Reinforcement Learning for NLP

Caiming Xiong

Salesforce Research

CS224N/Ling284

Outline

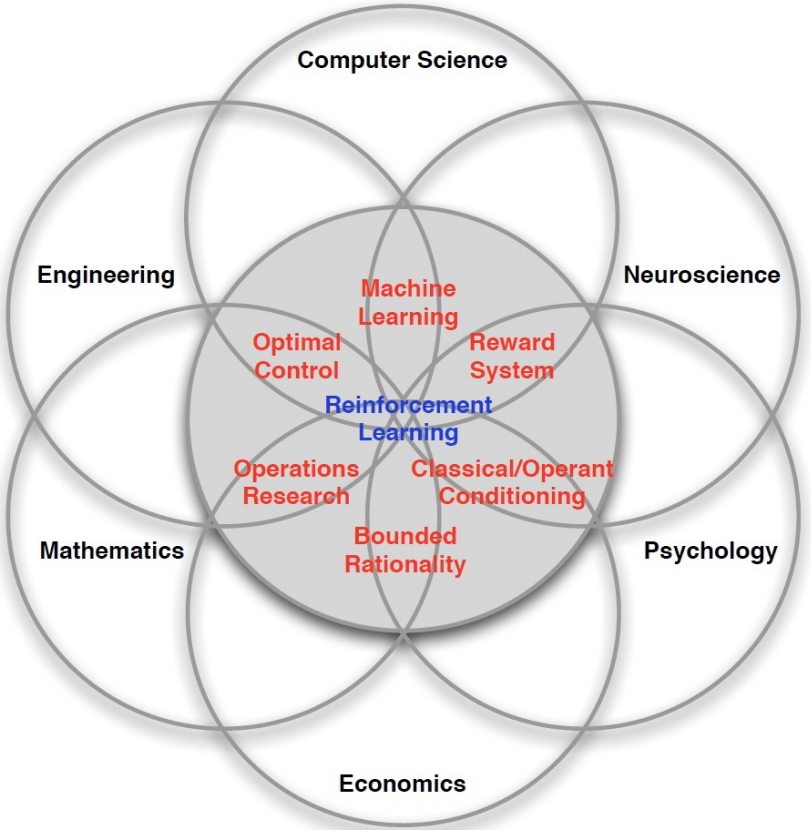
Introduction to Reinforcement Learning

Policy-based Deep RL

Value-based Deep RL

Examples of RL for NLP

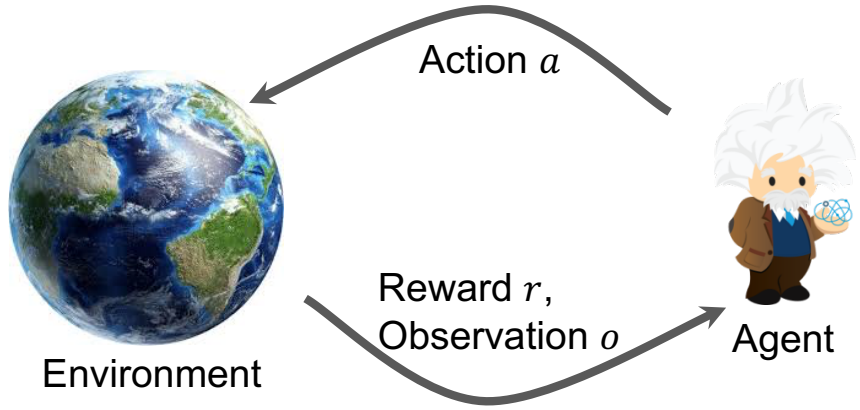
Many Faces of RL



By David Silver

What is RL?

- RL is a general-purpose framework for sequential decision-making
- Usually describe as agent interacting with unknown environment
- Goal: select action to maximize a future cumulative reward



Motor Control

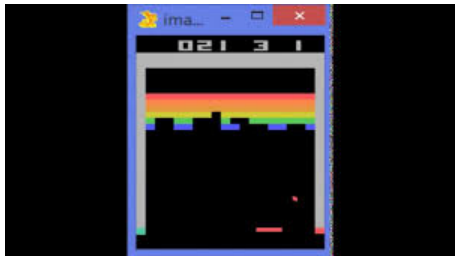
- Observations: images from camera, joint angle
- Actions: joint torques
- Rewards: navigate to target location, serve and protect humans

Business Management

- Observations: current inventory levels and sales history
- Actions: number of units of each product to purchase
- Rewards: future profit

Similarly, there also are resource allocation and routing problems ….

