CS 577: NATURAL LANGUAGE PROCESSING

Abulhair Saparov

Lecture 26: Semantics III

FINAL EXAM LOGISTICS

- Thursday, December 18th (next week)
  - 10:30am–12:30pm
  - LILY G126
- Bring your ID for verification
- You may bring one 8.5” × 11” cheat sheet
  - Practice questions are available on Ed and course website

SEMANTICS

- We have discussed how to represent the meaning of natural language.
- Example meaning representations
  - Logic
- What makes a logical formalism good?
  - Compositionality
  - Coverage
  - Amenable to reasoning
  - etc.
- How do we convert natural language into logical forms?
  - Semantic parsing

DO WE NEED A LOGICAL FORM?

- It's not obvious that human language processing involves converting natural language into logical form.
- Counterargument: Logical forms enable reasoning.
- But why not do reasoning in natural language?
  - I.e., natural language is the logical formalism.
- One potential roadblock: Ambiguity.
- Logical forms in a formal language are unambiguous.
  - Natural language is infamously ambiguous.

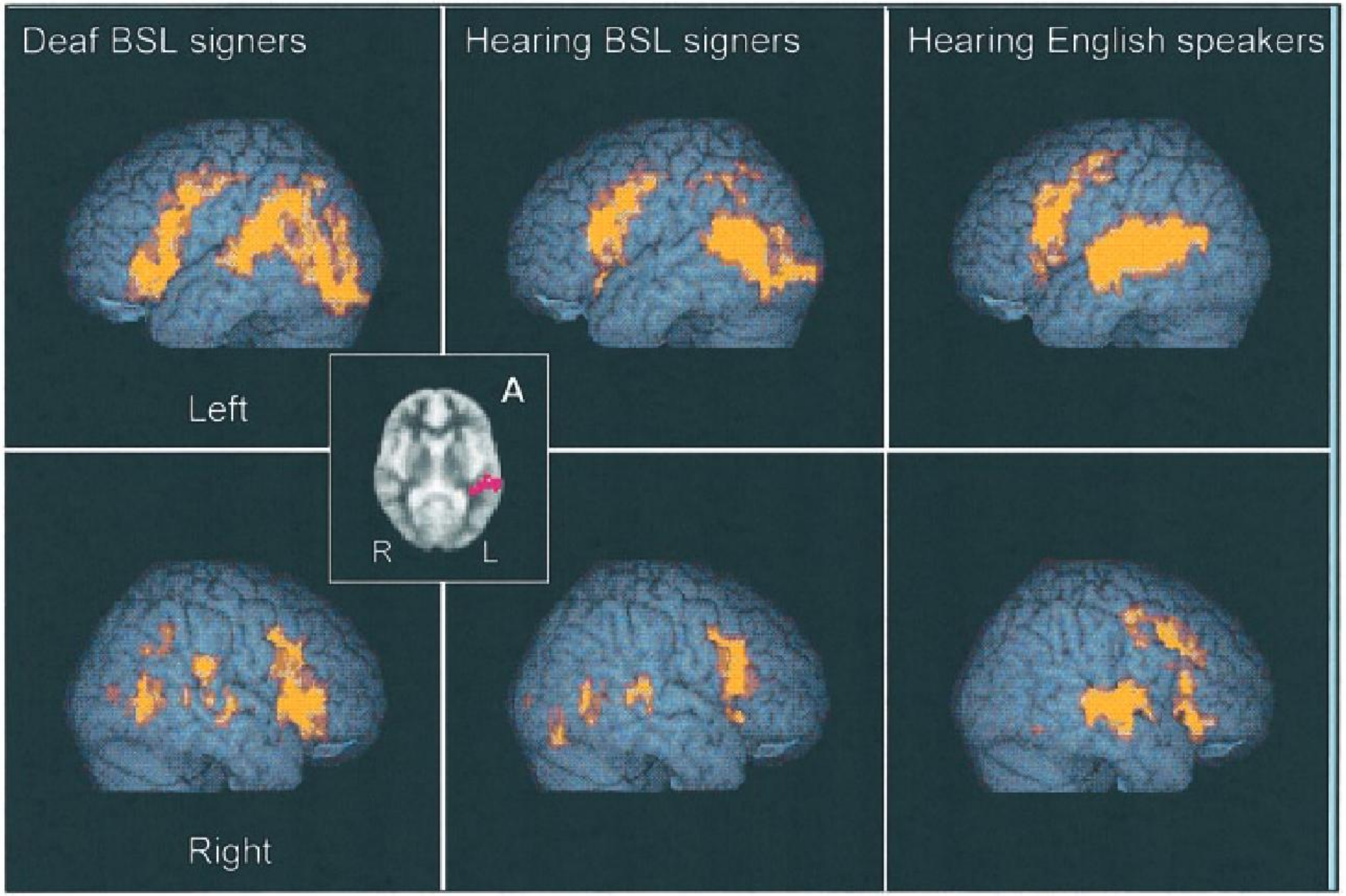
DO WE NEED A LOGICAL FORM?

- Consider the example:
  - ‘`All dogs chase a cat.`’
    - `∀d(dog(d) -> ∃c(cat(c) & chase(d,c)))`
    - `∃c(cat(c) & ∀d(dog(d) -> chase(d,c)))`
  - ‘`Sif and Fen are dogs.`’
    - `dog(sif) & dog(fen)`
  - ‘`Sif only chases Felix.`’
    - `chase(sif,felix) & ¬∃x(x≠felix & chase(sif,x))`
- If we take the second reading of ‘`All dogs chase a cat`’, we can prove that ‘`Fen chases Felix.`’
- If we take the first reading, the proof is no longer valid.

DO HUMANS USE LOGICAL FORMS?

- We discussed whether LLMs “use logical forms”, but what about humans?
- There is some neuroscientific evidence that humans perform reasoning in a more abstract, modality-independent fashion.
- MacSweeney (2002) performed brain scans of 11 volunteers while they performed a reading comprehension task.
  - 4 deaf subjects who know British Sign Language (BSL).
  - 4 hearing subjects who know BSL.
  - 3 hearing subjects who don’t know BSL.
  - Scanned subjects using fMRI (functional magnetic resonance imaging).

DO HUMANS USE LOGICAL FORMS?



DO HUMANS USE LOGICAL FORMS?

- There are brain regions that are active across both deaf and hearing subjects.
- But maybe this is due to common syntactic processing across modalities?
  - Not likely since BSL and spoken English are very different grammatically.
  - BSL has OSV word order and nouns are head-initial (e.g., ‘`car blue`’).

