Adversarial Learning for Neural Dialogue Generation

Li et al. [1]

Presenter : Shivank

- Jiwei Li, Will Monroe, Tianlin Shi, Dan Jurafsky - Stanford
- Sebastien Jean - NYU
- Alan Ritter - Ohio State

Table of contents

1. Introduction

   Problem Statement

   GAN Architecture

2. Adversarial Training

   Adversarial REINFORCE

   Policy Gradient Training

   REGS

   Teacher Forcing

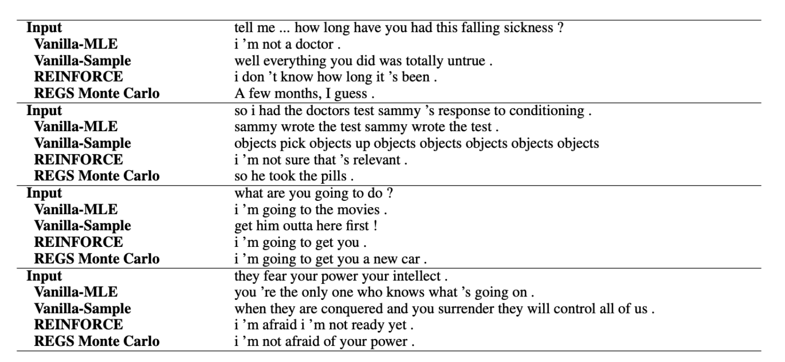
3. Adversarial Evaluation

4. Discussion Questions

Introduction

Problem Statement

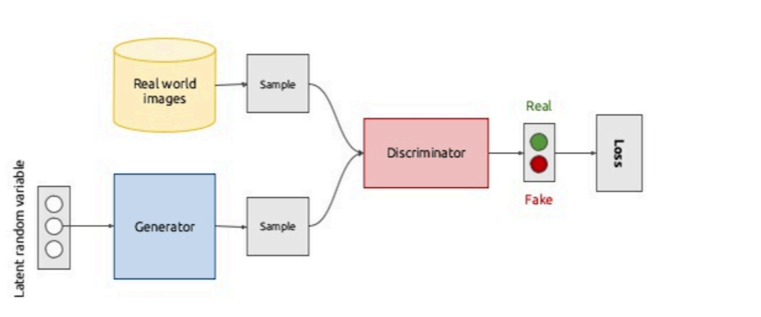
- Generating human-like responses given a human-to-human dialogue history
- Given a dialogue history x, generate response sequence $y = \{y_1, y_2, ..., y_T\}$



Input	tell me ... how long have you had this falling sickness ?
Vanilla-MLE	i ’m not a doctor .
Vanilla-Sample	well everything you did was totally untrue .
REINFORCE	i don ’t know how long it ’s been .
REGS Monte Carlo	A few months, I guess .
Input	so i had the doctors test sammy ’s response to conditioning .
Vanilla-MLE	sammy wrote the test sammy wrote the test .
Vanilla-Sample	objects pick objects up objects objects objects objects objects
REINFORCE	i ’m not sure that ’s relevant .
REGS Monte Carlo	so he took the pills .
Input	what are you going to do ?
Vanilla-MLE	i ’m going to the movies .
Vanilla-Sample	get him outta here first !
REINFORCE	i ’m going to get you .
REGS Monte Carlo	i ’m going to get you a new car .
Input	they fear your power your intellect .
Vanilla-MLE	you ’re the only one who knows what ’s going on .
Vanilla-Sample	when they are conquered and you surrender they will control all of us .
REINFORCE	i ’m afraid i ’m not ready yet .
REGS Monte Carlo	i ’m not afraid of your power .

GAN Architecture

- A generator model to produce response sequences
- A discriminator analogous to the evaluator in the Turing test
- Outputs from discriminator are used as rewards for the generative model



Adversarial Training

Adversarial REINFORCE

- Generative Model
  - The G defines a policy to generate y, given x, and uses a Seq2Seq RNN model
- Discriminative Model
  - Binary Classifier - softmax with two classes
  - Input - vector encoding (using hierarchical encoder) of the dialogue history and response: {x|y}
  - Outputs probability of the input dialogue being human generated : $Q_+(x, y)$, or machine generated : $Q_-(x, y)$

