Weakly Supervised Natural Language Understanding

Ni Lao
mosaix.ai
1.3.2019
Plan

- **Weak Supervision NLP**
  - NLP & software 2.0
  - Semantics as a foreign language
  - Semantic search
  - Semantic Parsing Tasks -- WebQuestionsSP, WikiTableQuestions

- **Neural Symbolic Machines (ACL 2017)**
  - Compositionality (short term memory)
  - Scalable KB inference (symbolism)

- **Memory Augmented Policy Optimization (NIPS 2018)**
  - Experience replay & optimal updating strategy
  - RL vs MML vs ML
Language understanding for AI and humanity

- Experts with different views of AI agree on the potential of NLP

If you got a billion dollars to spend on a huge research project that you get to lead, what would you like to do? -- r/CyberByte, 2015

NLP is fascinating, allowing us to focus on highly-structured inference problems, on issues that go to the core of "what is thought" but remain eminently practical, and on a technology that surely would make the world a better place. -- Michael I Jordan

What kind of impact you hope deep learning has on our future? -- Steve Paikin, 2016

I hope it allows Google to ... search by the content of the document rather than by the words in the document ... I hope it will make for intelligent personal systems, who can answer questions in a sensible way ... It will make computers much easier to use. Because you'll be able to just say to your computer "print this damn thing" -- Geoffrey Hinton
What is understanding?

“If they find a parrot who could answer to everything, I would claim it to be an intelligent being without hesitation.”,
-- Alan Turing, 1950

Does the machine literally “understand” Chinese? Or is it merely simulating the ability to understand Chinese?
-- John Searle, 1980

The Imitation Game

The Chinese Room Argument
Full Supervision NLP

- Traditionally NLP is a labor-intensive business

<table>
<thead>
<tr>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discourse Processing</td>
</tr>
<tr>
<td>Semantic Parsing</td>
</tr>
<tr>
<td>Syntactic analysis</td>
</tr>
<tr>
<td>Morphological analysis</td>
</tr>
</tbody>
</table>

NLP algorithms simply lookup

Rules written by 1000 PhDs
Software 2.0

1. specify some goal on the behavior
   ○ e.g., “satisfy input output pairs of examples”,
   ○ e.g., “win a game of Go”

2. write a rough skeleton of the code that identifies a subset of program space to search
   ○ e.g. a neural net architecture

3. use the computational resources at disposal to search this space for a program that works.

Death of feature engineering. (The) users of the software will (play) a direct role in building it. Data labeling is a central component to system design.
Where does knowledge come from?

- Can only cover the most popular semantics used by human
Weak Supervision NLP

- Avoid the knowledge acquisition bottleneck with machine learning
- Then we can cover more semantics used by human

end to end examples (e.g., QA pairs) → machine learning → intelligent systems with knowledge
Language & Reasoning

- The formalist view
  - Language was primarily invented for reasoning [Everaert+ 2015]

- The functionalist view
  - Language is for communication [Kirby 2017]

- Cognitive coupling hypothesis
  - Sequential processing is “necessary for behaviours such as primate tool use, navigation, foraging and social action.” [Kolodny & Edelman 2018]
Can this KG pattern be learned?

Reasoning is needed to understand text

[Liang+ 2017]

Bart's father is Homer
Semantics is a language for computation

“impressionist painters during the 1920s”

painters [/painting] !/art_forms

impressionist <visual_artist> x.[/associated_periods_or_movements = /impressionism]

<artist> during the 1920s x.[/date_of_work < 1930; /date_of_work > 1920]
Largest city in US?

Semantics is a language for computation

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

NYC

Freebase, DBpedia, YAGO, NELL
Semantics as a foreign language

1) **Natural languages** are programming languages to control human behavior (either others or self)

2) For machines and human to understand each other, they just need translation models trained with control theory
Internet as an external memory

- How information should be organized for scalability?

“AS WE MAY THINK”
(1945)

Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and to coin one at random, memex will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.
The scalability of modern search engines

- Can respond to user's requests within a fraction of a second
- But are weak at text understanding and complex reasoning
Can we search entities on the web?

- Multimedia, business, products have a lot of reviews and descriptions
Semantic Parsing

- Natural language queries or commands are converted to computation steps on data and produce the expected answers or behavior.