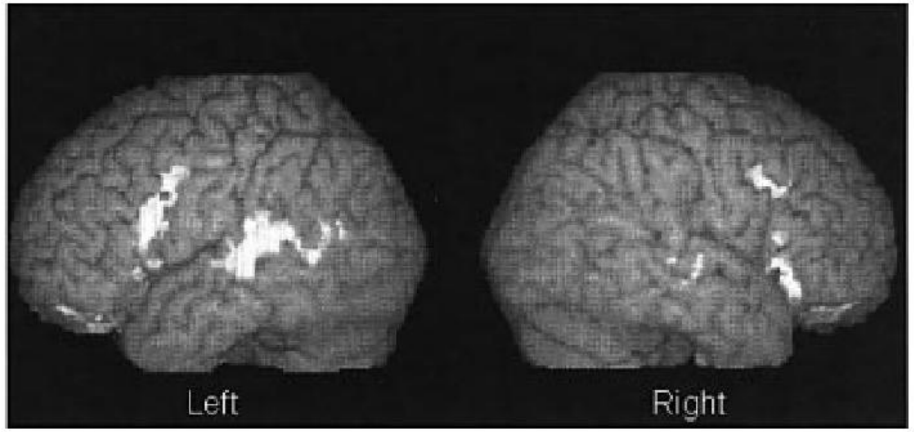
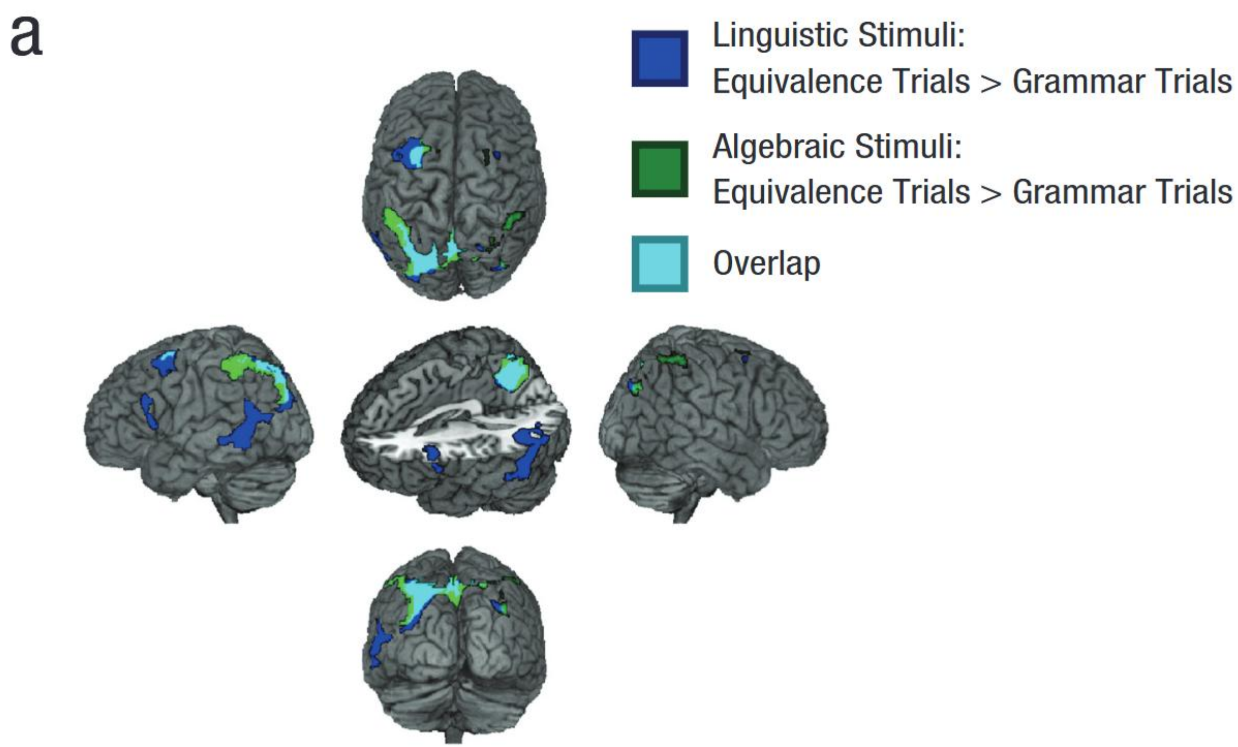


Fig. 2 Locations of common activation for audio-visual English (hearing) and BSL sentences (deaf). Activation up to 5 mm under the surface of the cortex is displayed.

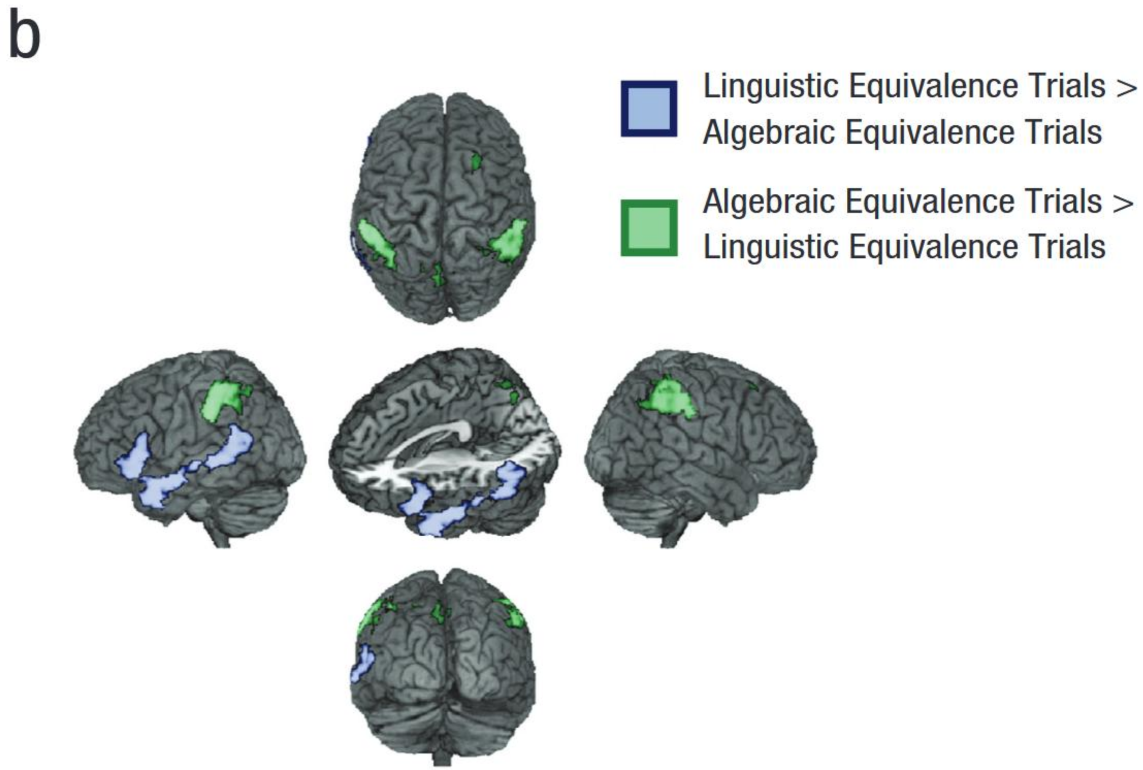
DO HUMANS USE LOGICAL FORMS?

- Monti et al. (2012) used fMRI to localize which brain areas were active when subjects are given a language task vs a mathematical reasoning task.



DO HUMANS USE LOGICAL FORMS?

- Monti et al. (2012) used fMRI to localize which brain areas were active when subjects are given a language task vs a mathematical reasoning task.



DO WE NEED LOGICAL FORMS?

- Monti et al. (2012) used fMRI to localize which brain areas were active when subjects are given a language task vs a mathematical reasoning task.
- Neuroscientific evidence supports the notion of a “language network” within the brain that is highly specialized for language processing.
- But this language network is not heavily involved in high-level reasoning.
  - E.g., mathematical reasoning.
- Logical forms are useful for other applications in NLP.
  - E.g., code generation
    - Text-to-SQL
    - etc…
  - Programs are logical forms!

SEMANTIC PARSING

- Semantic parsing is the task of converting natural language to logical form.
  - NLP models that are trained to convert natural language into logical form (e.g., code) are effectively performing semantic parsing.
- Consider the following example:
  - We want to parse '`Sif chases Felix`' into `chase(sif,felix)`
  - We can model the syntax of the natural language with a grammar.
  - Logic (and any formal language) can easily be described with a CFG.

