Neural Symbolic Machines

Semantic Parsing on Freebase with Weak Supervision

Chen Liang, Jonathan Berant, Quoc Le, Kenneth Forbus, Ni Lao

Research at Google

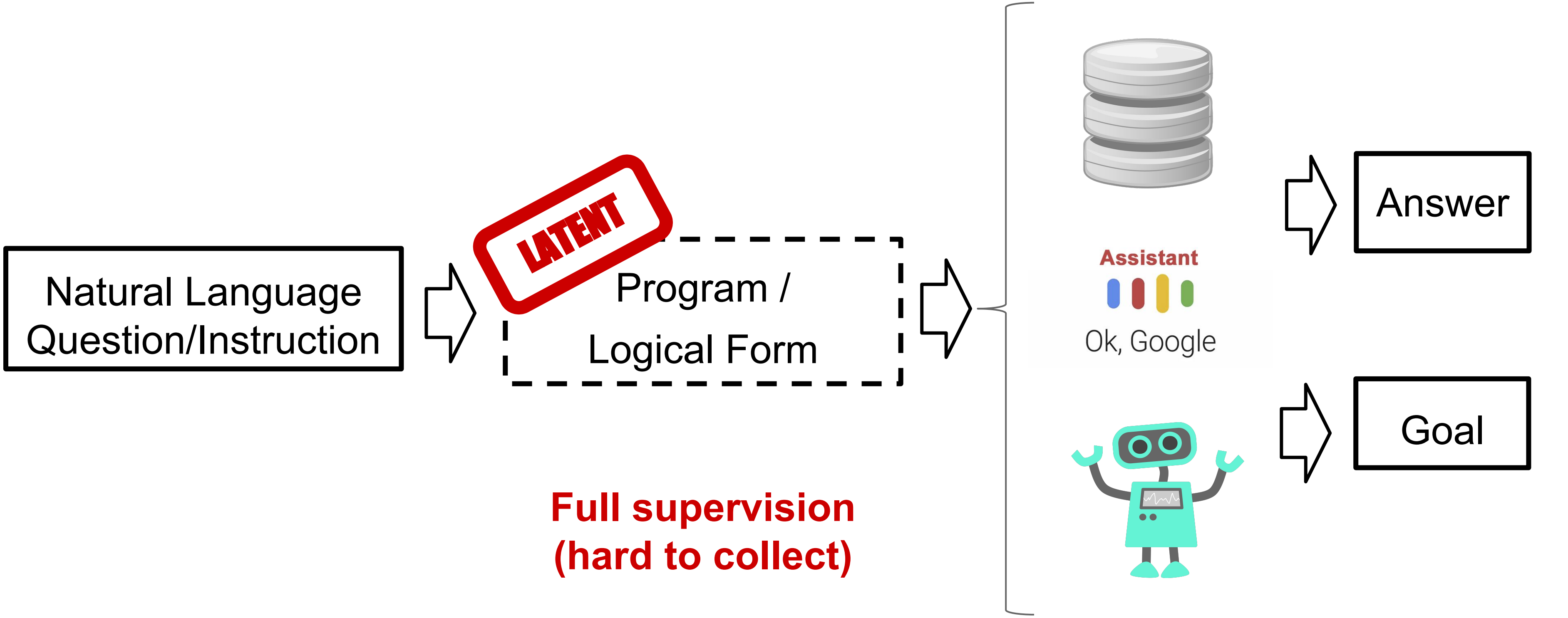
NORTHWESTERN UNIVERSITY

TEL AVIV UNIVERSITY

Overview

- Motivation: Semantic Parsing and Program Induction
- Neural Symbolic Machines
    - Key-Variable Memory
    - Code Assistance
    - Augmented REINFORCE
- Experiments and analysis

Semantic Parsing: Language to Programs



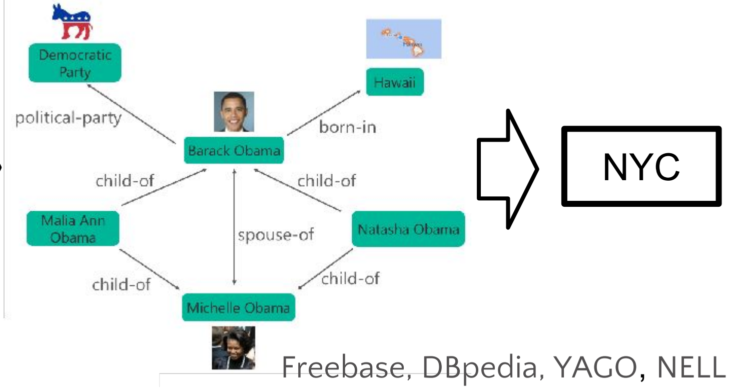
Weak supervision
(easy to collect)

[Berant, et al 2013; Liang 2013]

Question Answering with Knowledge Base

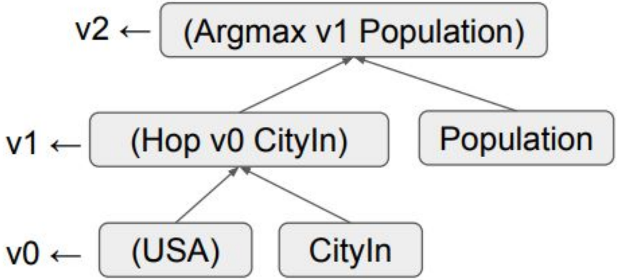
Largest city in US?

```
GO
(Hop V1 CityIn)
(Argmax V2 Population)
RETURN
```



NYC

1. Compositionality



2. Large Search Space

Freebase:
23K predicates, 82M entities, 417M triplets

WebQuestionsSP Dataset

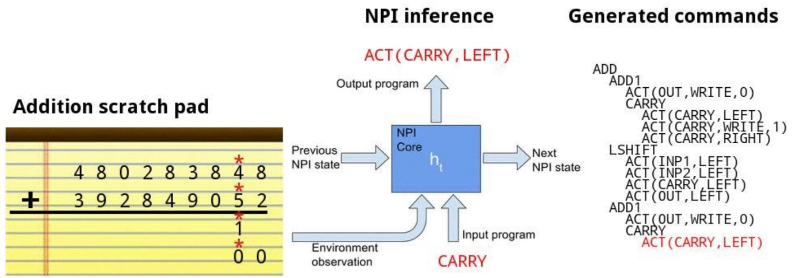
- 5,810 questions Google Suggest API & Amazon MTurk[1]
- Remove invalid QA pairs[2]
- 3,098 training examples, 1,639 testing examples remaining
- Open-domain, and contains grammatical error
- Multiple entities as answer => macro-averaged F1