Policy Graident Training

- Probability of current utterances being human-generated, i.e., $Q_+(x, y)$, is used as a reward for the generator.
- Maximize the expected reward of generated utterance, $J(\theta) = \mathbb{E}_{y \sim p(y|x)}(Q_+(x, y)|\theta)$
- The gradient is calculated using the likelihood-ratio trick, i.e., $\nabla_\theta log(p(x|\theta) = \frac{\nabla_\theta(p(x|\theta)}{p(x|\theta)}$.

$$\nabla J(\theta) = [Q_+(x, y) - b(x, y)]\nabla log(\pi(y|x))$$

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

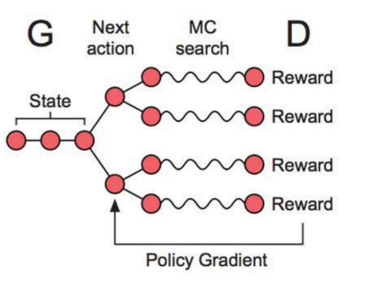
- π denotes the probability of generating the response $\{y|x\}$
- $b(x, y)$ denotes the baselines value to reduce the variance of the estimate while keeping it unbiased.

Reward for every generating step (REGS)

- Reward generated for all the actions, i.e., all the tokens generated in a sequence should not be same.
- For example:
    - What’s your name?
    - ’I am John’ - Correct Response
    - ’I don’t know’ - Reward for ’I’ is neutral, ’don’t know’ is negative

Monte Carlo Search

- Given a partially decoded sequence, s_p, sample a full sequence N times
- The N generated sequences share a common prefix, s_p
- The N sequences are passed through discriminator and average reward is used as the reward for s_p
- **Cons** : Significantly Time Consuming



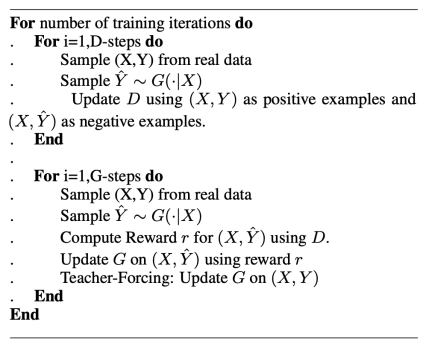
Rewards for partially decoded sequences

- Train a discriminator that assign rewards to both fully and partially decoded sequences.
- Break the generated sequence into partial sequences, namely $\{y^+_{1:t}\}$ and $\{y^-_{1:t}\}$.
- Radnomly sample one example from $\{y^+_{1:t}\}$ and one example from $\{y^-_{1:t}\}$, which are used to train the discriminator.

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

Teacher Forcing

- Updating generator leads to instability because generator is exposed to the gold-standard targets only indirectly.
- Also update the generator on the human generated response.



```
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
```

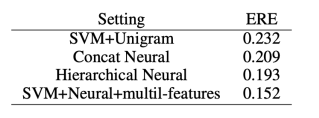
Adversarial Evaluation

Adversarial Evaluation & Success

- Akin to testing the ability of human evaluator.
- Given a sentence label it as human-generated or not.
- Adversarial evaluation consists of training on positive and negative examples, and testing on held-out dataset
- AdverSuc - Fraction of instances where a model can fool the evaluator during test time.
- Compare AdverSuc of a discriminator with gold standard AdverSuc.
    - Human generated (+), Human generated (-), AdverSuc : 0.5
    - Machine generated (+), Machine generated (-), AdverSuc : 0.5
    - Human generated (+), Random utterance (-), AdverSuc : 0
    - Human generated (+), Random + True responses (-), AdverSuc : 0
- Evaluator Relaibility Error (ERE) : average deviation of Adversuc from the gold standard AdverSuc

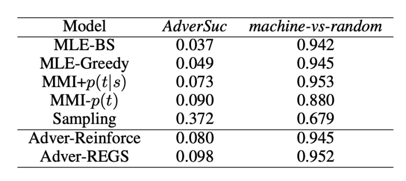
Machine-vs-Random Accuracy

- Accuracy of distinguishing between machine-generated responses and randomly sampled responses
- Table 1 shows ERE value for different classification architectures



Setting	ERE
SVM+Unigram	0.232
Concat Neural	0.209
Hierarchical Neural	0.193
SVM+Neural+multil-features	0.152

