Neural Symbolic Machines
Semantic Parsing on Freebase with Weak Supervision

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

Natural Language Question/Instruction → Program / Logical Form

Full supervision (hard to collect)

Program / Logical Form → Assistant

Assistant Ok, Google → Goal

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

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

2. Large Search Space
WebQuestionsSP Dataset

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

Grammatical error

- What do Michelle Obama do for a living?
- What character did Natalie Portman play in Star Wars?
- What currency do you use in Costa Rica?
- What did Obama study in school?
- What killed Sammy Davis Jr?

Multiple entities

- writer, lawyer
- Padme Amidala
- Costa Rican colon
- political science
- 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.

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

[Reed & Freitas 2015]

[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

Weak supervision
Manager

Neural
Programmer

Symbolic
Computer

Knowledge Base
Predefined Functions

Abstract
Scalable
Precise
Non-differentiable

HERE'S ANOTHER SHOVEL FULL OF ASSIGNMENTS.
Simple Seq2Seq model is not enough

1. Compositionality
   - \( v_2 \leftarrow \text{Argmax } v_1 \text{ Population} \)
   - \( v_1 \leftarrow \text{Hop } v_0 \text{ CityIn} \) and \( \text{Population} \)
   - \( v_0 \leftarrow \text{USA} \) and \( \text{CityIn} \)

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

<table>
<thead>
<tr>
<th>Key (Embedding)</th>
<th>Variable (Symbol)</th>
<th>Value (Data in Computer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_0$</td>
<td>R0</td>
<td>m.USA</td>
</tr>
<tr>
<td>$V_1$</td>
<td>R1</td>
<td>[m.SF, m.NYC, ...]</td>
</tr>
</tbody>
</table>

Expression is finished.

Computer
Key-Variable Memory: Reuse Intermediate Value

<table>
<thead>
<tr>
<th>Key (Embedding)</th>
<th>Variable (Symbol)</th>
<th>Value (Data in Computer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_0$</td>
<td>$R0$</td>
<td>m.USA</td>
</tr>
<tr>
<td>$V_1$</td>
<td>$R1$</td>
<td>[m.SF, m.NYC, ...]</td>
</tr>
</tbody>
</table>
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

Decoder Vocab

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_0$</td>
<td>R0</td>
</tr>
<tr>
<td>$V_1$</td>
<td>R1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$E_0$</td>
<td>Hop</td>
</tr>
<tr>
<td>$E_1$</td>
<td>Argmax</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$P_0$</td>
<td>CityIn</td>
</tr>
<tr>
<td>$P_1$</td>
<td>BornIn</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Variables: <10

Functions: <10

Predicates: 23K
Code Assistance: Syntactic Constraint

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

Decoder Vocab

- Variables: <10
- Functions: <10
- Predicates: 23K
# Code Assistance: Semantic Constraint

## Decoder Vocab

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_0$</td>
<td>$R0$</td>
</tr>
<tr>
<td>$V_1$</td>
<td>$R1$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$E_0$</td>
<td>$Hop$</td>
</tr>
<tr>
<td>$E_1$</td>
<td>$Argmax$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$P_0$</td>
<td>$CityIn$</td>
</tr>
<tr>
<td>$P_1$</td>
<td>$BornIn$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- **Variables**: <10
- **Functions**: <10
- **Predicates**: 23K

![Diagram](image.png)
Given definition of \( \text{Hop} \), need to output a predicate that is connected to \( \text{R}2 \) (\( m.\text{USA} \)).
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: SiblingsOf for ParentsOf.

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

1. Reduce variance at the cost of bias
2. Mix in approximate gold programs to bootstrap and stabilize training

- Top k in beam
- Approximate Gold Programs
- Policy gradient update
- Updated Model

Beam search

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**

<table>
<thead>
<tr>
<th>#Expressions</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Percentage</strong></td>
<td>0.4%</td>
<td>62.9%</td>
<td>29.8%</td>
<td>6.9%</td>
</tr>
<tr>
<td><strong>F1</strong></td>
<td>0.0</td>
<td>73.5</td>
<td>59.9</td>
<td>70.3</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Model</th>
<th>Avg. Prec. @1</th>
<th>Avg. Rec. @1</th>
<th>Avg. F1@1</th>
<th>Acc. @1</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>STAGG</em></td>
<td>67.3</td>
<td>73.1</td>
<td>66.8</td>
<td>58.8</td>
</tr>
<tr>
<td><em>NSM – our model</em></td>
<td>70.8</td>
<td>76.0</td>
<td>69.0</td>
<td>59.5</td>
</tr>
<tr>
<td><em>STAGG (full supervision)</em></td>
<td>70.9</td>
<td>80.3</td>
<td>71.7</td>
<td>63.9</td>
</tr>
</tbody>
</table>
Augmented REINFORCE

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

<table>
<thead>
<tr>
<th>Settings</th>
<th>Train Avg. F1@1</th>
<th>Valid Avg. F1@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>iterative ML only</td>
<td>68.6</td>
<td>60.1</td>
</tr>
<tr>
<td>REINFORCE only</td>
<td>55.1</td>
<td>47.8</td>
</tr>
<tr>
<td>Augmented REINFORCE</td>
<td>83.0</td>
<td>67.2</td>
</tr>
</tbody>
</table>
Thanks!