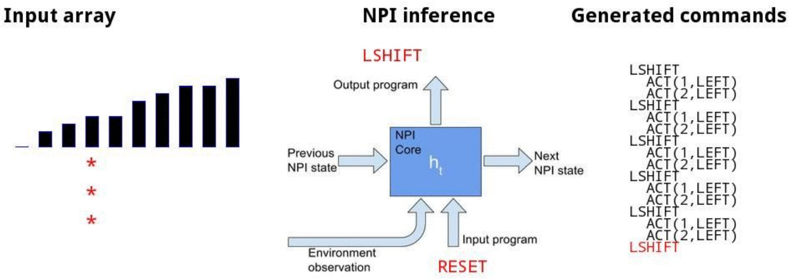
Grammatical error	Multiple entities
What do Michelle Obama do for a living?	writer, lawyer
What character did Natalie Portman play in Star Wars?	Padme Amidala
What currency do you use in Costa Rica?	Costa Rican colon
What did Obama study in school?	political science
What killed Sammy Davis Jr?	throat cancer

[Berant et al, 2013; Yih et al, 2016]

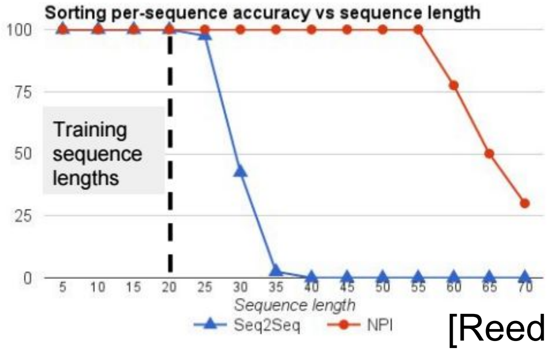
(Scalable) Neural Program Induction

- Impressive works to show NN can learn addition and sorting, but...



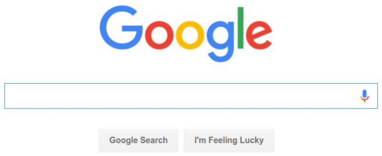


- The learned operations are not as scalable and precise.



[Reed & Freitas 2015]

- Why not use existing modules that are scalable, precise and interpretable?

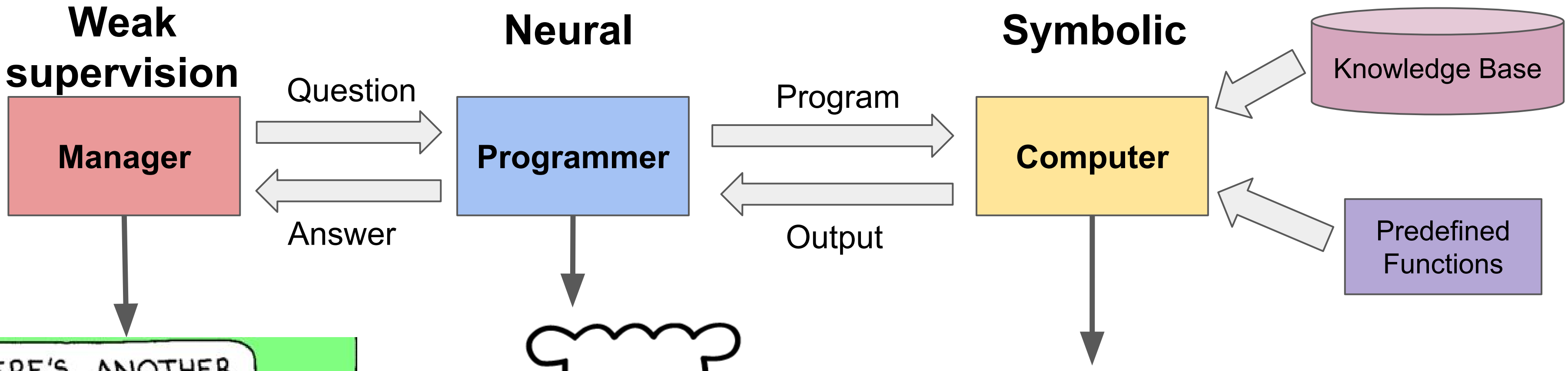


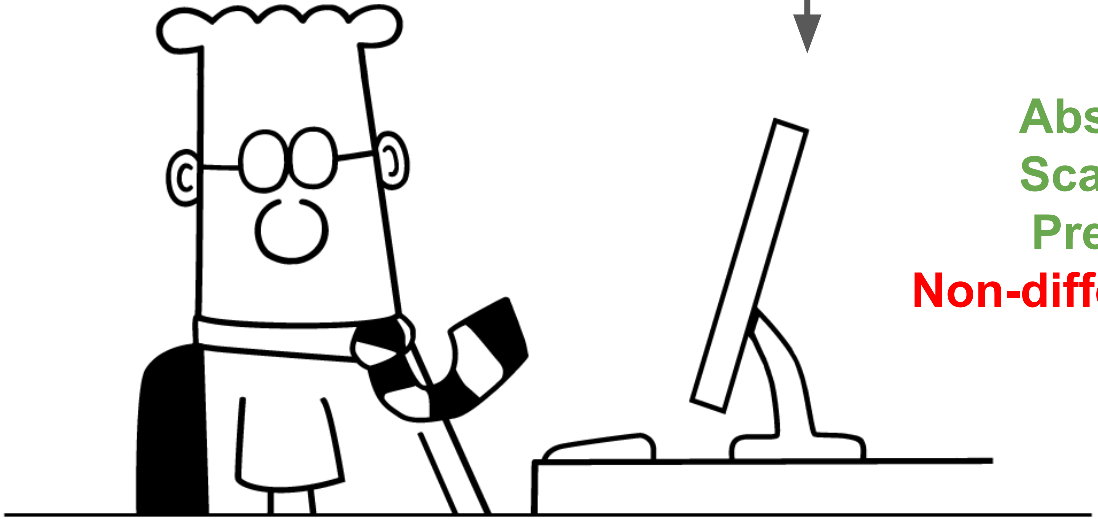
[Zaremba & Sutskever 2016]

Overview

- Motivation: Semantic Parsing and Program Induction
- Neural Symbolic Machines
  - Key-Variable Memory
  - Code Assistance
  - Augmented REINFORCE
- Experiments and analysis