SEMANTIC PARSING

- A simple CFG for English:

```
S -> N VP            N -> 'Sif'
VP -> V N            N -> 'Felix'
V -> 'chases'
```

- CFG for first-order logic:

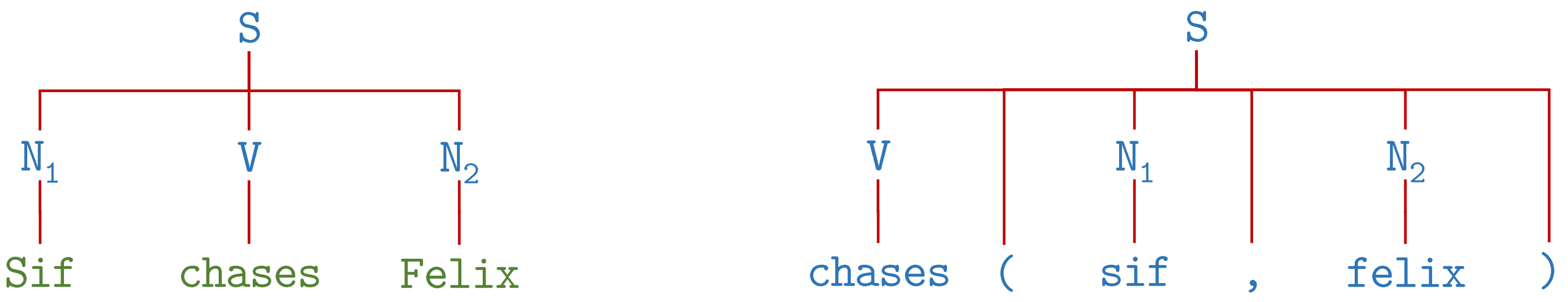
```
S -> S '&' S         S -> P '(' T ')'
S -> S '|' S         S -> P '(' T ',' T ')'
S -> S '=>' S        P -> 'chases'
S -> '(' S ')'       T -> V
S -> '¬' S           V -> 'x'
S -> '∀' V S         T -> 'sif'
S -> '∃' V S         T -> 'felix'
```

SYNCHRONOUS GRAMMARS

- Combine these grammars to model both English and FOL simultaneously?

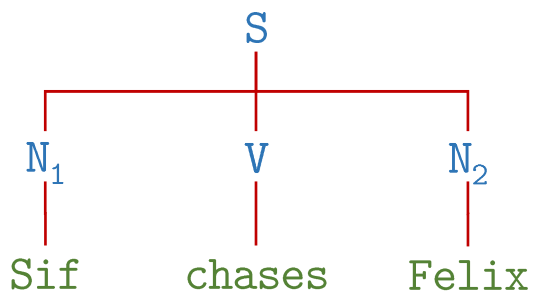
```
S -> <N1 V N2, V '(' N1 ',' N2 ')'>
N -> <'Sif', 'sif'>
N -> <'Felix', 'felix'>
V -> <'chases', 'chases'>
```

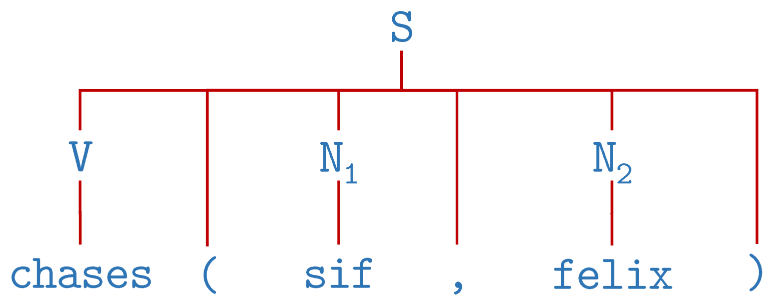
- This is a synchronous context free grammar (SCFG).
- We derive/parse the sentence and logical form simultaneously:



SYNCHRONOUS GRAMMARS

- But this grammar looks "flatter" than our earlier English grammar.
  - Notice the `VP -> V N` rule was "flattened" into the `S -> N VP` rule.
  - But it is impossible to avoid this in SCFG.
  - Consider the rule for the logical form: `S -> V '(' `N_1` ',' `N_2` ')'`
  - The `VP` is split into `V` and N_2, which are separated by many symbols.
- But it is possible to write synchronous grammars where `VP` is preserved using richer grammar formalisms (e.g., STAG).





PARSING WITH SYNCHRONOUS GRAMMARS

- In semantic parsing, however, we only have ‘`Sif chases Felix`’.
  - How do we obtain the logical form?
- Consider the grammar:

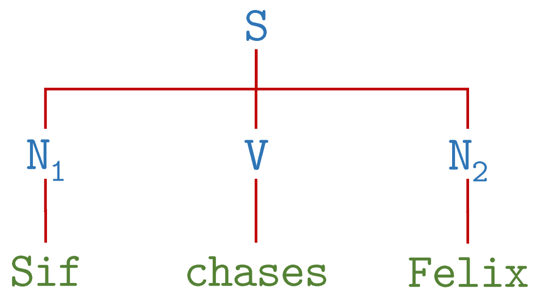
```
S -> <N1 V N2, V ‘(’ N1 ‘,’ N2 ‘)’>
N -> <‘Sif’, ‘sif’>
N -> <‘Felix’, ‘felix’>
V -> <‘chases’, ‘chases’>
```

- Focus on just the natural language part of the grammar:

```
S -> N1 V N2
N -> ‘Sif’
N -> ‘Felix’
V -> ‘chases’
```

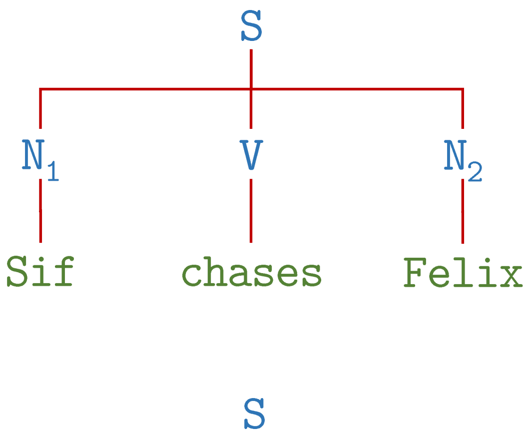
PARSING WITH SYNCHRONOUS GRAMMARS

- Use any CFG parsing method to parse ‘`Sif chases Felix`’.
  - E.g., Earley parsing



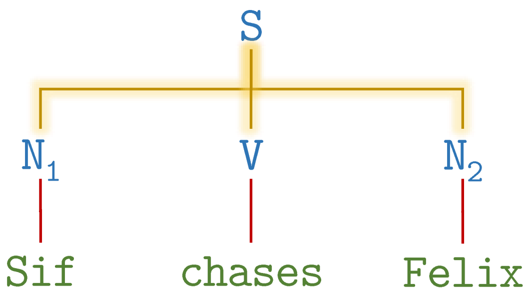
- Then we can reconstruct the derivation tree for the logical form by inspecting each rule in the above tree.
  - For each rule, we look at the right-hand side to determine how to construct the logical form.

