Policy Shaping and Generalized Update Equations for Semantic Parsing from Denotations

Dipendra Misra*, Ming-Wei Chang†, Xiaodong He◊ and Wen-tau Yih‡

* Cornell University † Google AI Language
◊ JD AI Research ‡ Allen Institute for Artificial Intelligence
Semantic Parsing with Execution

Environment

Semantic Parsing

Meaning Representation

Execution

Denotation (Answer)
Semantic Parsing with Execution

Environment

“What nation scored the most points?”

Select Nation Where Points is Max

“England”

Semantic Parsing

Execution

<table>
<thead>
<tr>
<th>Index</th>
<th>Name</th>
<th>Nation</th>
<th>Points</th>
<th>Games</th>
<th>Pts/game</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Karen Andrew</td>
<td>England</td>
<td>44</td>
<td>5</td>
<td>8.8</td>
</tr>
<tr>
<td>2</td>
<td>Daniella Waterman</td>
<td>England</td>
<td>40</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Christelle Le Duff</td>
<td>France</td>
<td>33</td>
<td>5</td>
<td>6.6</td>
</tr>
<tr>
<td>4</td>
<td>Charlotte Barras</td>
<td>England</td>
<td>30</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>Naomi Thomas</td>
<td>Wales</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
Indirect Supervision

• No gold programs during training

<table>
<thead>
<tr>
<th>Index</th>
<th>Name</th>
<th>Nation</th>
<th>Points</th>
<th>Games</th>
<th>Pts/game</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Karen Andrew</td>
<td>England</td>
<td>44</td>
<td>5</td>
<td>8.8</td>
</tr>
<tr>
<td>2</td>
<td>Daniella Waterman</td>
<td>England</td>
<td>40</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Christelle Le Duff</td>
<td>France</td>
<td>33</td>
<td>5</td>
<td>6.6</td>
</tr>
<tr>
<td>4</td>
<td>Charlotte Barras</td>
<td>England</td>
<td>30</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>Naomi Thomas</td>
<td>Wales</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

“What nation scored the most points?”

“Select Nation Where Points is Max”

“England”
Learning

- **Neural Model**
  - $x$: “What nation scored the most points?”
  - $y$: Select Nation Where Index is Minimum
  - neural models $\rightarrow$ score($x, y$): encode $x$, encode $y$, and produce scores

- **Argmax procedure**
  - Beamsearch: argmax score($x, y$)

- **Indirect supervision**
  - Find approximated gold meaning representations
  - Reinforcement learning algorithms
Semantic Parsing with Indirect Supervision

• Question: “What nation scored the most points?”
• Answer: “England”

<table>
<thead>
<tr>
<th>Index</th>
<th>Name</th>
<th>Nation</th>
<th>Points</th>
<th>Games</th>
<th>Pts/game</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Karen Andrew</td>
<td>England</td>
<td>44</td>
<td>5</td>
<td>8.8</td>
</tr>
<tr>
<td>2</td>
<td>Daniella Waterman</td>
<td>England</td>
<td>40</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Christelle Le Duff</td>
<td>France</td>
<td>33</td>
<td>5</td>
<td>6.6</td>
</tr>
<tr>
<td>4</td>
<td>Charlotte Barras</td>
<td>England</td>
<td>30</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>Naomi Thomas</td>
<td>Wales</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Step 1: Search For Training

Select Nation Where Points = 44
Select Nation Where Index is Minimum
Select Nation Where Pts/game is Maximum
Select Nation Where Points is Maximum
Select Nation Where Name = Karen Andrew

Step 2: Update

Maximum Marginal Likelihood
Reinforcement Learning
Margin Methods
Search for *Training*

**Goal**

Find the correct program and high-scoring incorrect programs.

- A correct program should execute to the gold answer.
- In general, there are several spurious programs that execute to the gold answer but are semantically incorrect.

**Challenge**

Distinguish the correct program from spurious programs.
Search for Training: Spurious Programs

• Search for training. Goal: find semantically correct parse!
• Question: “What nation scored the most points?”