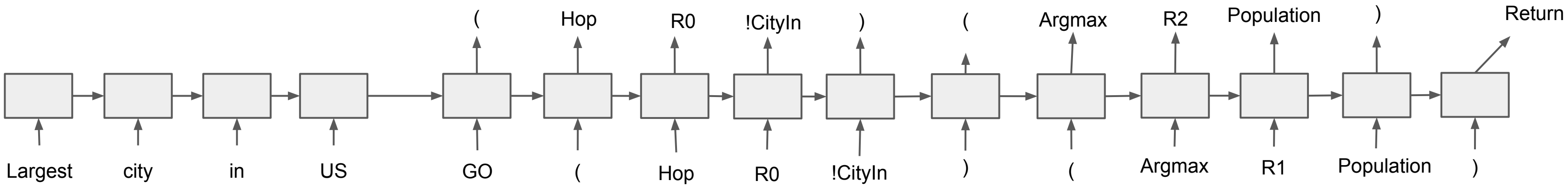
Neural Symbolic Machines



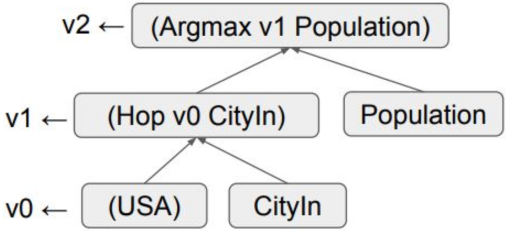


Abstract
Scalable
Precise
Non-differentiable

Simple Seq2Seq model is not enough



1. Compositionality



2. Large Search Space

23K predicates,
82M entities,
417M triplets

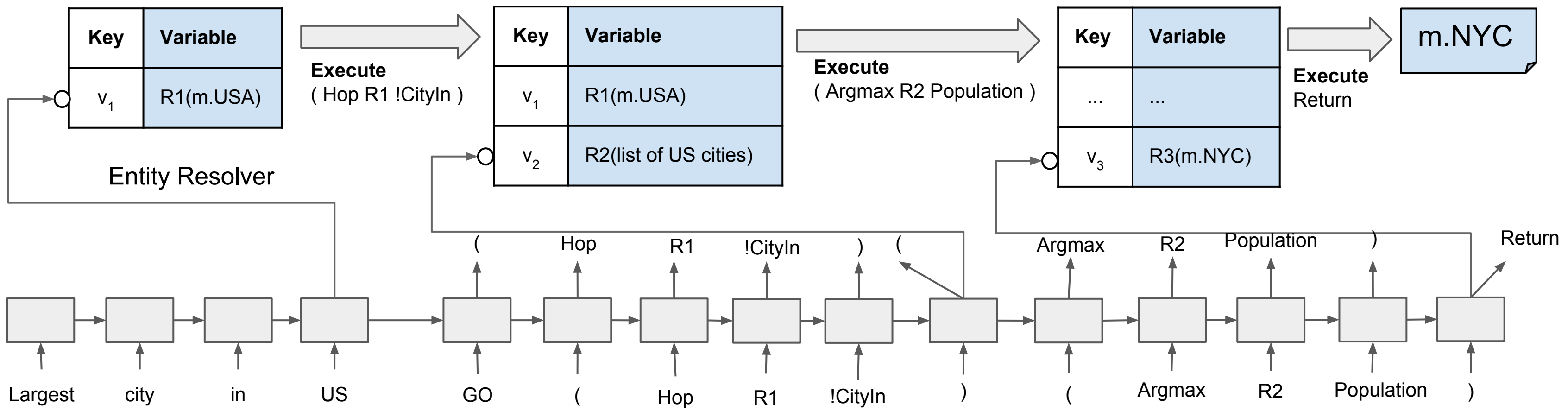
1.Key-Variable Memory

2.Code Assistance
3.Augmented REINFORCE

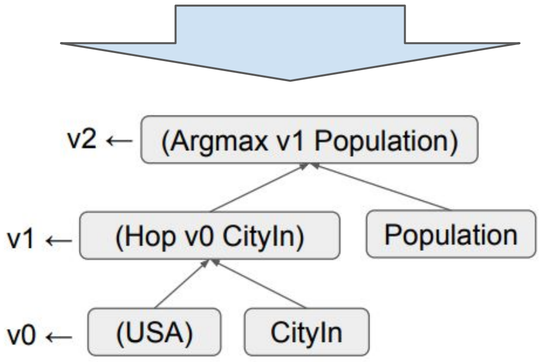
Overview

- Motivation: Semantic Parsing and Program Induction
- Neural Symbolic Machines
  - Key-Variable Memory
  - Code Assistance
  - Augmented REINFORCE
- Experiments and analysis

Key-Variable Memory for Compositionality

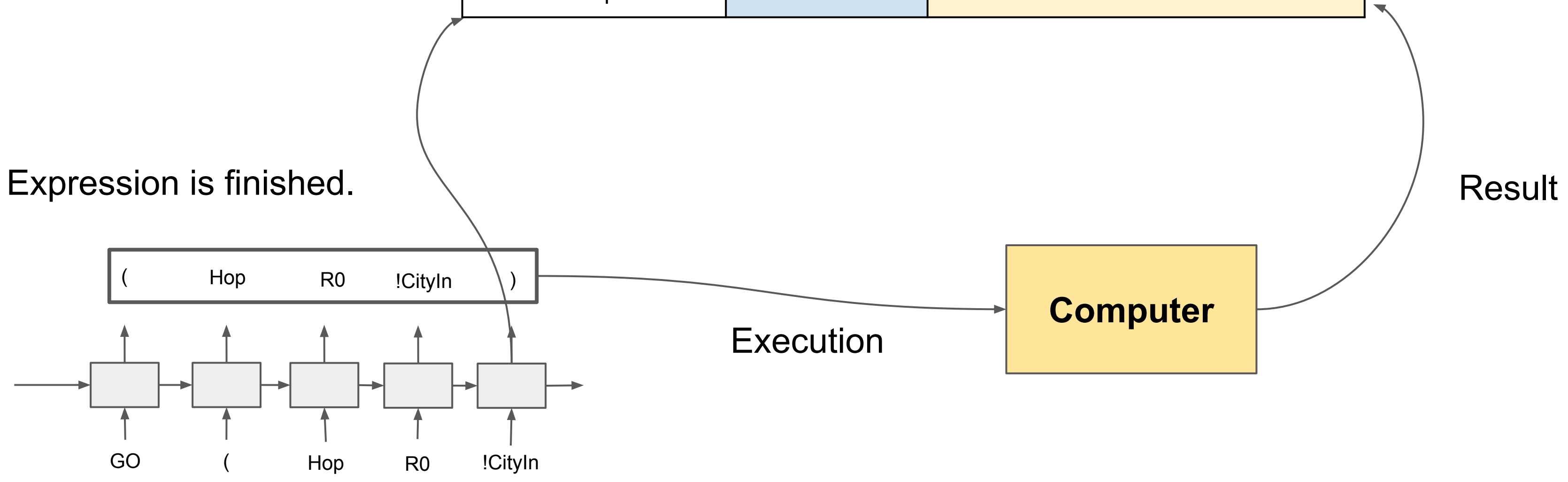


- A linearised bottom-up derivation of the recursive program.



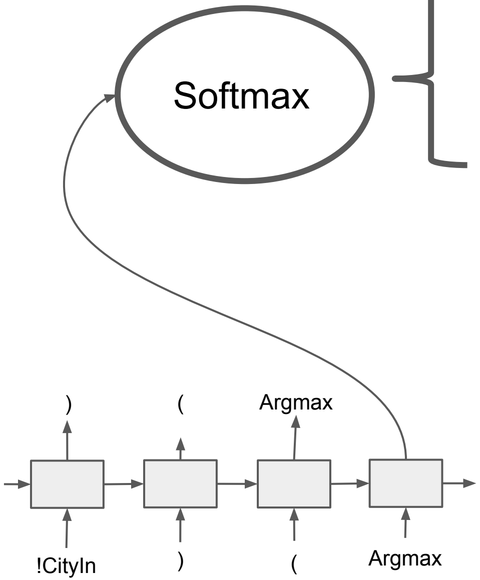
Key-Variable Memory: Save Intermediate Value

