Adversarial Learning for Neural Dialogue Generation

Li, Jiwei, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky

EMNLP’17

Presented by Yiren Wang (CS546, Spring 2018)
Main Contributions

• Goal
  • End-to-end neural dialogue generation system
  • To produce sequences that are indistinguishable from human-generated dialogue utterances

• Main Contributions
  • Adversarial training approach for response generation
  • Cast the task in a reinforcement learning framework.
Outline

• Model Architecture
• Adversarial Reinforce Learning:
  • Adversarial REINFORCE
  • Reward for Every Generation Step (REGS)
  • Teacher Enforcing
  • Overall Algorithm (Pseudocode)
• Experiment Results
• Summary
Adversarial Model

• Overall Architecture
Generative Model

- Model: Standard Seq2Seq model with Attention Mechanism
- Input: dialogue history $x$
- Output: response $y$

$$\text{Loss} = - \log p(\text{target}|\text{source})$$

(Sutskever et al., 2014; Jean et al., 2014)
Discriminative Model

- Model: binary classifier
  - Hierarchical encoder + 2-class softmax
- Input: dialogue utterances \{x, y\}
- Output: label indicating whether generated by human or by machine
  - \(Q_+({x, y})\) (by human)
  - \(Q_-({x, y})\) (by machine)
Adversarial REINFORCE

• **Policy Gradient Training**
• Discriminator score is used as reward for generator
• Generator is trained to maximize the expected reward

\[ J(\theta) = E_{y \sim p(y|x;\theta)} (Q_+({x, y})) \]
Policy Gradient Training

\[ J(\theta) = \mathbb{E}_{y \sim p(y|x; \theta)} (Q_+(\{x, y\})) \]

Approximated by likelihood ratio

\[ \nabla J(\theta) \approx [Q_+(\{x, y\}) - b(\{x, y\})] \]

\[ \nabla \log \pi(y|x) \]

\[ = [Q_+(\{x, y\}) - b(\{x, y\})] \]

\[ \nabla \sum_t \log p(y_t|x, y_{1:t-1}) \]

By human \( Q_+(\{x, y\}) \) (reward)
Policy Gradient Training

\[ J(\theta) = E_{y \sim p(y|x;\theta)}(Q_+(\{x, y\})) \]

Approximated by likelihood ratio

\[ \nabla J(\theta) \approx [Q_+(\{x, y\}) - b(\{x, y\})] \]

\[ = \nabla \log \pi(y|x) \]

\[ = [Q_+(\{x, y\}) - b(\{x, y\})] \]

\[ \nabla \sum_{t} \log p(y_t|x, y_{1:t-1}) \]

Baseline value to reduce the variance of the estimate while keeping it unbiased

Policy updates in the parameter space

Human Dialogues \rightarrow Discriminator

\{x, y\}

By human \( Q_+(\{x, y'\}) \)

(reward)

Generator

classification score
Problem with vanilla REINFORCE

• Expectation of reward is approximated by only one sample
• Reward associated with the sample is used for all actions

\[ Q_+({\{x, y\}}) - b({\{x, y\}}) \]

Input : What’s your name
Human : I am John
Machine : I don’t know (negative reward)
Problem with vanilla REINFORCE

• Expectation of reward is approximated by only one sample
• Reward associated with the sample is used for all actions

Input: What’s your name
Human: I am John

Machine: I don’t know (negative reward)

Machine: I don’t know (neutral reward) (negative reward)
Reward for Every Generation Step (REGS)

• **Strategies**
  • Monte Carlo (MC) Search
  • Training Discriminator For Rewarding Partially Decoded Sequences
Strategy I: Monte Carlo (MC) Search

- Repeats sampling N times
- Average score is the reward
Strategy I: Monte Carlo (MC) Search

• Repeats sampling $N$ times
• Average score is the reward
Strategy I: Monte Carlo (MC) Search