Select Nation Where Points = 44 ⇒ “England”
Select Nation Where Index is Minimum ⇒ “England”
Select Nation Where Pts/game is Maximum ⇒ “England”
Select Nation Where Point is Maximum ⇒ “England”

• All programs above generate right answers but only one is correct.
Update Step

**Goal**

Update the model using the programs found by search.

- Generally there are several methods to update the model.
- Examples: maximum marginal likelihood, reinforcement learning, margin methods.

**Challenge**

Find the right update strategy from various possibilities.
Contributions

- (1) **Policy Shaping** for handling spurious programs
- (2) **Generalized Update Equation** for generalizing common update strategies and allowing novel updates.

- (1) and (2) seem independent, but they interact with each other!!

- 5% absolute improvement over SOTA on SQA dataset
Learning from Indirect Supervision

- Question $x$, Table $t$, Answer $z$, Parameters $\theta$

1. [Search for Training] With $x$, $t$, $z$, beam search suitable $K = \{y'\}$

2. [Update] Update $\theta$, according $K = \{y'\}$
Spurious Programs

- Question $x$, Table $t$, Answer $z$, Parameters $\theta$

1. [Search for Training] With $x$, $t$, $z$, beam search suitable $\{y'\}$

- If the model selects a spurious program for update then it increases the chance of selecting spurious programs in future.
Policy Shaping [Griffith et al., NIPS-2013]

- Policy shaping is a way to incorporate prior knowledge.

- Formally, given a policy $p_\theta(y|x, t)$ and a critique policy $q(y|x, t)$ containing prior knowledge, we define

  $$p_s(y|x, t) \propto p_\theta(y|x, t) q(y|x, t)$$

  as our shaped policy.
Search with Shaped Policy

- Question $x$, Table $t$, Answer $z$, Parameters $\theta$

1. [Search for Training] With $x$, $t$, $z$, beam search suitable $\{y'\}$

- Perform beam search using the shaped policy score.

$$p_s(y|x,t) \propto p(y|x,t)q(y|x,t)$$
Critique Policy

- Contains prior knowledge to bias the model away from spurious programs.
- We consider the following simple critique policy:
  \[ q(y \mid x, t) \propto \exp\{\eta \times \text{critique}(y, x, t)\} \]
  where critique contains the following two scores:
  
  1. Surface-form Match: Features triggered for constants in the program that match a token in the question.
  
  2. Lexical Pair Score: Features triggered between keywords and tokens (e.g., Maximum and “most”).
Critique Policy Features

Question: “What nation scored the most points?”

Select Nation Where Points = 44
Select Nation Where Index is Minimum
Select Nation Where Pts/game is Maximum
Select Nation Where Points is Maximum
Select Nation Where Name = Karen Andrew
Learning Pipeline Revisited

1. [Search for Training] With $x, t, z$, beam search suitable $K = \{y'\}$

- Using policy shaping to find “better” $K$  
  ⇐ Shaping affects here

2. [Update] Update $\theta$, according $K = \{y'\}$

- What is the better objective function $J_\theta$?
Objective Functions Look Different!

- **Maximum Marginal Likelihood (MML)**
  \[
  J = \log p(z \mid x, t) = \log \sum_{y \in \mathcal{K}} p(z, y \mid x, t) = \log \sum_{y \in \mathcal{K}} p(z \mid y)p(y \mid x, t)
  \]

- **Reinforcement learning (RL)**
  \[
  J = \sum_{y \in \mathcal{K}} p(y \mid x, t)R(y, z)
  \]

- **Maximum Margin Reward (MMR)**
  \[
  J = -1\{|\mathcal{V}| > 0\}\{\text{score}(\bar{y}, x, t) - \text{score}(\hat{y}, x, t) + \delta(\hat{y}, \bar{y}, z)\}
  \]

  Maximum Reward Program

  Most violated program generated according to reward augment inference
Update Rules are Similar

