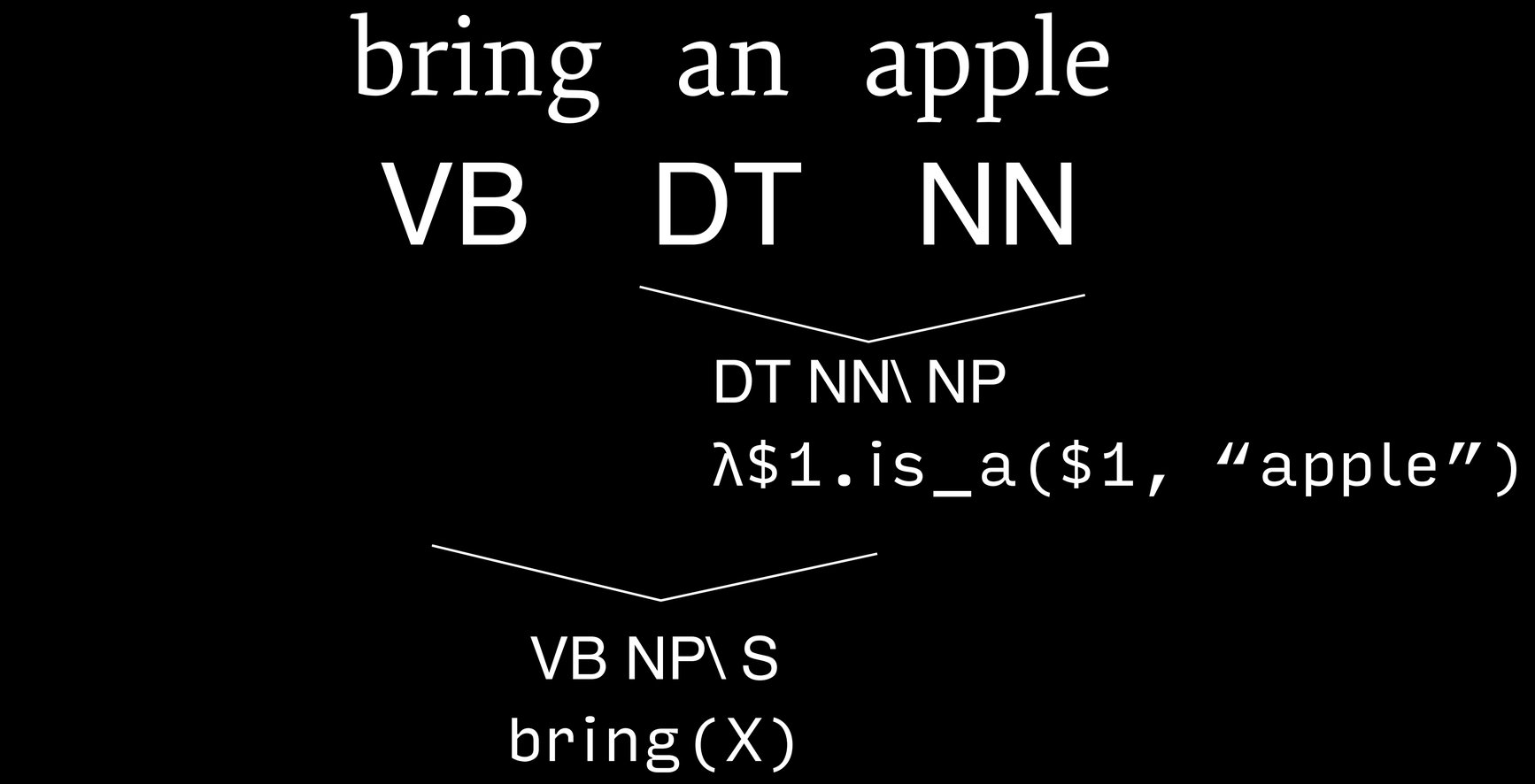
Neural Semantic Parsing with Anonymization for Command Understanding in General-Purpose Service Robots