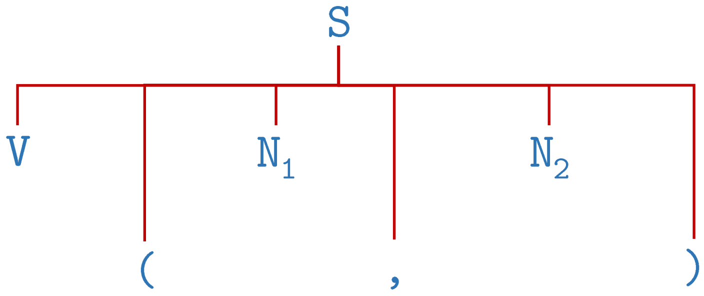
PARSING WITH SYNCHRONOUS GRAMMARS




```
S -> <N1 V N2, V '(' N1 ',' N2 ')'>
N -> <'Sif', 'sif'>
N -> <'Felix', 'felix'>
V -> <'chases', 'chases'>
```

PARSING WITH SYNCHRONOUS GRAMMARS





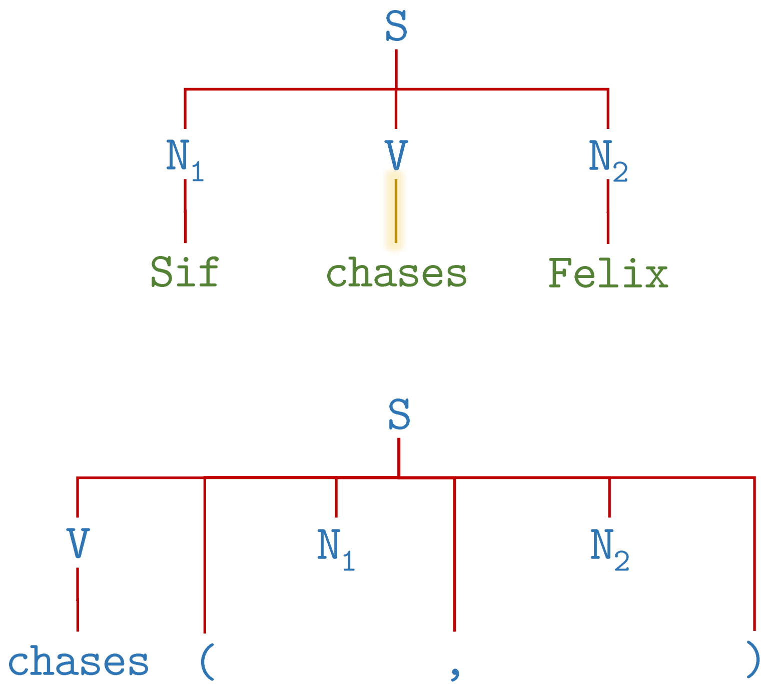
S -> <N_1 V N_2, V ‘(’ N_1 ‘,’ N_2 ‘)’>

N -> <‘Sif’, ‘sif’>

N -> <‘Felix’, ‘felix’>

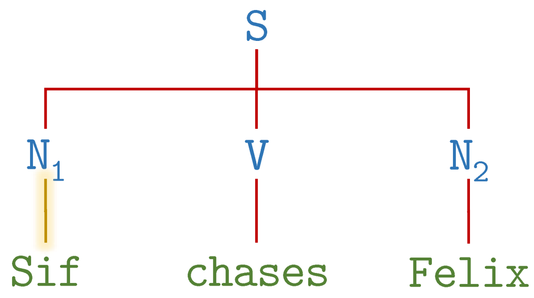
V -> <‘chases’, ‘chases’>

PARSING WITH SYNCHRONOUS GRAMMARS




```
S -> <N1 V N2, V ‘(’ N1 ‘,’ N2 ‘)’>
N -> <‘Sif’, ‘sif’>
N -> <‘Felix’, ‘felix’>
V -> <‘chases’, ‘chases’>
```

PARSING WITH SYNCHRONOUS GRAMMARS

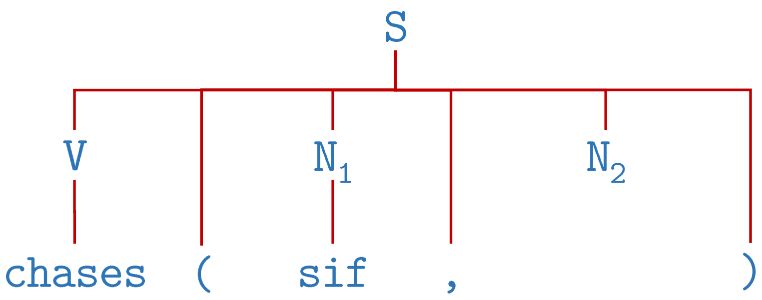


S -> <N_1 V N_2, V '(' N_1 ',' N_2 ')'>

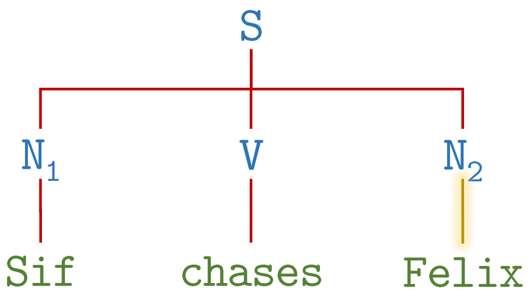
N -> <'Sif', 'sif'>

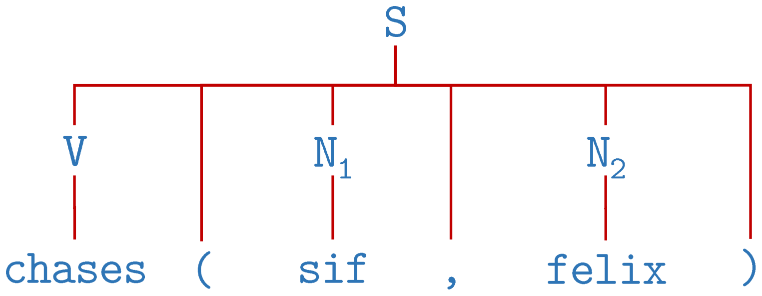
N -> <'Felix', 'felix'>

V -> <'chases', 'chases'>



PARSING WITH SYNCHRONOUS GRAMMARS





S -> <N_1 V N_2, V ‘(’ N_1 ‘,’ N_2 ‘)’>
N -> <‘Sif’, ‘sif’>
N -> <‘Felix’, ‘felix’>
V -> <‘chases’, ‘chases’>

PARSING WITH SYNCHRONOUS GRAMMARS

- Note that, when we reconstruct the logical form, there may be more than one matching rule.
- Consider the slightly modified grammar:

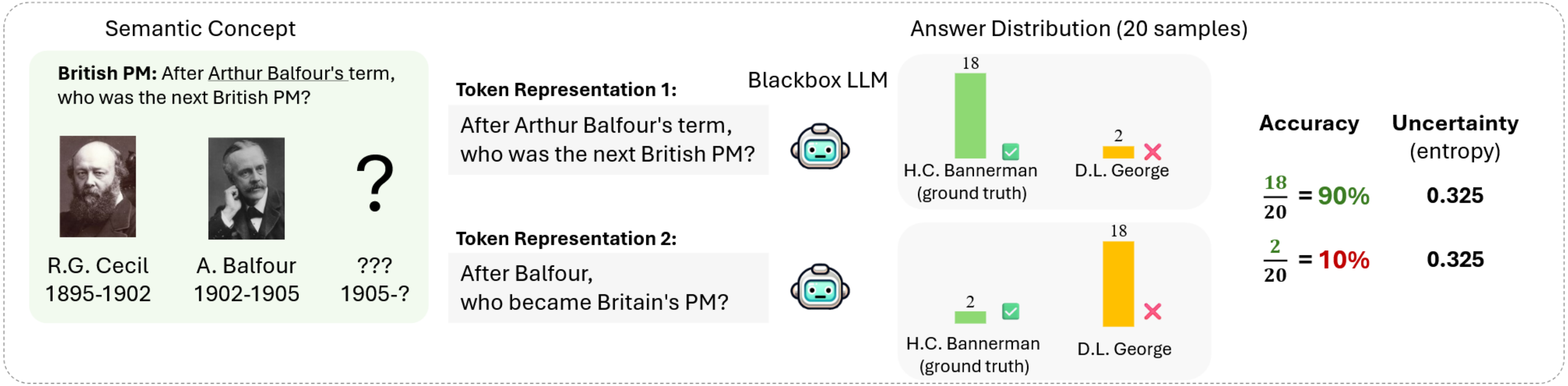
```
S -> <N₁ V N₂, V '(' N₁ ',' N₂ ')'>
N -> <'Sif', 'sif'>
N -> <'Felix', 'felix_lee'>
N -> <'Felix', 'felix_mendelssohn'>
V -> <'chases', 'chases'>
```

- When we parse `'Sif chases Felix'` and inspect the rule `N -> 'Felix'`, there are two matching rules.
  - We can choose either rule to produce a valid logical form.
- There are two valid logical forms:
  - `chases(sif,felix_lee)` and `chases(sif,felix_mendelssohn)`
  - Example of semantic ambiguity.
- SCFG can capture both syntactic and semantic ambiguity.