Figure 1: Table 1



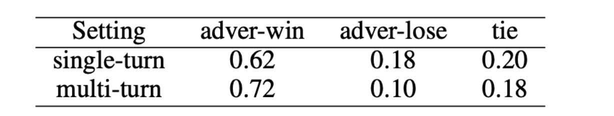
Model	*AdverSuc*	*machine-vs-random*
MLE-BS	0.037	0.942
MLE-Greedy	0.049	0.945
MMI+$p(t\|s)$	0.073	0.953
MMI-$p(t)$	0.090	0.880
Sampling	0.372	0.679
Adver-Reinforce	0.080	0.945
Adver-REGS	0.098	0.952

Figure 2: Table 2

- Table 2 shows Seq2Seq models which uses different decoding techniques for predicting the target sequence, and the performance of *Hierarchical Neural* evaluator on outputs of these models.
- Sampling attains a high AdverSuc but low machine-vs-random accuracy - difficult to distinguish from human as well as random

Human Evaluation

- Employing crowd-sourced judges to evaluate a random sample of 200 items.
- Input message, the generated output and real output for single turn and multi-turn (3 turns) conversations
- Ties were allowed



Setting	adver-win	adver-lose	tie
single-turn	0.62	0.18	0.20
multi-turn	0.72	0.10	0.18

Discussion Questions

Question 1

In the Adversarial REINFORCE section, authors mention that, b(x,y) denotes the baseline value, which is helpful in reducing the variance. It is unclear how this baseline is calculated. What does the baseline signify qualitatively? How do you think the results would change if we don't use a baseline value?

- .

Question 2

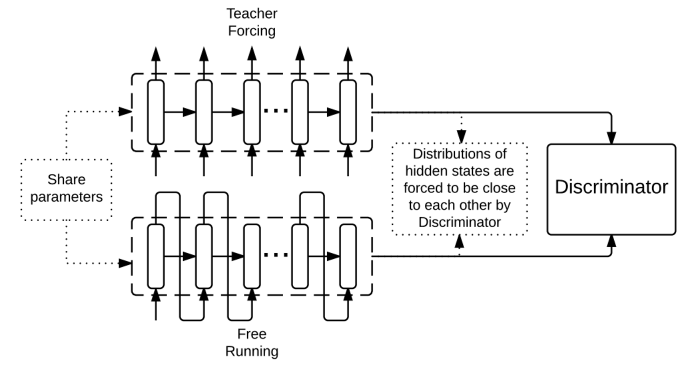
For the REGS, the discriminator becomes less accurate after partially decoded sequences are added in the training examples. What can be other approaches apart from ’MC’ and ’training on partially decoded sequences’, that may be more accurate as well as time-effective?

- .

Question 3

What can be the other strategies apart from Teacher Forcing that can be used to achieve stability in training of the generative model? Are there any new strategies that have popped up in the literature?

- Sai - Professor forcing, curriculum learning
- Robert - GPT-2 to initialize generator, BERT for discriminator


Teacher Forcing
Share parameters
Distributions of hidden states are forced to be close to each other by Discriminator
Discriminator
Free Running

Question 4

The (ERE), i.e., the Evaluator Reliability Error, is the average deviation of AdverSuc from the gold standard AdverSuc, on the four tasks mentioned in the paper. Do you think they should be equally weighted? Is there any other task which can be used to judge the reliability of the evaluator.

- Jiajing, Katherine, June Cho - Task 3 and 4 look more difficult; interesting to know effect of different weighing schemes
- Jialu, Charlie, Varsha - Equal weights are justified
- Katherine - Using machine vs machine responses; *how much the evaluator is relying on coherence versus other attributes.*
- Varsha - Using random + machine generated as negative examples
- Jialu - Randomly truncate some words from human responses to use as negative examples
- Tianxing - Good performance on task 4, automatically ensures that for task 1

Question 5

Adversarial training strategy did not observe a clear improvement in performance on the machine translation task. Why? Any recent work, which has improved performance on machine translation using adversarial training?

- Himank - “Improving NMT with Conditional Sequence GANs”, generation is biased towards high BLEU scores.
- Heather - GANS doesn’t work well in the machine translation because the search space is so huge that sampled translations are not sufficient for discriminator training.
- Frank - NLP space is not continuous
- Jialu - If we use BLEU score, I think the entropy of the target sequences will not influence the result that much.
- Ziwei - Big discrepancy between the distributions of the generated sequences and target sequences, implies easier to produce human like responses.

References

[1] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.