Nick Walker, Yu-Tang Peng, Maya Cakmak

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING


Bring me an apple

Traditional semantic parsing



Zettlemoyer and Collins 2007

Neural machine translation

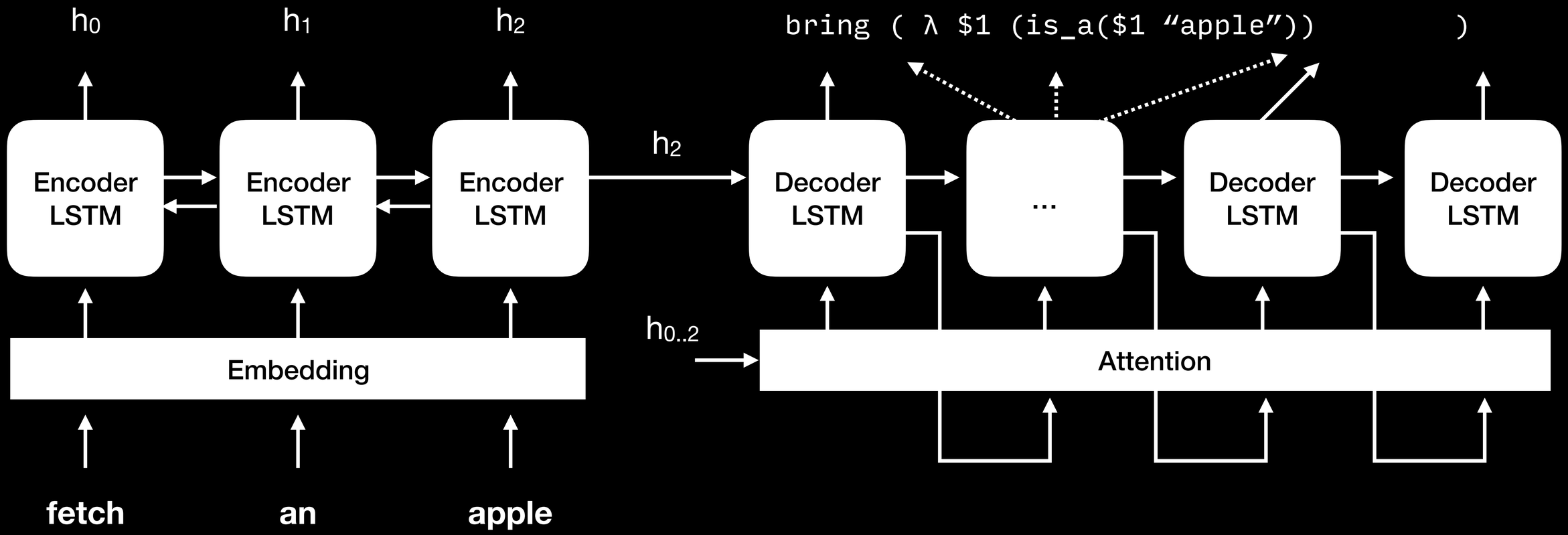
bring an apple

f ↓

bring(λ $1 (is_a($1 “apple”)))