CAN LLMS DO SEMANTIC PARSING?

- If LLMs have some kind of internal meaning representation, they need a way to convert natural language into this representation.
- How do we test for this ability?
- Idea: End-to-end test.
  - Give the model a natural language reasoning task and measure its performance.
  - But this could confound semantic parsing ability with reasoning ability.
- Idea: Rephrase the natural language input such that the meaning does not change, then evaluate model performance.
  - Prompt sensitivity suggests LLMs are not mapping semantically-equivalent inputs into the same "logical form".

CAN LLMS DO SEMANTIC PARSING?



- Idea: Rephrase the natural language input such that the meaning does not change, then evaluate model performance.
  - Prompt sensitivity suggests LLMs are not mapping semantically-equivalent inputs into the same “logical form”.

[Cox et al., 2025]

CAN LLMS DO SEMANTIC PARSING?

- Another idea: Use the LLM to perform semantic parsing directly.
- Liu et al. (2023) measured the zero-shot parsing performance of ChatGPT on the text-to-SQL task.
  - They compared against supervised baselines which were trained specifically for this task.
  - The measured three metrics:
    - Validity: Does the predicted SQL have valid syntax?
    - Execution accuracy: Does the predicted SQL produce the same result as the ground truth SQL?
    - Test-suite accuracy: Similar to execution accuracy, but tested over many databases (a test suite).

CAN LLMS DO SEMANTIC PARSING?

- Another idea: Use the LLM to perform semantic parsing directly.
- Liu et al. (2023) measured the zero-shot parsing performance of ChatGPT on the text-to-SQL task.

Methods / Datasets	SPIDER			SPIDER-SYN			SPIDER-REALISTIC		
	VA	**EX**	**TS**	**VA**	**EX**	**TS**	**VA**	**EX**	**TS**
T5-3B + PICARD	98.4	79.3	69.4	98.2	69.8	61.8	97.1	71.4	61.7
RASAT + PICARD	98.8	80.5	70.3	98.3	70.7	62.4	97.4	71.9	62.6
RESDSQL-3B + NatSQL	99.1	84.1	73.5	98.8	76.9	66.8	98.4	81.9	70.1
ChatGPT	97.7	70.1(14↓)	60.1	96.2	58.6(18.3↓)	48.5	96.8	63.4(18.5 ↓)	49.2

CAN LLMS DO SEMANTIC PARSING?

- Another idea: Use the LLM to perform semantic parsing directly.
- Liu et al. (2023) measured the zero-shot parsing performance of ChatGPT on the text-to-SQL task.

Methods / Datasets	SPIDER-DK			ADVETA(RPL)			ADVETA(ADD)		
	VA	**EX**	**TS**	**VA**	**EX**	**TS**	**VA**	**EX**	**TS**
T5-3B + PICARD	97.8	62.5	-	92.7	50.6	-	97.2	69.4	-
RASAT + PICARD	98.5	63.9	-	92.9	51.5	-	97.4	70.7	-
RESDSQL-3B + NatSQL	98.8	66.0	-	93.9	54.4	-	97.9	71.9	-
ChatGPT	96.4	62.6(3.4 ↓)	-	91.4	58.5(**4.1** ↑)	-	93.1	68.1(3.8 ↓)	-

- Few-shot prompting may help further.

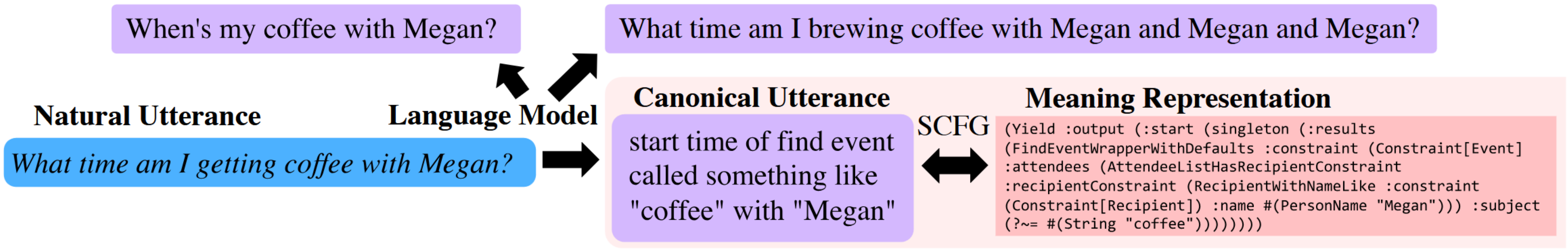
(since validity is already very high, constrained decoding may not be as helpful)

CAN LLMS DO SEMANTIC PARSING?

- Another idea: Suppose we have a synchronous grammar of English and SQL.
  (or more generally, a natural language and a formal language)
- But it is difficult to write a full grammar of English.
- What if, instead, we wrote a synchronous grammar for a simplified subset of English and the formal language.
  - This subset is called “canonical form.”
  - And we use an LLM to translate from general English into canonical form.

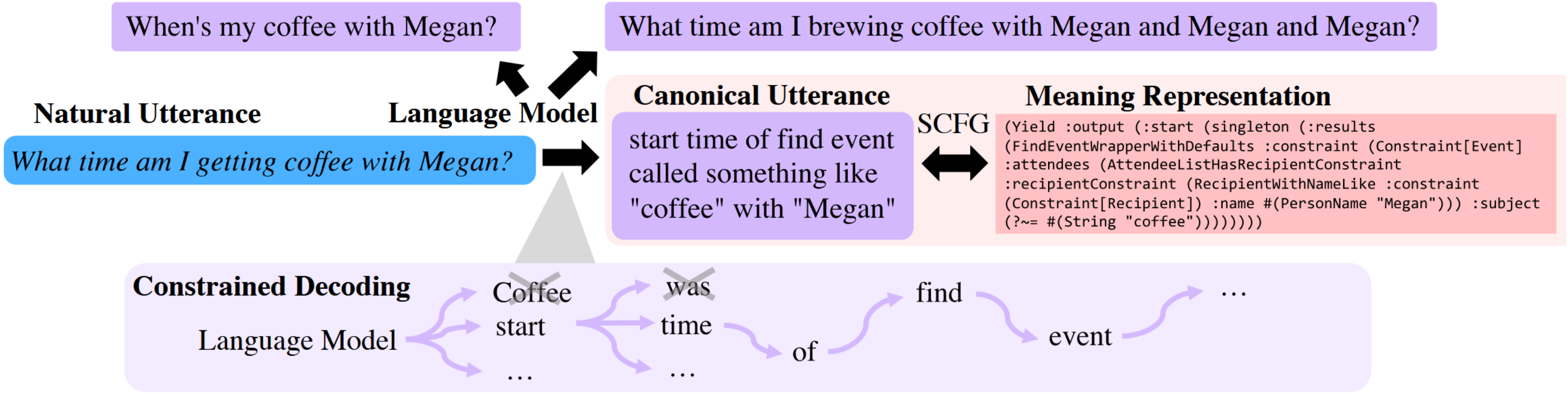
CAN LLMS DO SEMANTIC PARSING?