[Berant+ 2013]
[Liang 2013]
Related Works

- Training from full supervision is labor-intensive
  - DeepCoder [Balog, 2016]
  - NPI [Reed & Freitas, 2015]
  - Seq2Tree [Dong & Lapata, 2016]
  - ...

- Traditional semantic parsing models require feature engineering
  - SEMPRE [Berant et al, 2013]
  - STAGG [Yih et al, 2015]
  - ...

- End-to-end differentiable models cannot scale to large databases
  - Neural Programmer [Neekalantan et al, 2015]
  - Neural Turing Machines [Graves et al, 2014]
  - Neural GPU [Kaiser & Sutskever, 2015]
  - ...

- Combine deep learning, symbolic reasoning and reinforcement learning
  - Neural Symbolic Machines (Liang, Berant, Le, Forbus, Lao, 2017)
  - Memory Augmented Policy Optimization (MAPO) (Liang, Norouzi, Berant, Le, Lao, 2018)
**Question Answering with Knowledge Base**

**Largest city in US?**

- **GO**
  - (Hop V1 CityIn)
  - (Argmax V2 Population)
- **RETURN**

- **NYC**

**Paraphrase**

Many ways to ask the same question, e.g.,
- “What was the date that Minnesota became a state?”
- “When was the state Minnesota created?”

**Compositionality**

E.g., Freebase:
- 23K predicates,
- 82M entities,
- 417M triplets

**Large Search Space (Optimization)**

E.g., Freebase:
- 23K predicates,
- 82M entities,
- 417M triplets
WebQuestionsSP Dataset

- 5,810 questions from 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

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

- 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]
Brief Summary & Preview

DL leads to the death of feature engineering (but not domain knowledge)
DL makes it easier to leverage pre-trained unsupervised models

<table>
<thead>
<tr>
<th></th>
<th>SEMPRE/STAGG</th>
<th>This talk</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Paraphrase</strong></td>
<td>Co-occurrences collected from 1B docs (ClueWeb) and Freebase</td>
<td>Embeddings trained from 840B text tokens (GloVe)</td>
</tr>
<tr>
<td><strong>(semi-supervised)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Compositionality</strong></td>
<td>Relies on syntactic structure (text spans) and domain specific rules to constrain the generation of logic forms</td>
<td>A (LISP like) language specifies computations on KG</td>
</tr>
<tr>
<td><strong>of semantics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Search in the</strong></td>
<td>Priority queue &amp; heuristic rules</td>
<td>RL &amp; a program interpreter with syntactical &amp; semantic checks</td>
</tr>
<tr>
<td><strong>program space</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Modeling</strong></td>
<td>Expert designed features</td>
<td>Deep learning</td>
</tr>
</tbody>
</table>

Details
WikiTableQuestions: Dataset

**Breadth**
- No fixed schema: Tables at test time are not seen during training, needs to generalize based on column name.

**Depth**
- More compositional questions, thus require longer programs
- More operations like arithmetic operations and aggregation operations

<table>
<thead>
<tr>
<th>Year</th>
<th>City</th>
<th>Country</th>
<th>Nations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1896</td>
<td>Athens</td>
<td>Greece</td>
<td>14</td>
</tr>
<tr>
<td>1900</td>
<td>Paris</td>
<td>France</td>
<td>24</td>
</tr>
<tr>
<td>1904</td>
<td>St. Louis</td>
<td>USA</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>Athens</td>
<td>Greece</td>
<td>201</td>
</tr>
<tr>
<td>2008</td>
<td>Beijing</td>
<td>China</td>
<td>204</td>
</tr>
<tr>
<td>2012</td>
<td>London</td>
<td>UK</td>
<td>204</td>
</tr>
</tbody>
</table>

$x_1$: “Greece held its last Summer Olympics in which year?”
$y_1$: \{2004\}

$x_2$: “In which city’s the first time with at least 20 nations?”
$y_2$: \{Paris\}

$x_3$: “Which years have the most participating countries?”
$y_3$: \{2008, 2012\}

$x_4$: “How many events were in Athens, Greece?”
$y_4$: \{2\}

$x_5$: “How many more participants were there in 1900 than in the first year?”
$y_5$: \{10\}

