Neural Symbolic Machines: Learning Semantic Parsers on Freebase with Weak Supervision

Chen Liang\textsuperscript{1}, Jonathan Berant\textsuperscript{2}, Quoc Le, Kenneth Forbus, Ni Lao

\textsuperscript{1}Work done while the author was interning at Google
\textsuperscript{2}Work done while the author was a visiting scholar at Google

### Semantic Parsing

- **Natural language**
- **Logical form / Program**
- **Desired behavior / Answer**

### Question Answering with Freebase (WebQuestionsSP Dataset)

- **Large-scale Knowledge Base**
  - Properties of Hundreds of millions of entities
  - Plus relations among them
- **Previous state-of-the-art**
  - Staged Query Graph Generation (STAGG)
  - Who first voiced Meg on Family Guy\textsuperscript{3}?
  - \( \lambda x. \text{casted} \text{FamilyGuy}(y) \land \text{actor}(x,y) \land \text{character}(y, \text{MegGriffin}) \)

### MPC Framework

- **Neural Symbolic Machines with Key-Variable Memory**
- **Key Variable**
- **Execute**
- **Entity Resolver**
- **Non-differentiable Abstract Scalable Neural Computer Interface**

### Neural Computer Interface

- **Sampling v.s. Beam Search**
  - Programming is deterministic: closer to a maze than Atari game
  - Uses beam search (final beam with normalized probabilities) to generate training examples

### Iterative ML Training

- **Optimize log likelihood of approximate gold programs**
- \( J^{ML}(\theta) = \sum_q \log P(a^{\text{best}}_{0:T}(q)|q,\theta) \)
- **Fast, but suboptimal**
  - Spurious programs: wrong programs that happen to produce the correct answers, e.g.,
  - answering PlaceOfBirth with PlaceOfDeath
  - Lacking negative examples: hard to differentiate related relations, e.g., ParentOf, ChildrenOf, SiblingOf.

### Augmented REINFORCE

- **Optimize the expected F1 of generated programs**
- \( J^{RL}(\theta) = \sum_q R_{\theta} \left[ P(a_{0:T}|q,\theta) \right] \)
- **Problem**: slow and stuck at local optima
- **Augmentation**: add approximate gold program into final beam with a reasonably large probability

### Implementation

- **200 decoders, 50 KB servers, 1 trainer, 251 machines in total**
- **Since the network is small, we didn’t see much speedup from GPU**

### Experiments & Analysis

- **New state-of-the-art without manual engineering**

<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>STAGG</td>
<td>67.3</td>
<td>73.1</td>
<td>66.8</td>
<td>58.8</td>
</tr>
<tr>
<td>NSM – our model</td>
<td>70.8</td>
<td>76.0</td>
<td>69.0</td>
<td>59.5</td>
</tr>
<tr>
<td>STAGG (full supervision)</td>
<td>70.9</td>
<td>80.3</td>
<td>71.7</td>
<td>63.9</td>
</tr>
</tbody>
</table>

- **Comparison of iterative ML, REINFORCE and augmented REINFORCE**

<table>
<thead>
<tr>
<th>Settings</th>
<th>Train Avg. F1@1</th>
<th>Val Avg. F1@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>iterative ML only</td>
<td>66.6</td>
<td>47.8</td>
</tr>
<tr>
<td>REINFORCE only</td>
<td>55.1</td>
<td>47.8</td>
</tr>
<tr>
<td>Augmented REINFORCE</td>
<td>58.0</td>
<td>47.2</td>
</tr>
</tbody>
</table>

- **Techniques to reduce overfitting**

<table>
<thead>
<tr>
<th>Settings</th>
<th>( \Delta \text{ Avg. F1@1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Pretrained word embeddings</td>
<td>-5.5</td>
</tr>
<tr>
<td>- Pretrained relation embeddings</td>
<td>-2.7</td>
</tr>
<tr>
<td>- Dropout on GRU input and output</td>
<td>-2.4</td>
</tr>
<tr>
<td>- Dropout on softmax</td>
<td>-1.1</td>
</tr>
<tr>
<td>- Anonymize entity tokens</td>
<td>-2.0</td>
</tr>
</tbody>
</table>