- **Maximum Marginal Likelihood (MML)**

  \[
  \nabla J = \sum_{y \in \mathcal{Y}} \frac{p(z, y | x, t)}{\sum_{y'} p(z, y' | x, t)} \left\{ \nabla \text{score}(y, x, t) - \sum_{y' \in \mathcal{Y}} p(y' | x, t) \nabla \text{score}(y', x, t) \right\}
  \]

- **Reinforcement learning (RL)**

  \[
  \nabla J = 1 \left\{ \nabla \text{score}(y_{\text{samp}}, x, t) - \sum_{y' \in \mathcal{Y}} p(y' | x, t) \nabla \text{score}(y', x, t) \right\}
  \]

- **Maximum Margin Reward (MMR)**

  \[
  \nabla J = 1 \left\{ \nabla \text{score}(\hat{y}, x, t) - \sum_{y' \in \mathcal{Y}} 1[y' = \bar{y}] \nabla \text{score}(y', x, t) \right\}
  \]
Generalized Update Equation

$$\Delta = \sum_{y \in K} w(y, x, t) \left\{ \nabla \text{score}(y, x, t) - \sum_{y' \in K} q(y' | x, t) \nabla \text{score}(y', x, t) \right\}$$

Empirically determine $w$ and $q$.

2. [Update] Update $\theta$, according $K = \{y'\}$
Improvement over Margin Approaches

- **MMR**

\[
\nabla J = 1 \left\{ \nabla \text{score}(\hat{y}, x, t) - \sum_{y' \in \mathcal{K}} 1[y' = \tilde{y}] \nabla \text{score}(y', x, t) \right\}
\]

- **MAVER**

\[
\nabla J = 1 \left\{ \nabla \text{score}(\hat{y}, x, t) - \sum_{y' \in \mathcal{K}} \frac{1\{y' \in V\}}{|V|} \nabla \text{score}(y', x, t) \right\}
\]
• Policy shaping helps improve performance.
• With policy shaping, different updates matters even more
• Achieves new state-of-the-art (previously 44.7%) on SQA
Comparing Updates

**MML:** \( \nabla J = \sum_{y \in \mathcal{K}} \frac{p(z, y | x, t)}{\sum_{y'} p(z, y' | x, t)} \left\{ \nabla \text{score}(y, x, t) - \sum_{y' \in \mathcal{K}} p(y' | x, t) \nabla \text{score}(y', x, t) \right\} \)

**MMR:** \( \nabla J = 1 \left\{ \nabla \text{score}(\hat{y}, x, t) - \sum_{y' \in \mathcal{K}} 1[y' = \hat{y}] \nabla \text{score}(y', x, t) \right\} \)

- MMR and MAVER are more “aggressive” than MML
  - MMR and MAVER update towards to one program
  - MML updates toward to all programs that can generate the correct answer
Conclusion

- Discussed problem with search and update steps in semantic parsing from denotation.

- Introduced policy shaping for biasing the search away from spurious programs.

- Introduced generalized update equation that generalizes common update strategies and allows novel updates.

- Policy shaping allows more aggressive update!
BACKUP
Generalized Update as an Analysis Tool

\[ \Delta = \sum_{y \in \mathcal{K}} w(y, x, t) \left\{ \nabla \text{score}_\theta(y, x, t) - \sum_{y' \in \mathcal{K}} q(y' | x, t) \nabla \text{score}_\theta(y', x, t) \right\} \]

- MMR and MAVER are more “aggressive” than MML
  - MMR and MAVER only pick one
  - MML gives credits to all \{y\} that satisfies \{z\}
  - MMR and MAVER benefit more from shaping
Learning from Indirect Supervision

- Question $x$, Table $t$, Answer $z$, Parameters $\theta$

1. [Search for Training] With $x$, $t$, $z$, beam search suitable $\{y\}'$

- Search in training. Goal: finding semantically correct $y'$

2. [Update] Update $\theta$, according $\{y\}'$

- Many different ways of update $\theta$
Shaping and update

Better search ⇒ more aggressive update

1. [Search for Training] With $x, t, z$, beam search suitable $K = \{y'\}$

   - Using policy shaping to find “better” $K$ \(\Leftarrow\) Shaping affects here directly