[Pasupat & Liang, 2015]
## WikiTableQuestions: semantics

<table>
<thead>
<tr>
<th>Function</th>
<th>Arguments</th>
<th>Returns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>((\text{hop} \ v \ p))</td>
<td>(v): a list of rows. (p): a column.</td>
<td>(\text{a list of cells.})</td>
<td>Select the given column of the given rows.</td>
</tr>
<tr>
<td>((\text{argmax} \ v \ p)) (\text{(argmin} \ v \ p))</td>
<td>(v): a list of rows. (p): a number or date column.</td>
<td>(\text{a list of rows.})</td>
<td>From the given rows, select the ones with the largest / smallest value in the given column.</td>
</tr>
<tr>
<td>((\text{filter}<em>{&gt;} \ v \ q \ p)) (\text{(filter}</em>{=} \ v \ q \ p)) (\text{(filter}<em>{&lt;} \ v \ q \ p)) (\text{(filter}</em>{=} \ v \ q \ p)) (\text{(filter}<em>{=} \ v \ q \ p)) (\text{(filter}</em>{=} \ v \ q \ p))</td>
<td>(v): a list of rows. (q): a number or date. (p): a number or date column.</td>
<td>(\text{a list of rows.})</td>
<td>From the given rows, select the ones whose given column has certain order relation with the given value.</td>
</tr>
<tr>
<td>((\text{filter}<em>{\in} \ v \ q \ p)) (\text{(filter}</em>{!\in} \ v \ q \ p))</td>
<td>(v): a list of rows. (q): a string. (p): a string column.</td>
<td>(\text{a list of rows.})</td>
<td>From the given rows, select the ones whose given column contain / do not contain the given string.</td>
</tr>
</tbody>
</table>
## WikiTableQuestions: semantics

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(first v)</td>
<td>Select the first row of v. From the given rows, select the one with the smallest index.</td>
</tr>
<tr>
<td>(last v)</td>
<td>Select the last row of v. From the given rows, select the one with the largest index.</td>
</tr>
<tr>
<td>(previous v)</td>
<td>Select the row that is above the given row.</td>
</tr>
<tr>
<td>(next v)</td>
<td>Select the row that is below the given row.</td>
</tr>
<tr>
<td>(count v)</td>
<td>Count the number of rows in v.</td>
</tr>
<tr>
<td>(max v p)</td>
<td>Compute the maximum value in column p.</td>
</tr>
<tr>
<td>(min v p)</td>
<td>Compute the minimum value in column p.</td>
</tr>
<tr>
<td>(average v p)</td>
<td>Compute the average value in column p.</td>
</tr>
<tr>
<td>(sum v p)</td>
<td>Compute the sum value in column p.</td>
</tr>
<tr>
<td>(mode v p)</td>
<td>Get the most common value in column p.</td>
</tr>
<tr>
<td>(same_as v p)</td>
<td>Get the rows whose given column is the same as the given row.</td>
</tr>
<tr>
<td>(diff v0 v1 p)</td>
<td>Compute the difference in the given column of the given two rows.</td>
</tr>
</tbody>
</table>
### WikiTableQuestions: example solutions

<table>
<thead>
<tr>
<th>Superlative</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>nt-13901</strong>: the most points were scored by which player?</td>
<td>Sort all rows by column ‘points’ and take the first row. Output the value of column ‘player’ for the rows in v0.</td>
</tr>
</tbody>
</table>

| (argmax all_rows r.points-num) | (hop v0 r.player-str) |

<table>
<thead>
<tr>
<th>Difference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>nt-457</strong>: how many more passengers flew to los angeles than to saskatoon?</td>
<td>Find the row with ‘saskatoon’ matched in column ‘city’. Find the row with ‘los angeles’ matched in column ‘city’. Calculate the difference of the values in the column ‘passenger’ of v0 and v1.</td>
</tr>
</tbody>
</table>

| (filter \(\_in\) all_rows [ ‘saskatoon’] r.city-str) | (filter \(\_in\) all_rows [ ‘los angeles’] r.city-str) | (diff v1 v0 r.passengers-num) |
More examples