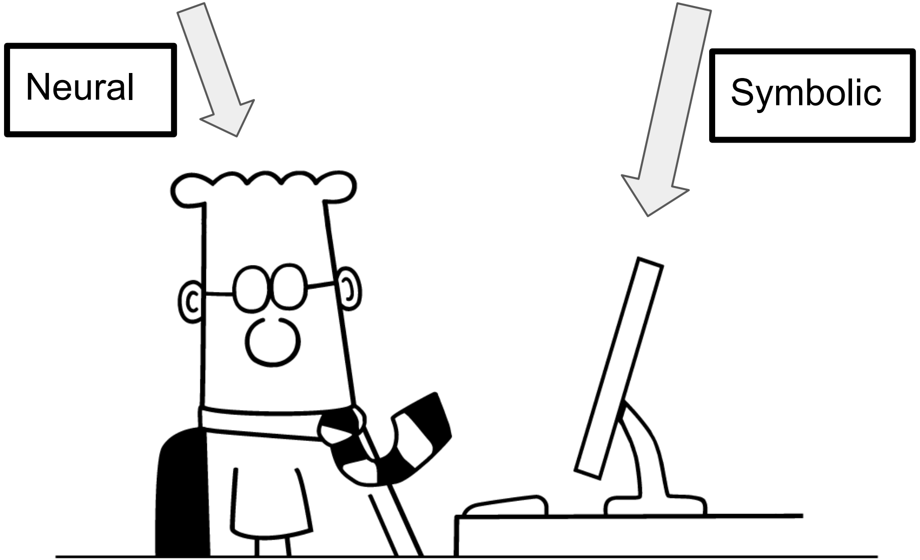
Key (Embedding)	**Variable** (Symbol)	**Value** (Data in Computer)
V_0	`R0`	`m.USA`
V_1	`R1`	`[m.SF, m.NYC, ...]`



Key-Variable Memory: Reuse Intermediate Value

Key (Embedding)	**Variable** (Symbol)	**Value** (Data in Computer)
V_0	`R0`	`m.USA`
V_1	`R1`	`[m.SF, m.NYC, ...]`



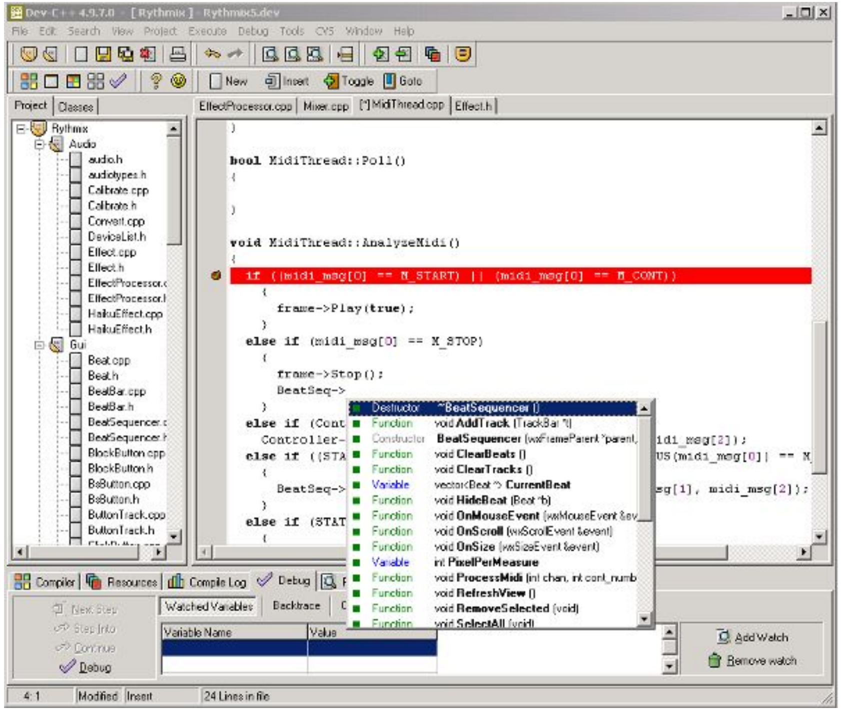


Overview

- Motivation: Semantic Parsing and Program Induction
- Neural Symbolic Machines
    - Key-Variable Memory
    - Code Assistance
    - Augmented REINFORCE
- Experiments and analysis