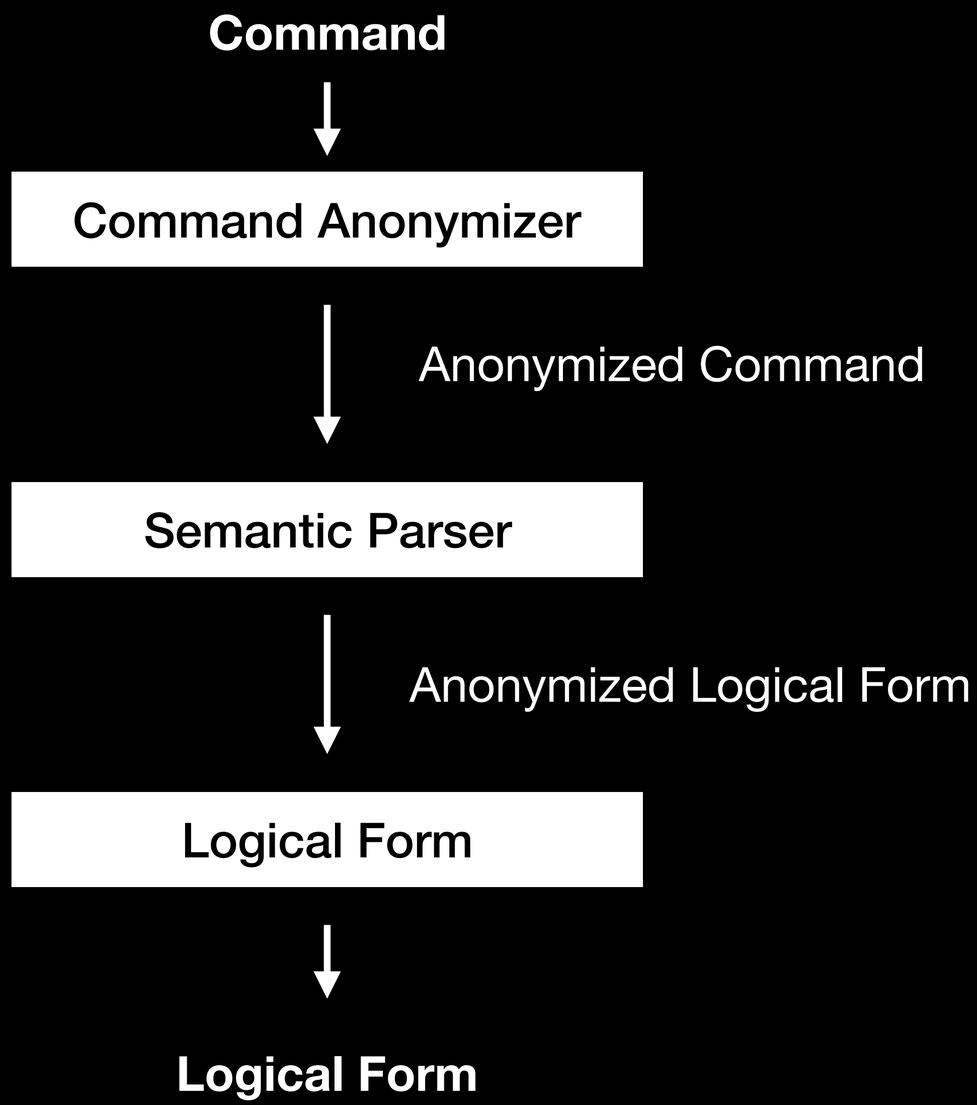
Dong and Lapata 2016

Neural machine translation

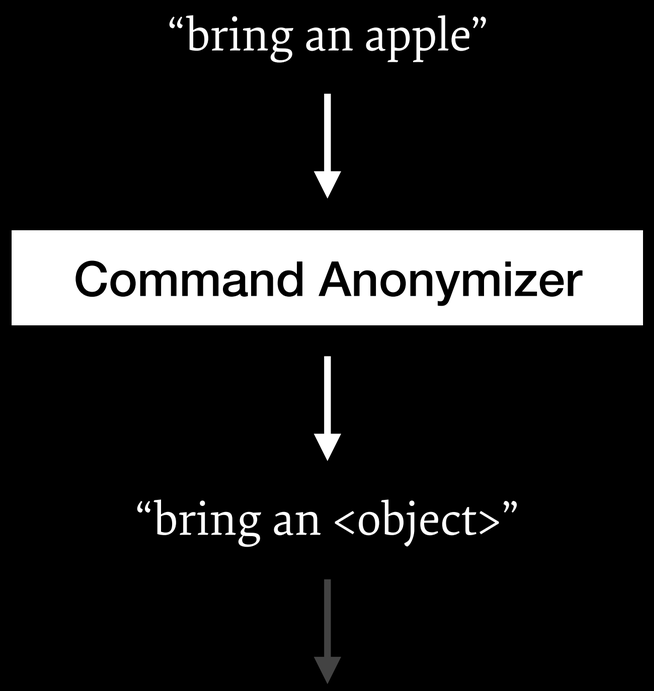


Dong and Lapata 2016

How do we frame command understanding so that neural semantic parsing methods can work for robotics domains?