<table>
<thead>
<tr>
<th>Before / After</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>nt-10832: which nation is before peru?</strong></td>
</tr>
<tr>
<td>(filter_in all_rows ['peru'] r.nation-str)</td>
</tr>
<tr>
<td>(previous v0)</td>
</tr>
<tr>
<td>(hop v1 r.nation-str)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compare &amp; Count</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>nt-647: in how many games did sri lanka score at least 2 goals?</strong></td>
</tr>
<tr>
<td>(filter_\ge all_rows [2] r.score-num)</td>
</tr>
<tr>
<td>(count v0)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>nt-1133: other than william stuart price, which other businessman was born in tulsa?</strong></td>
</tr>
<tr>
<td>(filter_in all_rows ['tulsa'] r.hometown-str)</td>
</tr>
<tr>
<td>(filter_!in v0 ['william stuart price'] r.name-str)</td>
</tr>
<tr>
<td>(hop v1 r.name-str)</td>
</tr>
</tbody>
</table>
Plan

- **Weak Supervision NLP**
  - NLP & software 2.0
  - Semantics as a foreign language
  - Semantic search
  - Semantic Parsing Tasks -- WebQuestionsSP, WikiTableQuestions

- **Neural Symbolic Machines** (ACL 2017)
  - Compositionality (short term memory)
  - Scalable KB inference (symbolism)

- **Memory Augmented Policy Optimization** (NIPS 2018)
  - Experience replay & optimal updating strategy
  - RL vs MML vs ML
A bit background on Seq2Seq

- Separate a sequence model into encoder and decoder
- Improves a phrase-based SMT system by re-ranking top candidates
- Cannot perform well by itself due to the information bottleneck

[Hochreiter & Schmidhuber 1997]
[Sutskever, Vinyals, Le 2014]
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.
Computer with Domain Specific Languages

- **Lisp Interpreter**
  - Program => \( exp_1 \ exp_2 \ldots \ exp_n \) <END>
  - Exp => \((f \ arg_1 \ arg_2 \ldots \ arg_n)\)

- **What functions will be useful for the given task?**
  - 10 operations for WebQuestions
  - 22 different operations for WikiTableQuestions
    - hop
    - argmax, argmin
    - \(\text{filter}_=, \text{filter}_\neq, \text{filter}_>, \text{filter}_<, \text{filter}_>, \text{filter}_<=, \text{filter}_\leq, \text{filter}_\geq, \text{filter}_\geq\)
    - first, last, previous, next
    - max, min, average, sum, mode, diff, same

ACL [Liang+ 2017]
Code Assistance to Prune Search Space

ACL [Liang+ 2017]

Pen and paper

IDE

A lot of computations!
Key-Variable Memory for Semantic Compositionality

- Equivalent to a linearised bottom-up derivation of the recursive program
Augmented REINFORCE

- REINFORCE get stuck at local maxima
- Iterative ML training is not directly optimizing the F1 score
- Augmented REINFORCE obtains better 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>
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>
</tr>
</thead>
<tbody>
<tr>
<td>STAGG</td>
<td>67.3</td>
<td>73.1</td>
<td>66.8</td>
</tr>
<tr>
<td>NSM – our model</td>
<td>70.8</td>
<td>76.0</td>
<td>69.0</td>
</tr>
<tr>
<td>STAGG (full supervision)</td>
<td>70.9</td>
<td>80.3</td>
<td>71.7</td>
</tr>
</tbody>
</table>
Plan

- **Weak Supervision NLP**
  - NLP & software 2.0
  - Semantics as a foreign language
  - Semantic search
  - Semantic Parsing Tasks -- WebQuestionsSP, WikiTableQuestions

- **Neural Symbolic Machines** (ACL 2017)
  - Compositionality (short term memory)
  - Scalable KB inference (symbolism)

- **Memory Augmented Policy Optimization** (NIPS 2018)
  - Experience replay & optimal updating strategy
  - RL vs MML vs ML
What is RL?