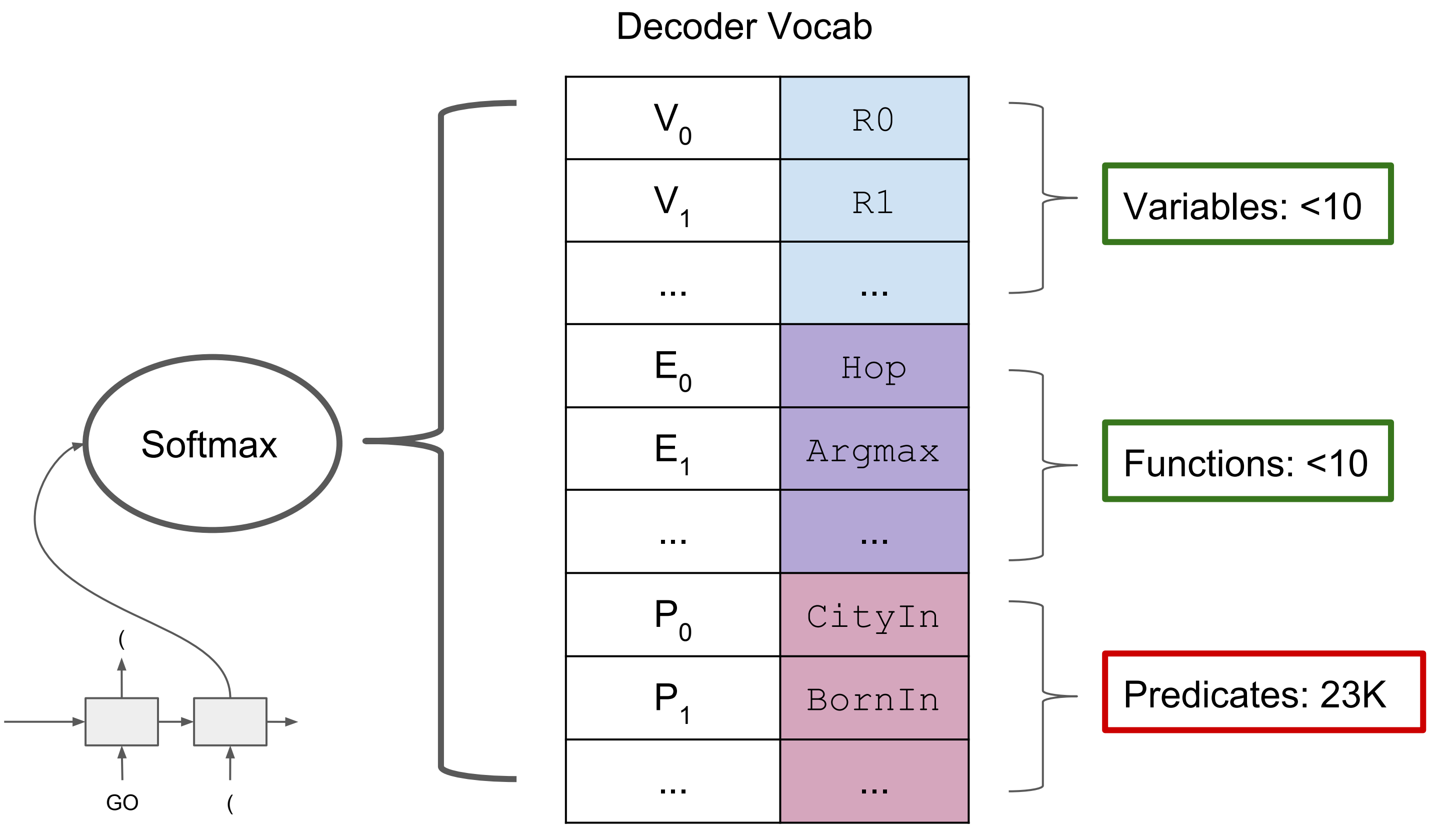
Code Assistance: Prune Search Space

Pen and paper



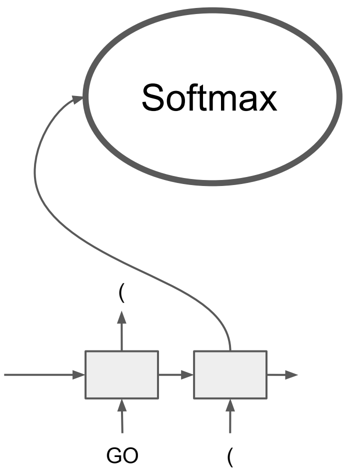
IDE

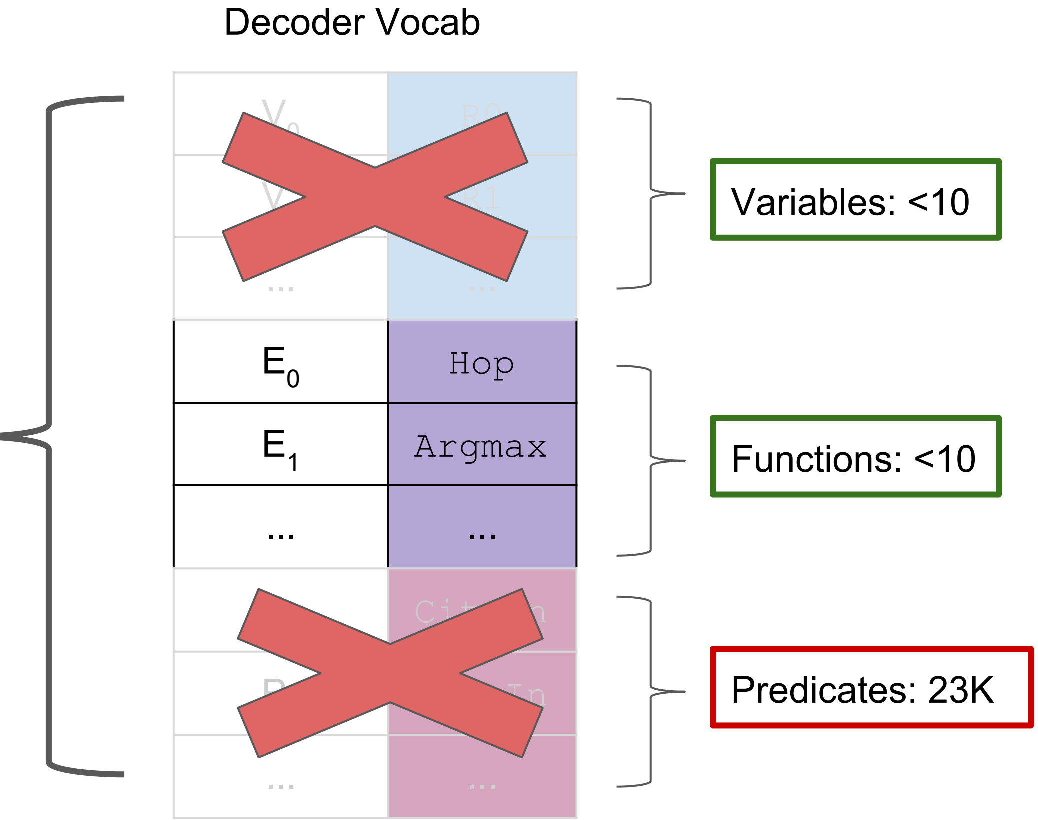
Code Assistance: Syntactic Constraint



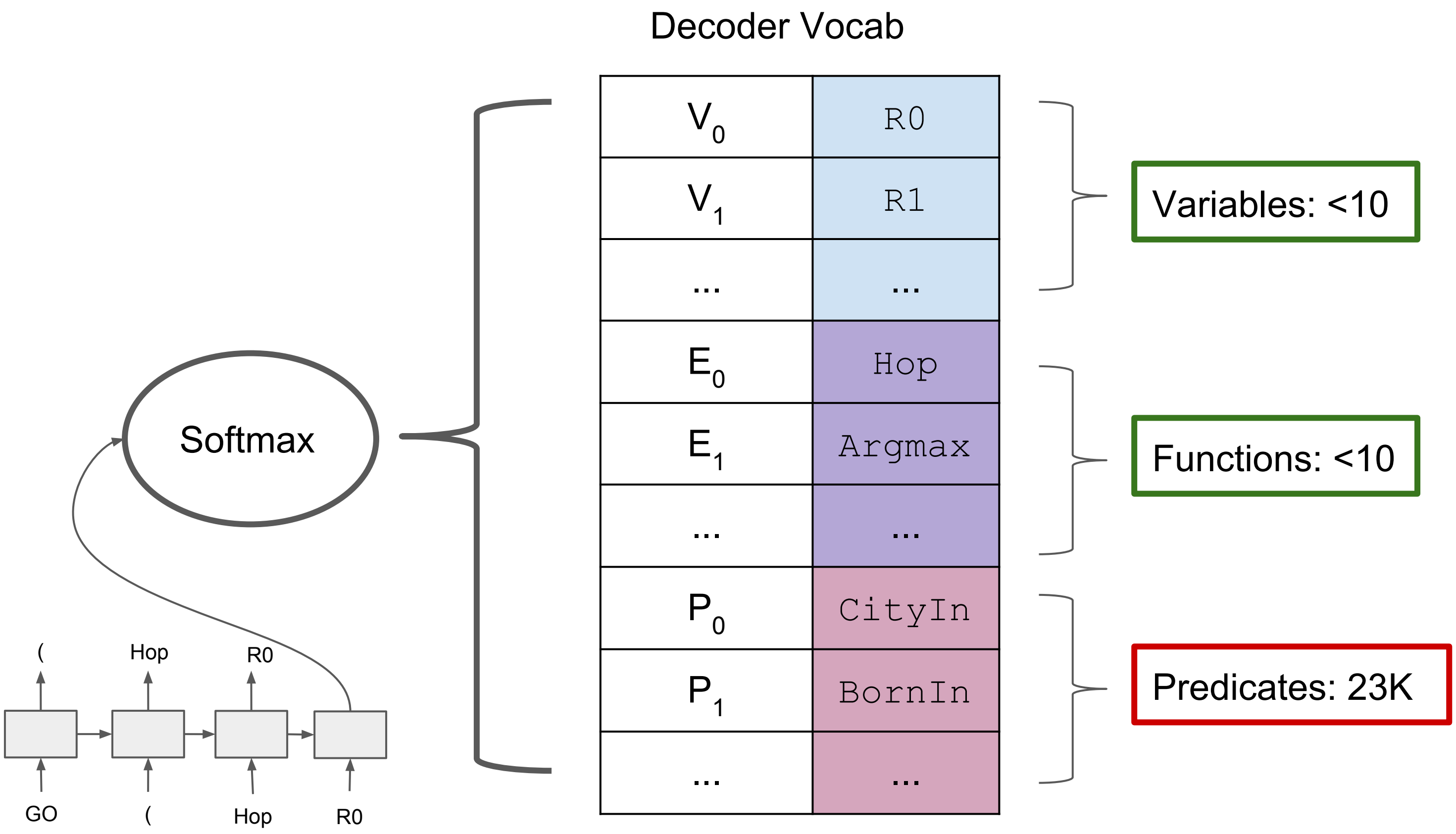
Code Assistance: Syntactic Constraint