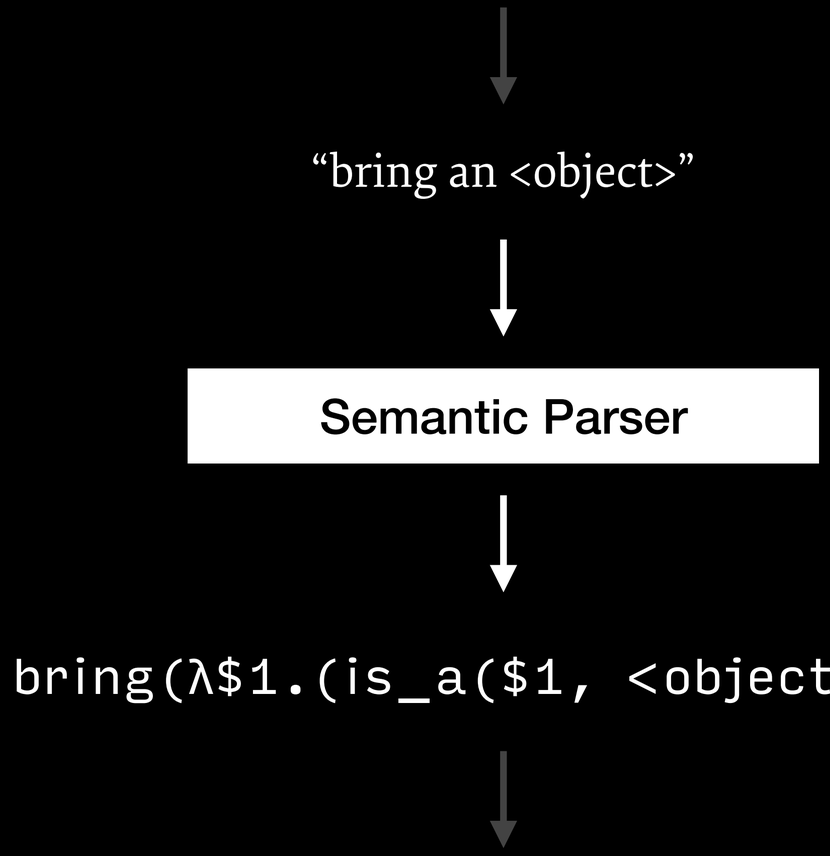

Command
Command Anonymizer
Anonymized Command
Semantic Parser
Anonymized Logical Form
Logical Form
Logical Form

Simplification: Anonymize Commands



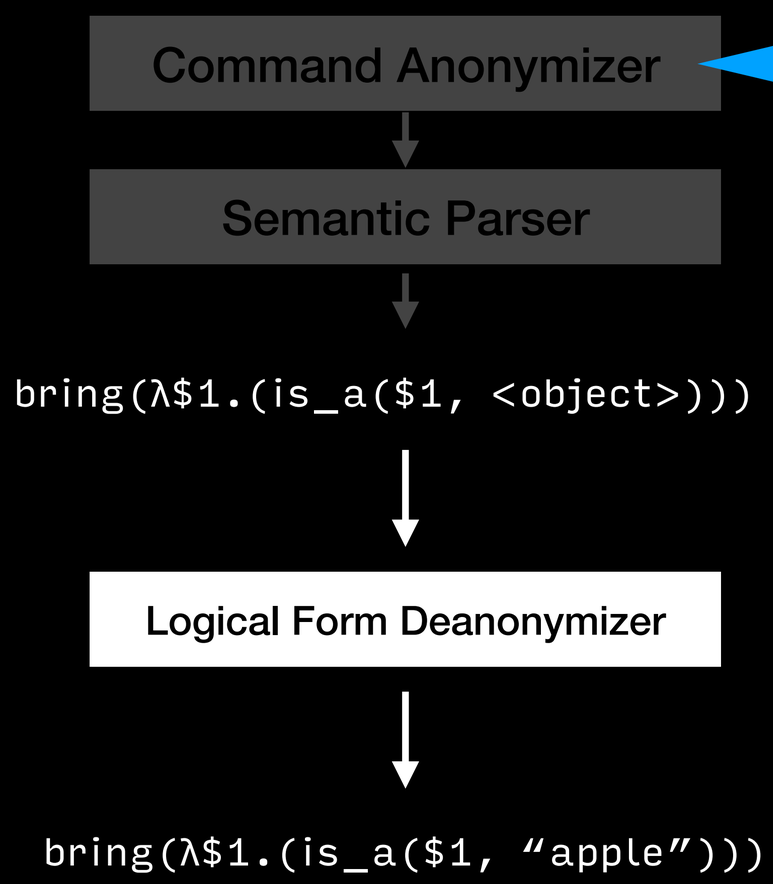
- use robot’s ontology to simplify commands
- improve predictability
- hit or miss