2. [Update] Update $\theta$, according $K = \{y'\}$

   - What is the better objective function $J_\theta$? \(\Leftarrow\) Shaping affects here indirectly
Novel Learning Algorithm

<table>
<thead>
<tr>
<th>Intensity</th>
<th>Competing Distribution</th>
<th>Dev Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o shaping</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum Marginal Likelihood (MML)</td>
<td>Maximum Marginal Likelihood (MML)</td>
<td>32.4</td>
</tr>
<tr>
<td>Maximum Margin Reward (MMR)</td>
<td>Maximum Margin Reward (MMR)</td>
<td>40.7</td>
</tr>
<tr>
<td>Maximum Margin Reward (MMR)</td>
<td>Maximum Marginal Likelihood (MML)</td>
<td>41.9</td>
</tr>
</tbody>
</table>

- Mixing the MMR’s intensity and MML’s competing distribution gives an update that outperforms MMR.
Novel Learning Algorithms

- Novel update equations can be derived by changing $w$ and $q$.
- For example,

$$\Delta = \sum_{y \in \mathcal{K}} \frac{p(z, y|x, t)}{\sum_{y'} p(z, y'|x, t)} \left\{ \nabla \text{score}_\theta (y, x, t) - \sum_{y' \in \mathcal{K}} \frac{1\{y \in \mathcal{V}\}}{|\mathcal{V}|} \nabla \text{score}_\theta (y', x, t) \right\}$$

- Intensity of MML
- Competing distribution of MAVER

- Allows iterating over various updates (including standard ones) by treating them as parameters of a single equation.
Learning Method #1 –
Maximum Marginal Likelihood (MML)

- Given a set of programs $\mathcal{K}$ found by search, maximize the log marginal likelihood.

$$J = \log p(z|x, t) = \log \sum_{y \in \mathcal{K}} p(z, y|x, t) = \log \sum_{y \in \mathcal{K}} p(z|y)p(y|x, t)$$

where $p(y|x, t) \propto \exp\{\text{score}_\theta(y, x, t)\}$
$p(z|y) = 1$ if $y$ produces answer $z$, else 0
Learning Method #2 – Reinforcement Learning (RL)

- Given a set of programs $\mathcal{K}$ found by search and a reward function $R(\cdot, \cdot)$, maximize the expected reward.

$$J = \sum_{y \in \mathcal{K}} p(y|x, t)R(y, z)$$

- Policy Gradient: Gradient approximated by sampling a program $y_{samp}$ from $\mathcal{K}$
Learning Method #3 –
Maximum Margin Reward (MMR)

- Given a set of programs $\mathcal{K}$ found by search and a reward function $R(\cdot, \cdot)$, we define the violated set as:

$$\mathcal{V} = \{y | \text{score}(\hat{y}, x, t) < \text{score}(y, x, t) + \delta(\hat{y}, y', z); \ y \in \mathcal{K}\}$$

where $\hat{y}$ is a maximum reward program in $\mathcal{K}$, margin $\delta(\hat{y}, y, z) = R(\hat{y}, z) - R(y, z)$

- MMR minimizes the largest violation corresponding to $y'$

$$J = -\{\mathcal{V} \: \text{is not empty} \} \{\text{score}(y', x, t) - \text{score}(\hat{y}, x, t) + \delta(\hat{y}, y', z)\}$$
Learning Method #4 – Maximum Margin Average Violation Reward (MAVER)

- Minimizing only the most violation makes MMR less stable.

- Therefore, we consider a novel stable alternative that minimizes average violation.

\[ J = -\frac{1}{|V|} \sum_{y' \in V} \{ \text{score}(y', x, t) - \text{score}(\hat{y}, x, t) + \delta(\hat{y}, y', z) \} \]