- This was the idea proposed by Shin et al (2021).



CAN LLMS DO SEMANTIC PARSING?

- This was the idea proposed by Shin et al (2021).
- Since we have an SCFG of the canonical form, we can use constrained decoding to ensure the LLM outputs the correct form.



CAN LLMS DO SEMANTIC PARSING?

- They used few-shot prompting to convert the natural sentence into canonical form.

```
Let's translate what a human user says into
    what a computer might say.

   Human: when is the weekly standup
Computer: start time of weekly standup
   Human: what date is the weekly standup
Computer: date of weekly standup
              ...
   Human: how long is the weekly standup
Computer:
```

[Shin et al., 2021]

CAN LLMS DO SEMANTIC PARSING?

- Once we have the canonical form, we can use the SCFG to semantically parse it into logical form.
- They experimented with a number of semantic parsing datasets.
  - One such dataset is called Overnight, which uses a Lisp-like LF.

which january 2nd meetings is alice attenting [sic]

meeting whose date is jan 2 and whose attendee is alice

```
(call listValue (call filter
   (call filter (call getProperty
         (call singleton en.meeting) (string !type))
      (string date) (string =) (date 2015 1 2))
   (string attendee) (string =) en.person.alice))
```

[Shin et al., 2021]

CAN LLMS DO SEMANTIC PARSING?

- Once we have the canonical form, we can use the SCFG to semantically parse it into logical form.
- They experimented with a number of semantic parsing datasets.
  - Another dataset is Break, which uses QDMR (Question Decomposition Meaning Representation).

What color are a majority of the objects?

(colors of (objects)) where (number of (objects for each (colors of (objects))) is highest)

1. objects
2. colors of #1
3. number of #1 for each #2
4. #2 where #3 is highest

[Shin et al., 2021]

CAN LLMS DO SEMANTIC PARSING?

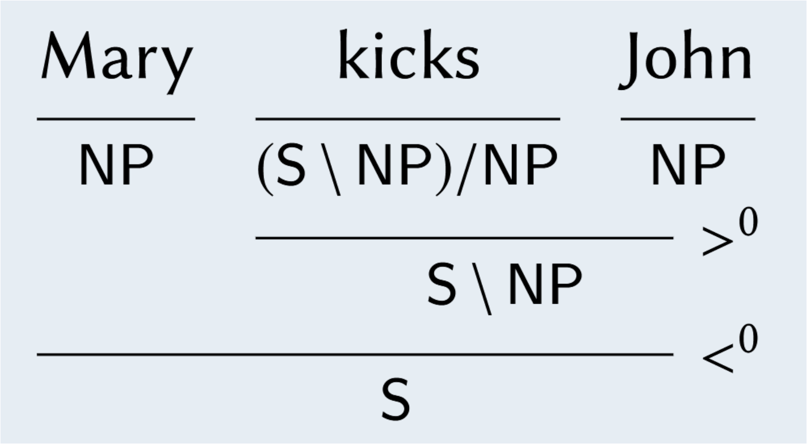
- Results on the Break dataset:

Model	Train n	nem
Wolfson et al.	44,321	0.42
Coleman & Reneau	44,321	0.42
GPT-3 Constrained Canonical	1,000	0.32*
GPT-3 Constrained Canonical	100	0.24*
GPT-3 Constrained Canonical	25	0.20*
GPT-3 Constrained Canonical	200	0.31*
GPT-3 Constrained Meaning	200	0.24*
GPT-3 Unconstrained Canonical	200	0.20*
GPT-3 Unconstrained Meaning	200	0.17*
GPT-3 Constrained Canonical	200	0.24
BART^f Constrained Canonical	200	0.22
BART^f Constrained Meaning	200	0.22
BART^f Unconstrained Canonical	200	0.18
BART^f Unconstrained Meaning	200	0.19

[Shin et al., 2021]

COMBINATORY CATEGORIAL GRAMMAR

- CCG can also be used for semantic parsing.
- Recall that CCG is a mildly-context sensitive grammar formalism.
- We discussed how it can be used to model syntax.
  - E.g., for the sentence '`Mary kicks John`':



[Evang, 2018, Introduction to Semantic Parsing with CCG]

CCG SEMANTIC PARSING

- To use CCG for semantic parsing, add a LF term to each lexicon item.
  - E.g., '`Mary kicks John`':

$$\dfrac{\text{Mary}}{\mathrm{NP} : \mathit{mary}} \quad \dfrac{\text{kicks}}{(\mathrm{S} \backslash \mathrm{NP})/\mathrm{NP} : \lambda x.\lambda y.\mathit{kicks}(y, x)} \quad \dfrac{\text{John}}{\mathrm{NP} : \mathit{john}}$$

[Evang, 2018, Introduction to Semantic Parsing with CCG]

CCG SEMANTIC PARSING

- To use CCG for semantic parsing, add a LF term to each lexicon item.
  - E.g., ‘`Mary kicks John`’:

$$\dfrac{\text{Mary}}{\mathsf{NP} : \mathit{mary}} \quad \dfrac{\dfrac{\text{kicks}}{(\mathsf{S} \backslash \mathsf{NP})/\mathsf{NP} : \lambda x.\lambda y.\mathit{kicks}(y, x)} \quad \dfrac{\text{John}}{\mathsf{NP} : \mathit{john}}}{\mathsf{S} \backslash \mathsf{NP} : \lambda y.\mathit{kicks}(y, \mathit{john})} >^0$$

- In the forward and backward application rules, we combine the respective logical forms using function application.
  - I.e., we apply the function λ`x.`λ`y.kicks(y,x)` to the argument `john`.

[Evang, 2018, Introduction to Semantic Parsing with CCG]

CCG SEMANTIC PARSING

- To use CCG for semantic parsing, add a LF term to each lexicon item.
  - E.g., '`Mary kicks John`':

$$
\begin{array}{ccc}
\text{Mary} & \text{kicks} & \text{John} \\
\hline
\mathsf{NP} : \mathit{mary} & (\mathsf{S} \backslash \mathsf{NP})/\mathsf{NP} : \lambda x.\lambda y.\mathit{kicks}(y, x) & \mathsf{NP} : \mathit{john} \\
\cline{2-3}
 & \multicolumn{2}{c}{\mathsf{S} \backslash \mathsf{NP} : \lambda y.\mathit{kicks}(y, \mathit{john}) \quad >^0} \\
\hline
\multicolumn{3}{c}{\mathsf{S} : \mathit{kicks}(\mathit{mary}, \mathit{john}) \quad <^0}
\end{array}
$$

[Evang, 2018, Introduction to Semantic Parsing with CCG]