Directly Optimizing The Expected Reward

- **ML** optimizes the log likelihood of target sequences

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

- **RL** optimizes the expected reward under a stochastic policy

\[ J^{RL}(\theta) = \sum_q \mathbb{E}_{P(a_{0:T}|q,\theta)}[R(q, a_{0:T})] \]

[Williams 1992]
[Sutton & Barto 1998]
RL models generate their own training data

- Training sample management issue
  - Too many low quality examples ⇒ slow training
  - Boosting high reward experience ⇒ biased training
Challenges of applying RL

- Large search space (sparse rewards)
  - Supervised pretraining (MLE)
  - Systematic exploration [Houthooft+ 2017]
  - Curiosity [Schmidhuber 1991][Pathak2017]

- Credit assignment (delayed reward)
  - Bootstrapping
    - E.g., AlphaGo uses a value function to estimate the future reward
  - Rollout n-steps

- Train speed & stability (optimization)
  - Trust region approaches (e.g., PPO)
  - Experience replay  Our focus today

Book [Sutton & Barto 1998]
NIPS [Abbeel & Schulman 2016]
Efficiency challenge

● RL is still far from data efficient
  ○ E.g. the best learning algorithm (DeepMind RainbowDQN) “passes median human performance on 57 Atari games at about 18 million frames (around 90 hours) of gameplay, while most humans can pick up a game within a few minutes.”

● How to improve its efficiency?
Applying RL to NLP

● Benefits of RL
  ○ Weak supervision (e.g., expected answer, user clicks)
  ○ Directly optimizing the metric (e.g., F1, accuracy, BLEU etc.)
  ○ Work with structured hidden variables (e.g., logical forms/programs)

● Challenges with existing solutions
  ○ Large search space sparse reward often leads to slow and unstable training
  ○ Spurious reward often lead to biased solutions
Memory Augmented Policy Optimization (MAPO)

Good return?

Memory

Learner

Actor

Systematic Exploration

Model checkpoint

NIPS [Liang+ 2018]
Most of the past experience are not helpful for improving the current model.
Augment REINFORCE with Memory

Linear combination of maximum likelihood objective and expected return:

$$
\lambda \log\text{-}\text{likelihood on a top-k buffer} + (1 - \lambda) \text{expected return}
$$

$$
\lambda \sum_{y \in \text{TopK}} \log p(y | x) + (1 - \lambda) \mathbb{E}_{\tilde{y} \sim p(y|x)} R(\tilde{y})
$$

- Not robust against spurious programs
- The composite objective is ad-hoc, and the gradient is biased

[Abolafia+ 2017]
[Liang+ 2016]
Spurious programs: right answer, wrong reason

Which nation won the most silver medal?

- **Correct program:**
  (argmax rows “Silver”)
  (hop v1 “Nation”)  

- **Many spurious programs:**
  (argmax rows “Gold”)
  (hop v1 “Nation”)
  
  (argmax rows “Bronze”)
  (hop v1 “Nation”)
  
  (argmin rows “Rank”)
  (hop v1 “Nation”)  
  ...

<table>
<thead>
<tr>
<th>Rank</th>
<th>Nation</th>
<th>Gold</th>
<th>Silver</th>
<th>Bronze</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Nigeria</td>
<td>14</td>
<td>12</td>
<td>9</td>
<td>35</td>
</tr>
<tr>
<td>2</td>
<td>Algeria</td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>Kenya</td>
<td>8</td>
<td>11</td>
<td>4</td>
<td>23</td>
</tr>
<tr>
<td>4</td>
<td>Ethiopia</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>Ghana</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>Ivory Coast</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>Egypt</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>Senegal</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>Morocco</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>Tunisia</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>Madagascar</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>Rwanda</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>Zimbabwe</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>Seychelles</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Combating spurious rewards

- **Reinforce** a rewarded experience only if the model (current policy) also thinks that it is the right thing to do

**Correctness of the behavior**

**Experiences**

**Model’s preference**

**Reinforce of the behavior**
Lower the gradient variance without introducing bias

Given a memory buffer of (sequence, return) pairs: \( \mathcal{B} \equiv \left\{ (y^{(i)}, r^{(i)}) \right\}_{i=1}^{n} \), re-express expected return as,

\[
p(\mathcal{B}) \underbrace{\mathbb{E}_{p(\tilde{y})|\tilde{y}\in \mathcal{B}} R(\tilde{y})}_{\text{inside the buffer}} + (1 - p(\mathcal{B})) \underbrace{\mathbb{E}_{p(\tilde{y})|\tilde{y}\not\in \mathcal{B}} R(\tilde{y})}_{\text{outside the buffer}}
\]