Simplification:
Anonymized Logical Representation

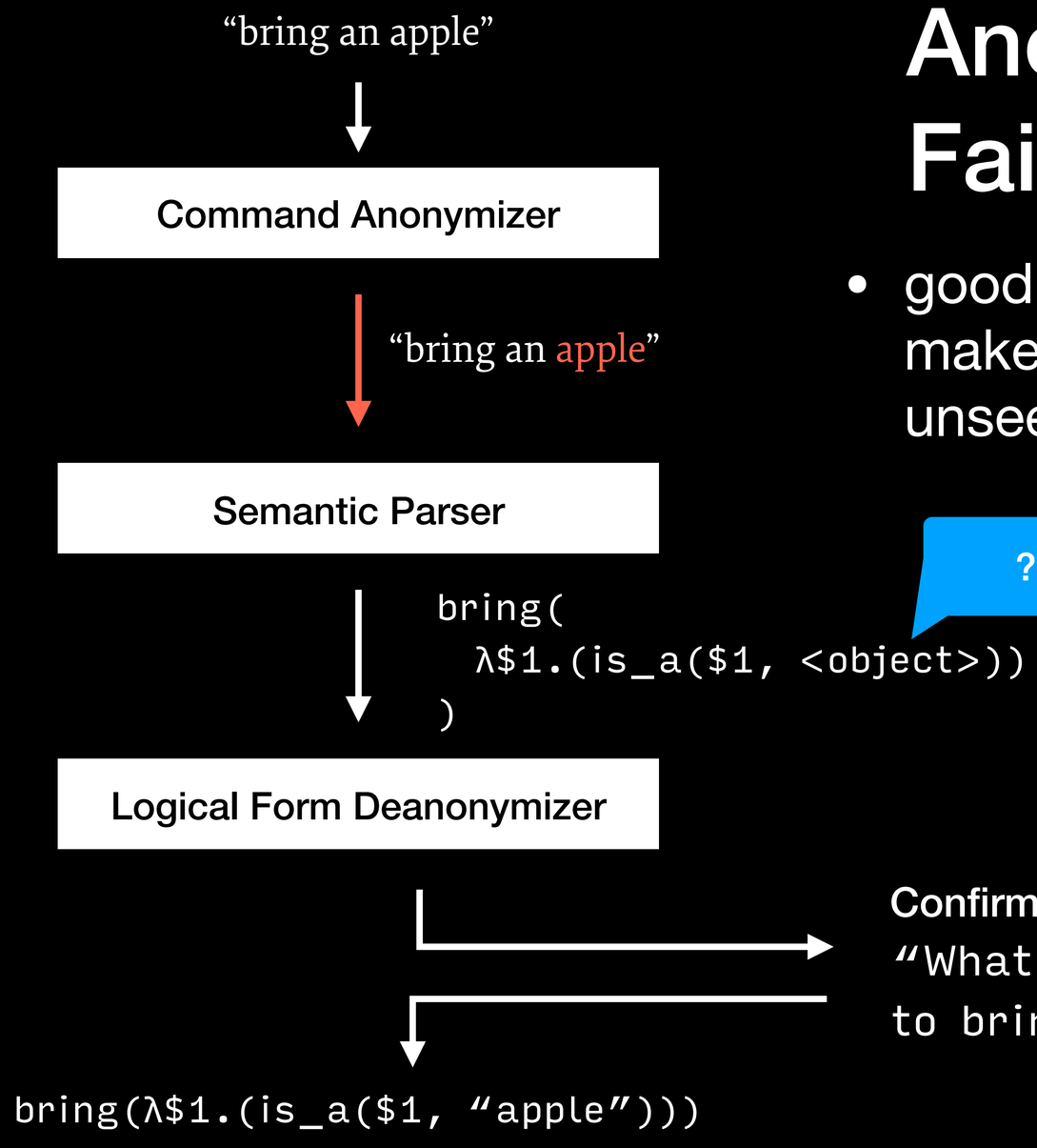


- skip argument assignment
- no longer executable

Getting executability back

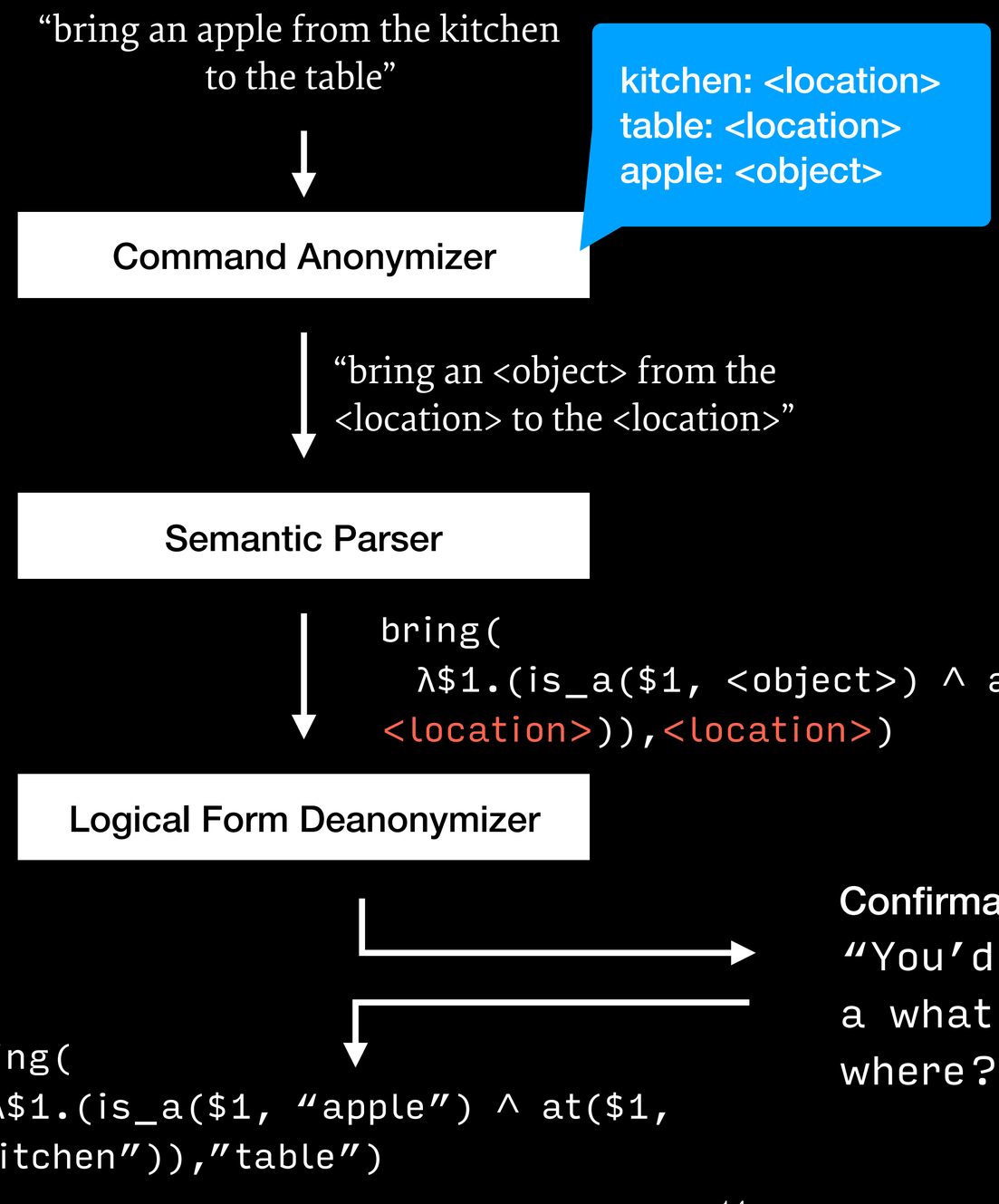


- if command anonymization worked, deanonymization can be straightforward
- but in other cases…



Anonymization Failure

- good word embeddings make parser robust to unseen entities



Deanonymization Recovery

- ambiguous cases
- handle in dialogue

How well does a neural semantic parser work under this regime in a robotics domain?

Getting Annotated Data

```
$vbbring me the $object =
bring(λ$1.(is a($1, $object)))
```

Getting Annotated Data

- Modified RoboCup@Home 2018 GPSR command generator
- Generates command + logical form
  - 21 predicates
- 125 annotations →1211 anonymized commands
- 101 anonymized logical forms

Paraphrased Data

- Different words, same meaning
- 1836 paraphrases from 95 crowd workers