Last token is '(', so has to output a function name next.



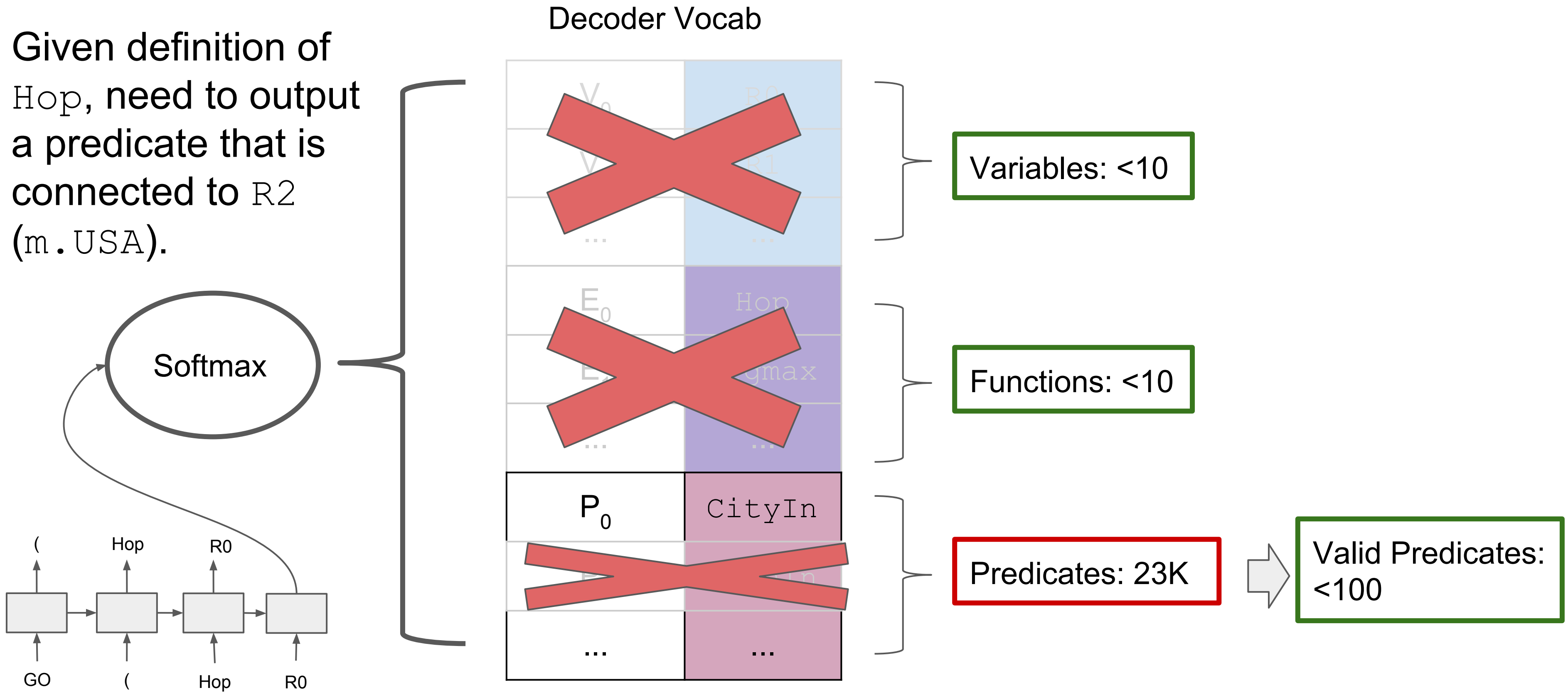


Code Assistance: Semantic Constraint



Code Assistance: Semantic Constraint

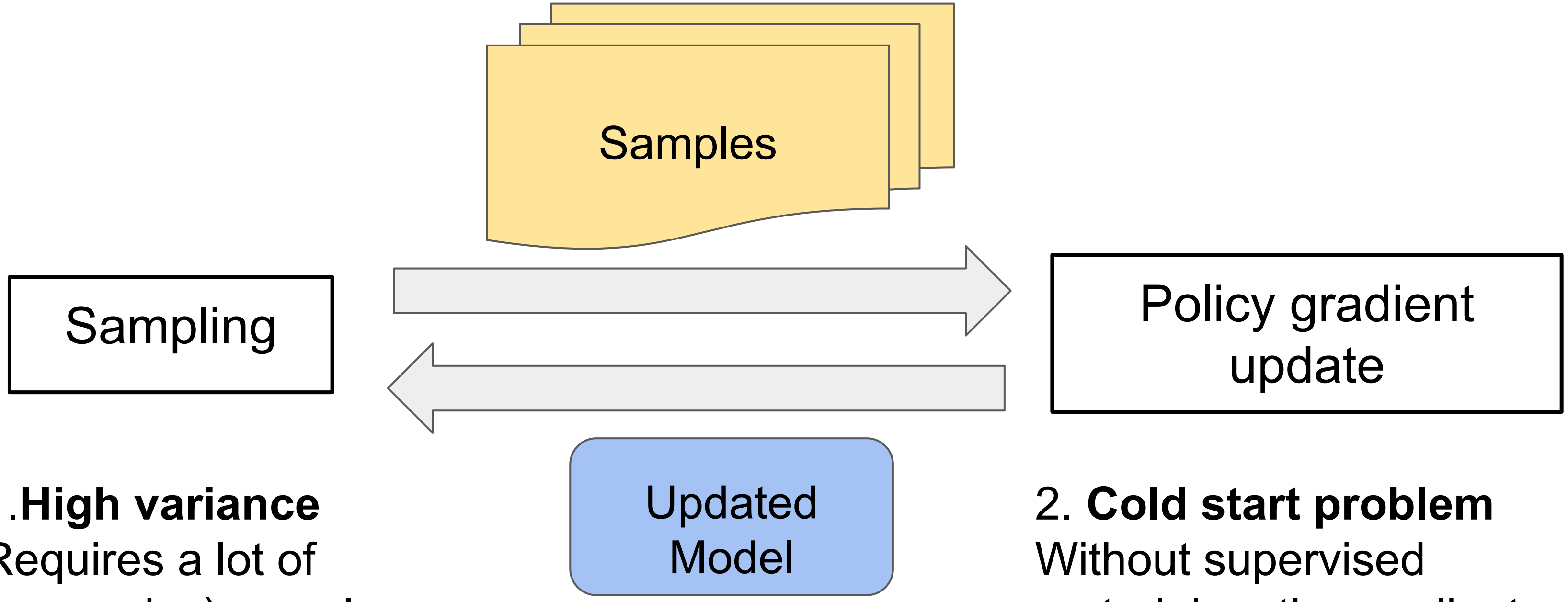
Given definition of `Hop`, need to output a predicate that is connected to `R2` (`m.USA`).