CCG SEMANTIC PARSING

- E.g., ‘`Mary sings and dances`’:

$$
\begin{array}{cccc}
\text{Mary} & \text{sings} & \text{and} & \text{dances} \\
\hline
\mathsf{NP} & \mathsf{S}\backslash\mathsf{NP} & ((\mathsf{S}\backslash\mathsf{NP})\backslash(\mathsf{S}\backslash\mathsf{NP}))/(\mathsf{S}\backslash\mathsf{NP}) & \mathsf{S}\backslash\mathsf{NP} \\
\mathit{mary} & \lambda x.\mathit{sings}(x) & \lambda f.\lambda g.\lambda x.g(x) \land f(x) & \lambda x.\mathit{dances}(x)
\end{array}
$$

$$\frac{((\mathsf{S}\backslash\mathsf{NP})\backslash(\mathsf{S}\backslash\mathsf{NP}))/(\mathsf{S}\backslash\mathsf{NP}) \;\; \mathsf{S}\backslash\mathsf{NP}}{(\mathsf{S}\backslash\mathsf{NP})\backslash(\mathsf{S}\backslash\mathsf{NP}) \quad \lambda g.\lambda x.g(x) \land \mathit{dances}(x)} >^0$$

$$\frac{\mathsf{S}\backslash\mathsf{NP} \;\; (\mathsf{S}\backslash\mathsf{NP})\backslash(\mathsf{S}\backslash\mathsf{NP})}{\mathsf{S}\backslash\mathsf{NP} \quad \lambda x.\mathit{sings}(x) \land \mathit{dances}(x)} <^0$$

$$\frac{\mathsf{NP} \;\; \mathsf{S}\backslash\mathsf{NP}}{\mathsf{S} \quad \mathit{sings}(\mathit{mary}) \land \mathit{dances}(\mathit{mary})} <^0$$

[Evang, 2018, Introduction to Semantic Parsing with CCG]

CCG SEMANTIC PARSING

- E.g., ‘`Mary and John sing`’:

Mary	and	John	sing
NP	$((\mathsf{S}/(\mathsf{S}\backslash\mathsf{NP}))\backslash(\mathsf{S}/(\mathsf{S}\backslash\mathsf{NP})))/(\mathsf{S}/(\mathsf{S}\backslash\mathsf{NP}))$	NP	$\mathsf{S}\backslash\mathsf{NP}$
$mary$	$\lambda f.\lambda g.\lambda h.g(h) \wedge f(h)$	$john$	$\lambda x.sings(x)$

$$\frac{\mathsf{NP} : mary}{\mathsf{S}/(\mathsf{S}\backslash\mathsf{NP}) : \lambda f.f(mary)}\, T^{>} \qquad \frac{\mathsf{NP} : john}{\mathsf{S}/(\mathsf{S}\backslash\mathsf{NP}) : \lambda f.f(john)}\, T^{>}$$

$$\frac{\text{and} \quad \text{John}}{(\mathsf{S}/(\mathsf{S}\backslash\mathsf{NP}))\backslash(\mathsf{S}/(\mathsf{S}\backslash\mathsf{NP})) : \lambda g.\lambda h.g(h) \wedge h(john)}\, >^{0}$$

$$\frac{\text{Mary} \quad \text{and John}}{\mathsf{S}/(\mathsf{S}\backslash\mathsf{NP}) : \lambda h.h(mary) \wedge h(john)}\, <^{0}$$

$$\frac{\text{Mary and John} \quad \text{sing}}{\mathsf{S} : sing(mary) \wedge sing(john)}\, >^{0}$$

[Evang, 2018, Introduction to Semantic Parsing with CCG]

CCG SEMANTIC PARSING

- Recall that with syntactic CCG parsing, we could extend CKY and obtain a parsing algorithm with worst-case running time $O(n^6)$.
  - In the chart, we have a cell for each span `(i,j)`.
  - But for semantic parsing, we need a cell for each $(\texttt{i},\texttt{j},x)$ where `i < j` are sentence positions and x is any logical form.
  - The number of possible logical forms is very large.
- Thus, exact CCG semantic parsing is very expensive.
  - Non-polynomial running time.
- Instead, we typically use beam search:
  - For each span, only keep the top k search states.

LLMS AND AMBIGUITY

- We have established that ambiguity is ubiquitous in natural language.
  - Both syntactic and semantic ambiguity.
  - Humans automatically disambiguate sentences using context.
    - We don't even realize how widespread it is.
- Liu et al. (2023) designed a corpus of natural language entailment examples to test whether models correctly deal with ambiguity.
  - They tested for several different types of ambiguity.

LLMS AND AMBIGUITY

Example	Disambiguation 1	Disambiguation 2	Type
P: I'm afraid the cat was hit by a car. H: The cat was not hit by a car. {NEUTRAL, CONTRADICT} : [7 N, 2 C]	P: I'm worried... NEUTRAL : [9 N]	P: I'm sorry to share that... CONTRADICT : [9 C]	*Pragmatic* (44.8%)

LLMS AND AMBIGUITY

Example	Disambiguation 1	Disambiguation 2	Type
P: I'm afraid the cat was hit by a car. H: The cat was not hit by a car. ⟅NEUTRAL, CONTRADICT⟆ 🧑‍💻: [7 N, 2 C]	P: I'm worried... NEUTRAL 🧑‍💻: [9 N]	P: I'm sorry to share that... CONTRADICT 🧑‍💻: [9 C]	*Pragmatic* (44.8%)
P: John and Anna are married. H: John and Anna are not a couple. ⟅NEUTRAL, CONTRADICT⟆ 🧑‍💻: [5 N, 4 C]	P: ... are both married. NEUTRAL 🧑‍💻: [7 N, 2 E]	P: ... are married to each other. CONTRADICT 🧑‍💻: [9 C]	*Lexical* (20.0%)

[Liu et al., 2023]

LLMS AND AMBIGUITY

Example	Disambiguation 1	Disambiguation 2	Type
P: I'm afraid the cat was hit by a car. H: The cat was not hit by a car. {NEUTRAL, CONTRADICT} 🧑‍💻: [7 N, 2 C]	P: I'm worried... NEUTRAL 🧑‍💻: [9 N]	P: I'm sorry to share that... CONTRADICT 🧑‍💻: [9 C]	*Pragmatic* (44.8%)
P: John and Anna are married. H: John and Anna are not a couple. {NEUTRAL, CONTRADICT} 🧑‍💻: [5 N, 4 C]	P: ... are both married. NEUTRAL 🧑‍💻: [7 N, 2 E]	P: ... are married to each other. CONTRADICT 🧑‍💻: [9 C]	*Lexical* (20.0%)
P: This seminar is full now, but interesting seminars are being offered next quarter too. H: There will be more interesting seminars... {ENTAIL, NEUTRAL} 🧑‍💻: [7 E, 2 N]	H: There will be more seminars ... that are interesting. ENTAIL 🧑‍💻: [9 E]	H: There will be seminars... that are more interesting. NEUTRAL 🧑‍💻: [9 N]	*Syntactic* (8.6%)

[Liu et al., 2023]

LLMS AND AMBIGUITY

Example	Disambiguation 1	Disambiguation 2	Type
P: I'm afraid the cat was hit by a car. H: The cat was not hit by a car. {NEUTRAL, CONTRADICT} 🧑‍💻: [7 N, 2 C]	P: I'm worried... NEUTRAL 🧑‍💻: [9 N]	P: I'm sorry to share that... CONTRADICT 🧑‍💻: [9 C]	*Pragmatic* (44.8%)
P: John and Anna are married. H: John and Anna are not a couple. {NEUTRAL, CONTRADICT} 🧑‍💻: [5 N, 4 C]	P: ... are both married. NEUTRAL 🧑‍💻: [7 N, 2 E]	P: ... are married to each other. CONTRADICT 🧑‍💻: [9 C]	*Lexical* (20.0%)
P: This seminar is full now, but interesting seminars are being offered next quarter too. H: There will be more interesting seminars... {ENTAIL, NEUTRAL} 🧑‍💻: [7 E, 2 N]	H: There will be more seminars ... that are interesting. ENTAIL 🧑‍💻: [9 E]	H: There will be seminars... that are more interesting. NEUTRAL 🧑‍💻: [9 N]	*Syntactic* (8.6%)
P: The novel has been banned in many schools because of its explicit language. H: The novel has not been banned in many schools. {NEUTRAL, CONTRADICT} 🧑‍💻: [4 N, 5 C]	H: There are many schools where the novel has not been banned. NEUTRAL 🧑‍💻: [9 N]	H: It is not the case that the novel has been banned in many schools. CONTRADICT 🧑‍💻: [9 C]	*Scopal* (7.6%)

