of features that were active during both paragraphs, and divide by the total number of active features.
- They compare this to a baseline where the LM is shown two unrelated paragraphs in the same language pair.

LMS USE LATENT MEANING REPRESENTATIONS?

- Lindsey et al. (2025) performed a similar analysis on Claude 3.5 Haiku.



[Lindsey et al., 2025]

SEMANTICS

- We have discussed meaning representations/logical formalisms:
  - Second-order logic
  - Higher-order logic
  - Event semantics
- What makes a logical formalism good?
  - Compositionality
- Next time: How to convert natural language into logical forms?
  - Semantic parsing

QUESTIONS?