Overview

- Motivation: Semantic Parsing and Program Induction
- Neural Symbolic Machines
  - Key-Variable Memory
  - Code Assistance
  - Augmented REINFORCE
- Experiments and analysis

REINFORCE Training

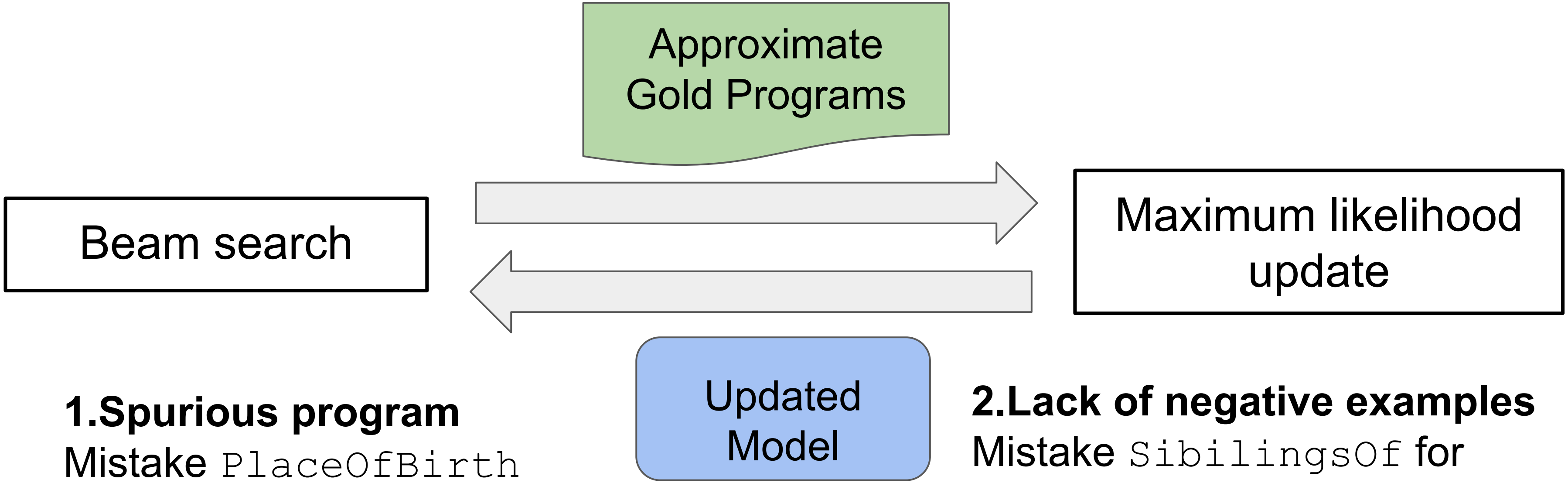


1.**High variance**
Requires a lot of (expensive) samples

2. **Cold start problem**
Without supervised pretraining, the gradients at the beginning

$$\nabla_\theta J^{RL}(\theta) = \sum_q \sum_{a_{0:T}} P(a_{0:T}|q,\theta)[R(q,a_{0:T}) - B(q)]\nabla_\theta \log P(a_{0:T}|q,\theta)$$

Iterative Maximum Likelihood Training (Hard EM)

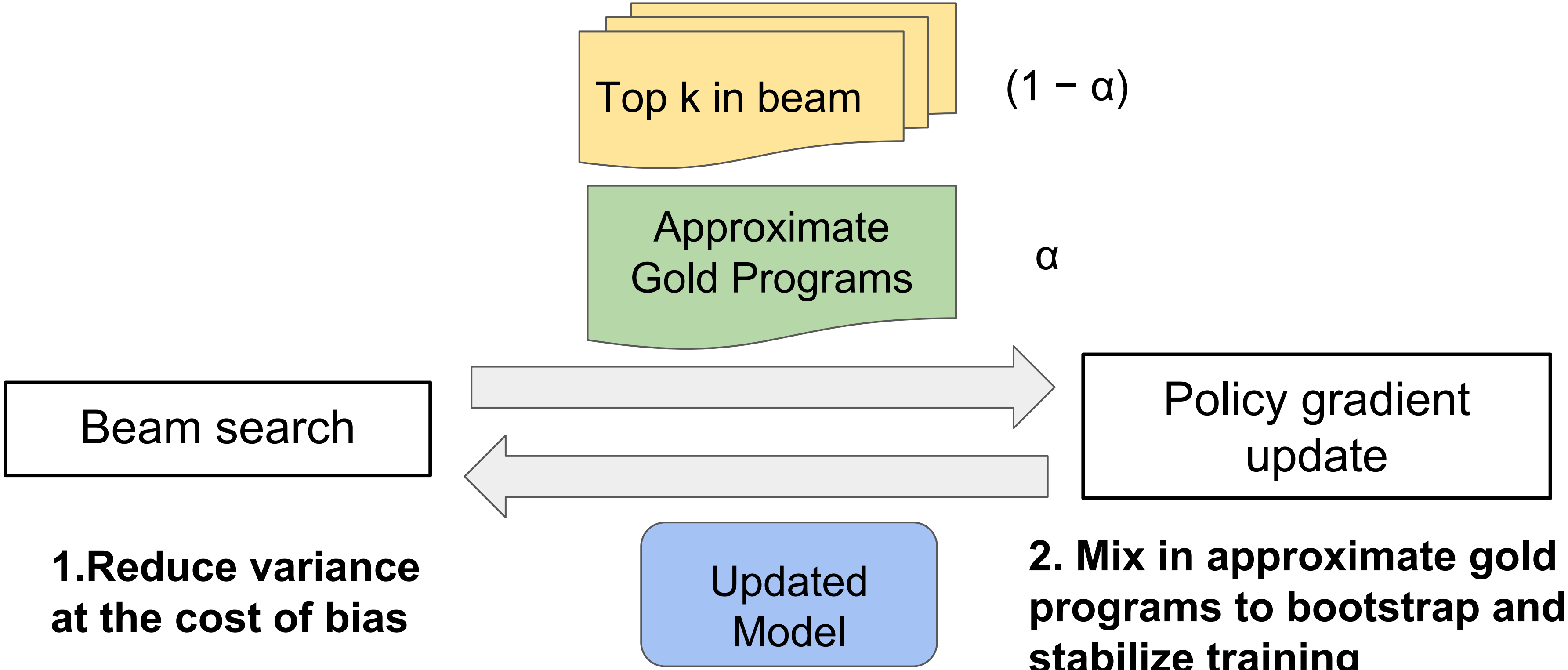


1.Spurious program
Mistake `PlaceOfBirth` for `PlaceOfDeath`.