- **Importance sampling**
  - Sample more frequently inside the buffer
  - Rejection sampling for samples outside the buffer.
Optimal Sample Allocation

Given that we want to apply stratified sampling to estimate the gradient of REINFORCE with baseline under 1/0 rewards. It can be shown that the optimal strategy is to allocate the same number of samples to reward vs no reward experiences.
Comparison of model update strategies

Question 1
Correctness of the behavior
Experiences & Reward
Model’s preference
On-policy optimization (REINFORCE)
Iterative Maximum Likelihood (IML)
Maximum Marginal Likelihood (MML)
MAPO

Question 2

NIPS [Liang+ 2018]
Comparison

- REINFORCE does not work at all
- MAPO is slower but less biased

The shaded area represents the standard deviation of the dev accuracy

NIPS [Liang+ 2018]
SOTA results with weak supervision

<table>
<thead>
<tr>
<th>Model</th>
<th>E.S.</th>
<th>Dev.</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pasupat &amp; Liang (2015)[28]</td>
<td>-</td>
<td>37.0</td>
<td>37.1</td>
</tr>
<tr>
<td>Neelakantan et al. (2017)[26]</td>
<td>1</td>
<td>34.1</td>
<td>34.2</td>
</tr>
<tr>
<td>Neelakantan et al. (2017)[26]</td>
<td>15</td>
<td>37.5</td>
<td>37.7</td>
</tr>
<tr>
<td>Haug et al. (2017)[15]</td>
<td>1</td>
<td>-</td>
<td>34.8</td>
</tr>
<tr>
<td>Haug et al. (2017)[15]</td>
<td>15</td>
<td>-</td>
<td>38.7</td>
</tr>
<tr>
<td>Zhang et al. (2017)[51]</td>
<td>-</td>
<td>40.4</td>
<td>43.7</td>
</tr>
<tr>
<td>MAPO</td>
<td>1</td>
<td>42.7</td>
<td>43.8</td>
</tr>
<tr>
<td>MAPO (ensembled)</td>
<td>5</td>
<td>-</td>
<td>46.2</td>
</tr>
</tbody>
</table>

Table 3: Results on WikiTableQuestions. E.S. is the number of ensembles (if applicable).

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev.</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhong et al. (2017)[52]</td>
<td>60.8</td>
<td>59.4</td>
</tr>
<tr>
<td>Wang et al. (2017)[40]</td>
<td>67.1</td>
<td>66.8</td>
</tr>
<tr>
<td>Xu et al. (2017)[46]</td>
<td>69.8</td>
<td>68.0</td>
</tr>
<tr>
<td>Huang et al. (2018)[18]</td>
<td>68.3</td>
<td>68.0</td>
</tr>
<tr>
<td>Yu et al. (2018)[48]</td>
<td>74.5</td>
<td>73.5</td>
</tr>
<tr>
<td>Sun et al. (2018)[38]</td>
<td>75.1</td>
<td>74.6</td>
</tr>
<tr>
<td>Dong &amp; Lapata (2018)[12]</td>
<td>79.0</td>
<td>78.5</td>
</tr>
<tr>
<td>MAPO</td>
<td>72.4</td>
<td>72.6</td>
</tr>
<tr>
<td>MAPO (ensemble of 5)</td>
<td>-</td>
<td>74.9</td>
</tr>
</tbody>
</table>

Table 4: Results on WikiSQL. *All other methods use question-program pairs as strong supervision, while MAPO only uses question-answer pairs as weak supervision.
Scale up: Distributed Actor-Learner architecture

[Scale up: Distributed Actor-Learner architecture diagram]

[Liang et al, 2017; Espeholt et al, 2018; Liang et al, 2018]
Thanks!

- **Weak Supervision NLP**
  - NLP & software 2.0
  - Semantics as a foreign language
  - Semantic search
  - Semantic Parsing Tasks -- WebQuestionsSP, WikiTableQuestions

- **Neural Symbolic Machines** (ACL 2017)
  - Compositionality (short term memory)
  - Scalable KB inference (symbolism)

- **Memory Augmented Policy Optimization** (NIPS 2018)
  - Experience replay & optimal updating strategy
  - RL vs MML vs ML