• Repeats sampling N times
• Average score is the reward

√ More accurate
× Time consuming

Average reward
Strategy II: Reward Partially Decoded Sequences

- Break generated sequences into partial subsequences
- Sample one positive and one negative subsequence

\[ \nabla J(\theta) \approx \left[ Q_+(\{x, y\}) - b(\{x, y\}) \right] \]
\[ \nabla \log \pi(y|x) = [Q_+(\{x, y\}) - b(\{x, y\})] \]
\[ \nabla \sum_t \log p(y_t|x, y_{1:t-1}) \]

- Time efficient
- Less accurate
Unstable Training

Generator only indirectly exposed to the gold-standard target

• When generator deteriorates:
  • Discriminator does an excellent job distinguishing – from +
  • Generator only knows generated sequences are bad
  • But get lost what are good and how to push itself towards good
  • Loss of reward signals leads to a breakdown in training
Teacher Forcing

• Teacher Forcing:
  "having a teacher intervene and force it to generate true responses"

• Discriminator:
  • assigns a reward of 1 to the human responses

• Generator:
  • uses this reward to update itself on human generated examples

√ more direct access to the gold-standard targets
Overall Algorithm

For number of training iterations do
  For i=1,D-steps do
    Sample (X,Y) from real data
    Sample $\hat{Y} \sim G(\cdot|X)$
    Update $D$ using $(X,Y)$ as positive examples and $(X,\hat{Y})$ as negative examples.
  End
  
  For i=1,G-steps do
    Sample (X,Y) from real data
    Sample $\hat{Y} \sim G(\cdot|X)$
    Compute Reward $r$ for $(X,\hat{Y})$ using $D$.
    Update $G$ on $(X,\hat{Y})$ using reward $r$
    Teacher-Forcing: Update $G$ on $(X,Y)$
  End
End
<table>
<thead>
<tr>
<th>Input</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vanilla-MLE</td>
<td>i'm not a doctor.</td>
</tr>
<tr>
<td>Vanilla-Sample</td>
<td>well everything you did was totally untrue.</td>
</tr>
<tr>
<td>REINFORCE</td>
<td>i don't know how long it's been.</td>
</tr>
<tr>
<td>REGS Monte Carlo</td>
<td>A few months, I guess.</td>
</tr>
<tr>
<td></td>
<td>so i had the doctors test sammy's response to conditioning.</td>
</tr>
<tr>
<td>Vanilla-MLE</td>
<td>sammy wrote the test sammy wrote the test.</td>
</tr>
<tr>
<td>Vanilla-Sample</td>
<td>objects pick objects up objects objects objects objects objects objects</td>
</tr>
<tr>
<td>REINFORCE</td>
<td>i'm not sure that's relevant.</td>
</tr>
<tr>
<td>REGS Monte Carlo</td>
<td>so he took the pills.</td>
</tr>
<tr>
<td></td>
<td>what are you going to do?</td>
</tr>
<tr>
<td>Vanilla-MLE</td>
<td>i'm going to the movies.</td>
</tr>
<tr>
<td>Vanilla-Sample</td>
<td>get him outta here first!</td>
</tr>
<tr>
<td>REINFORCE</td>
<td>i'm going to get you.</td>
</tr>
<tr>
<td>REGS Monte Carlo</td>
<td>i'm going to get you a new car.</td>
</tr>
<tr>
<td></td>
<td>they fear your power your intellect.</td>
</tr>
<tr>
<td>Vanilla-MLE</td>
<td>you're the only one who knows what's going on.</td>
</tr>
<tr>
<td>Vanilla-Sample</td>
<td>when they are conquered and you surrender they will control all of us.</td>
</tr>
<tr>
<td>REINFORCE</td>
<td>i'm afraid i'm not ready yet.</td>
</tr>
<tr>
<td>REGS Monte Carlo</td>
<td>i'm not afraid of your power.</td>
</tr>
</tbody>
</table>

Table 1: Sampled responses from different models. More in Appendix Tables 5 and 6.
Summary

• Adversarial training for response generation
• Cast the model in the framework of reinforcement learning
  • Discriminator: Turing test
  • Generator: trained to maximize the reward from discriminator
Thanks!