- Reasonable validation checks required

Generated:
“tell me how many coke there are on the freezer”

Paraphrased:
“how many cokes are left in the freezer”

Experiment

- Split paraphrased data 70/10/20%
- Train and tune, then test on the held out 20%
- Measure accuracy, percentage of exact match predictions
- Assume ontology is empty! Command anonymization always fails

percentage accuracy (↑)

Train	Paraphrased
Test	Paraphrased
Grammar-based Oracle	
k-Nearest Neighbors	

percentage accuracy (↑)

Train	Paraphrased
Test	Paraphrased
Grammar-based Oracle	
k-Nearest Neighbors	
seq2seq	
+ GloVe	
+ GloVe;ELMo	
+ GloVe;OpenAI	
+ GloVe;$BERT_{base}$	
+ GloVe;$BERT_{large}$	

percentage accuracy (↑)

Train	Paraphrased
Test	Paraphrased
Grammar-based Oracle	1.1
k-Nearest Neighbors	42.8
seq2seq	64.4
+ GloVe	70.2
+ GloVe;ELMo	77.3
+ GloVe;OpenAI	78.2
+ GloVe;$\text{BERT}_{\text{base}}$	75.4
+ GloVe;$\text{BERT}_{\text{large}}$	**78.5**

percentage accuracy (↑)

Train	Paraphrased	Gen. + Paraphrased
Test	Paraphrased	Paraphrased
Grammar-based Oracle	1.1	1.1
k-Nearest Neighbors	42.8	49.8
seq2seq	64.4	79.6
+ GloVe	70.2	85.3
+ GloVe;ELMo	77.3	85.4
+ GloVe;OpenAI	78.2	89.0
+ GloVe;$BERT_{base}$	75.4	87.6
+ GloVe;$BERT_{large}$	**78.5**	**89.4**

You can train neural semantic parsers that work well for command-taking dialogues in robots.

Use our code and data!

`github.com/nickswalker/gpsr-command-understanding`

- Train your own models
- Use our baselines
- Beat our performance (data, splits available)
- Hack on a Python version of the command generator



Paper

Neural Semantic Parsing with Anonymization for Command Understanding in General-Purpose Service Robots

Nick Walker, Yu-Tang Peng, Maya Cakmak

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Slides DOI 10.5281/zenodo.3253252

arxiv.org/abs/1907.01115