2.Lack of negative examples
Mistake `SibilingsOf` for `ParentsOf`.

$$J^{ML}(\theta) = \sum_{q} \log P(a_{0:T}^{best}(q)|q, \theta)$$

Augmented REINFORCE



1.Reduce variance at the cost of bias

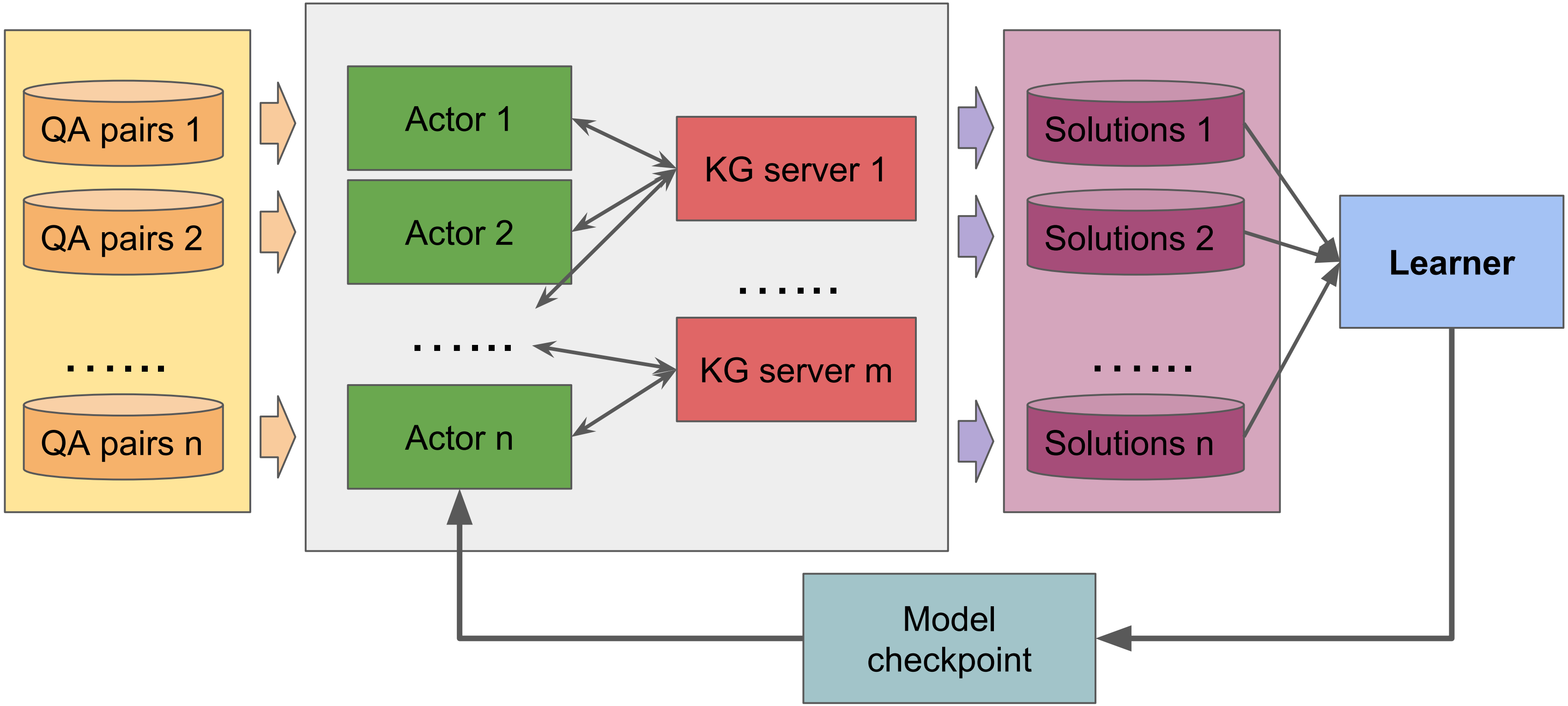
2. Mix in approximate gold programs to bootstrap and stabilize training

Overview

- Motivation: Semantic Parsing and Program Induction
- Neural Symbolic Machines
    - Key-Variable Memory
    - Code Assistance
    - Augmented REINFORCE
- Experiments and analysis

Distributed Architecture

- 200 actors, 1 learner, 50 Knowledge Graph servers



Generated Programs

- **Question**: “what college did russell wilson go to?”
- **Generated program**:

```
(hop v1 /people/person/education)
(hop v2 /education/education/institution)
(filter v3 v0 /common/topic/notable_types )
<EOP>
```

In which

```
v0 = “College/University” (m.01y2hnl)
v1 = “Russell Wilson” (m.05c10yf)
```

- **Distribution of the length of generated programs**

#Expressions	0	1	2	3
Percentage	0.4%	62.9%	29.8%	6.9%
F1	0.0	73.5	59.9	70.3

New State-of-the-Art on ***WebQuestionsSP***

- First end-to-end neural network to achieve SOTA on semantic parsing with weak supervision over large knowledge base
- The performance is approaching SOTA with full supervision

Model	Avg. Prec.@1	Avg. Rec.@1	Avg. F1@1	Acc.@1
STAGG	67.3	73.1	66.8	58.8
NSM – our model	70.8	76.0	**69.0**	59.5
STAGG (full supervision)	70.9	80.3	71.7	63.9

Augmented REINFORCE

- REINFORCE get stuck at local maxima
- Iterative ML training is not directly optimizing the F1 score
- Augmented REINFORCE obtains the best performances

Settings	Train Avg. F1@1	Valid Avg. F1@1
iterative ML only	68.6	60.1
REINFORCE only	55.1	47.8
Augmented REINFORCE	83.0	**67.2**

Weak supervision

Neural

Symbolic

Manager → Question → Programmer

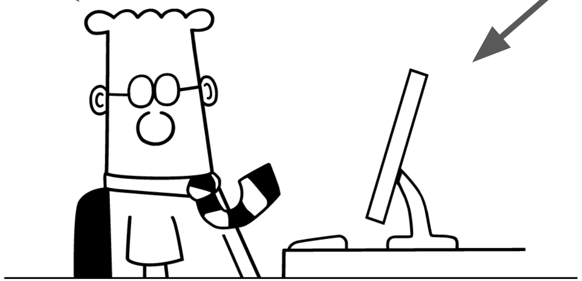
Programmer → Answer → Manager

Programmer → Programs → Computer

Computer → Outputs → Programmer

Knowledge Base → Computer

Predefined Functions → Computer



Augmented REINFORCE

Key-Variable Memory

Code Assistance

Thanks!