[Liu et al., 2023]

LLMS AND AMBIGUITY

Example	Disambiguation 1	Disambiguation 2	Type
P: It is currently March, and they plan to schedule their wedding for next December. H: They plan to schedule... for next year. {ENTAIL, CONTRADICT} 🧑‍💻: [3 E, 2 N, 4 C]	P: ... for December next year. ENTAIL 🧑‍💻: [9 E]	P: ... for the coming December. CONTRADICT 🧑‍💻: [9 C]	*Coreference* (2.9%)

[Liu et al., 2023]

LLMS AND AMBIGUITY

Example	Disambiguation 1	Disambiguation 2	Type
P: It is currently March, and they plan to schedule their wedding for next December. H: They plan to schedule... for next year. {ENTAIL, CONTRADICT} 🧑‍💻: [3 E, 2 N, 4 C]	P: ... for December next year. ENTAIL 🧑‍💻: [9 E]	P: ... for the coming December. CONTRADICT 🧑‍💻: [9 C]	*Coreference* (2.9%)
P: It is difficult to believe that the author of such a masterpiece could have been only 23 years old. H: The author of the masterpiece was only 23. {ENTAIL, NEUTRAL} 🧑‍💻: [3 E, 6 N]	P: It is shocking that... ENTAIL 🧑‍💻: [9 E]	P: It is questionable that... NEUTRAL 🧑‍💻: [9 N]	*Figurative* (1.9%)

[Liu et al., 2023]

LLMS AND AMBIGUITY

Example	Disambiguation 1	Disambiguation 2	Type
P: It is currently March, and they plan to schedule their wedding for next December. H: They plan to schedule... for next year. {ENTAIL, CONTRADICT}: [3 E, 2 N, 4 C]	P: ... for December next year. ENTAIL : [9 E]	P: ... for the coming December. CONTRADICT : [9 C]	*Coreference* (2.9%)
P: It is difficult to believe that the author of such a masterpiece could have been only 23 years old. H: The author of the masterpiece was only 23. {ENTAIL, NEUTRAL}: [3 E, 6 N]	P: It is shocking that... ENTAIL : [9 E]	P: It is questionable that... NEUTRAL : [9 N]	*Figurative* (1.9%)
P: A new study has found that nearly half of all Americans are in favor of gun control. H: The study found that half of all Americans are in favor of gun control. {ENTAIL, CONTRADICT}: [1 E, 2 N, 6 C]	H: ... that exactly half of all Americans... CONTRADICT : [8 C, 1 N]	H: ... that about half of all Americans... ENTAIL : [9 E]	*Other* (14.3%)

[Liu et al., 2023]

LLMS AND AMBIGUITY

- They test several LLMs with a handful of metrics:
  - Edit-F1: Compare the predicted with the reference disambiguation.
    - Treat them as unigrams and compute the F1 score.
  - Human: Judged correct by human annotators.
  - True/false accuracy: Only check the correctness of the final label (‘`true`’, ‘`false`’, vs ‘`inconclusive`’).

	EDIT-F1	Correct (human)	T/F Acc.
FLAN-T5	5.2	0.0	56.4
LLaMa	10.0	10.0	55.0
GPT-3	10.1	2.0	57.8
InstructGPT	14.5	4.0	49.6
ChatGPT	13.0	18.0	57.7
GPT-4	**18.0**	**32.0**	**63.0**

[Liu et al., 2023]

LLMS AND AMBIGUITY

- Saparina and Lapata (2025) proposed a method to improve text-to-SQL under ambiguity.

rating			
id	hotel_id	**stars**	**guest_score**
1	1	★★★★	8.5

hotels	
id	name
1	Radisson

Question:

return the **rating** of each hotel

Interpretations:

How many stars were assigned to each hotel?

```
SELECT h.name, r.stars FROM rating r JOIN hotels h
ON h.id = r.hotel_id
```

How did the customers review each hotel?

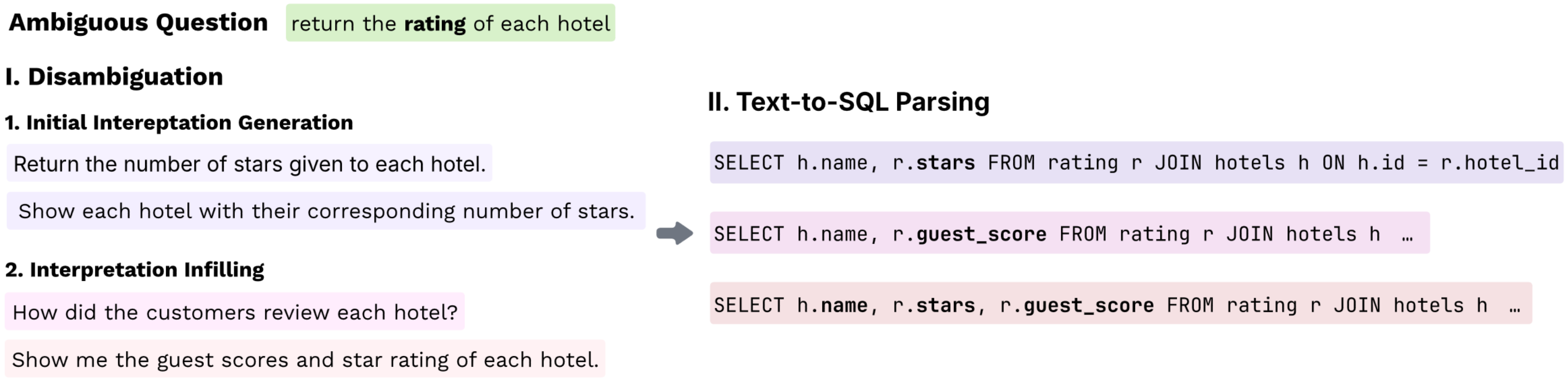
```
SELECT h.name, r.guest_score FROM rating r …
```

Show me the guest scores and star rating of each hotel.

```
SELECT h.name, r.stars, r.guest_score FROM rating r …
```

LLMS AND AMBIGUITY

- They propose a disambiguate-first parse-later approach:
  - They use an LLM (0-shot) to generate an initial list of interpretations.
  - But this list is often incomplete, so they train a supervised “infilling” model to predict missing interpretations.



[Saparina and Lapata, 2025]

LLMS AND AMBIGUITY

- They measure performance using two metrics:
  - Full interpretation coverage: Did you predict all valid interpretations?
  - Single interpretation coverage: Did you predict any valid interpretation?
- They test on two ambiguous SQL parsing datasets: AmbiQT and Ambrosia.

	AmbiQT		Ambrosia	
Method	Single	Full	Single	Full
End-to-End Text-to-SQL				
0-shot Prompt	62.3	12.3	29.4	0.9
3-shot Prompt	44.3	10.9	35.7	1.3
SFT	82.1	**63.2**	38.0	0.4
Disambiguate-and-Parse				
Interp. Prompt	81.8	26.0	81.9	16.9
w. Self-Correction	77.4	13.9	65.7	5.9
Gold Interp. SFT	87.4	61.2	62.6	0.3
Ours	**92.3**	53.2	**84.4**	**18.8**

[Saparina and Lapata, 2025]

QUESTIONS?