Games



State

- Experience is a sequence of observations, actions, rewards

- The state is a summary of experience

$$s_t = f(o_1, r_1, a_1, \ldots, a_{t-1}, o_t, r_t)$$

RL Agent

Major components:

- Policy: agent’s behavior function
- Value function: how good would be each state and/or action
- Model: agent’s prediction/representation of the environment

Policy

A function that maps from state to action:

- Deterministic policy:

$$a = \pi(s)$$

- Stochastic policy:

$$\pi(a|s) = \mathbb{P}[a|s]$$

$BQ^{\pi}(s,a) =$

Value Function

- Q-value function gives expected future total reward
  - from state and action (s, a)
  - under policy π
  - with discount factor $\gamma \in (0,1)$

$$Q^{\pi}(s,a) = \mathbb{E}\left[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots \mid s, a\right]$$

  - Show how good current policy

- Value functions can be defined using Bellman equation

$$Q^{\pi}(s,a) = \mathbb{E}_{s',a'}\left[r + \gamma Q^{\pi}(s',a') \mid s, a\right]$$

- Bellman backup operator

$$\mathrm{B}^{\pi}\mathrm{Q}(\mathrm{s},\mathrm{a}) = \mathrm{E}_{\mathrm{s}',\mathrm{a}'}[\mathrm{r} + \gamma Q^{\pi}(s',a') | s, a] \qquad Q,\ B^{\pi}Q,\ (B^{\pi})^2 Q,\ (B^{\pi})^3 Q,\ \cdots \rightarrow Q^{\pi}$$

Value Function

- For optimal Q-value function $Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)$, then policy function is deterministic, the Bellman equation becomes:

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q^*(s',a') \mid s,a\right]$$

$$\mathrm{B}^{\pi}\mathrm{Q}(\mathrm{s},\mathrm{a}) = \mathrm{E}_{s'}[\mathrm{r} + \gamma \max_{a'} Q^{\pi}(s',a')|s,a]$$

$$Q,\ BQ,\ B^2Q,\ \cdots \rightarrow Q^*$$

What is Deep RL?

- Use deep neural network to approximate
  - Policy
  - Value function
  - Model
- Optimized by SGD

Approaches

- **Policy-based Deep RL**
- **Value-based Deep RL**
- Model-based Deep RL

Deep Policy Network

- Represent policy by deep neural network that

$$\max_{\theta} E_{a\sim p(a|\theta,s)}[r(a)|\theta, s]$$

- Ideas: given a bunch of trajectories,
  - Make the good trajectories/action more probable
  - Push the actions towards good actions

Policy Gradient

How to make high-reward actions more likely:

$$
\begin{aligned}
\nabla_\theta \mathbb{E}_a[r(a)] &= \nabla_\theta \int da\ p(a|\theta, s) r(a) \\
&= \int da\ \nabla_\theta p(a|\theta, s) r(a) \\
&= \int da\ p(a|\theta, s) \frac{\nabla_\theta p(a|\theta, s)}{p(a|\theta, s)} r(a) \\
&= \int da\ p(a|\theta, s) \nabla_\theta \log p(a|\theta, s) r(a) \\
&= \mathbb{E}_a[\nabla_\theta \log p(a|\theta, s) r(a)]
\end{aligned}
$$

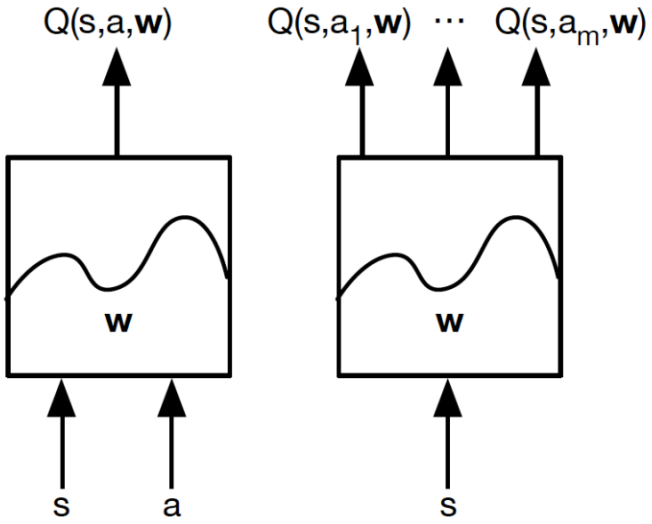
$$\hat{g} = r(a)\nabla_\theta \log p(a|\theta, s)$$

- Let's $r(a)$ say that measures how good the sample is.
- Moving in the direction of gradient pushes up the probability of the sample, in proportion to how good it is.

Deep Q-Learning

- Represent value function by Q-network

$$Q(s, a, \mathbf{w}) \approx Q^*(s, a)$$



Deep Q-Learning

- Optimal Q-values should obey Bellman equation

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q(s',a')^* \mid s,a\right]$$

- Treat right-hand side as target network, given (s,a,r,s'), optimize MSE loss via SGD:

$$I = \left(r + \gamma \max_{a} Q(s',a',\mathbf{w}) - Q(s,a,\mathbf{w})\right)^2$$

- Converges to optimal Q using table lookup representation

Deep Q-Learning

But diverges using neural networks due to:

- Correlations between samples
- Non-stationary targets

Deep Q-Learning

Experience Replay: remove correlations, build data-set from agent's own experience

s_1, a_1, r_2, s_2
s_2, a_2, r_3, s_3
s_3, a_3, r_4, s_4
...
$s_t, a_t, r_{t+1}, s_{t+1}$

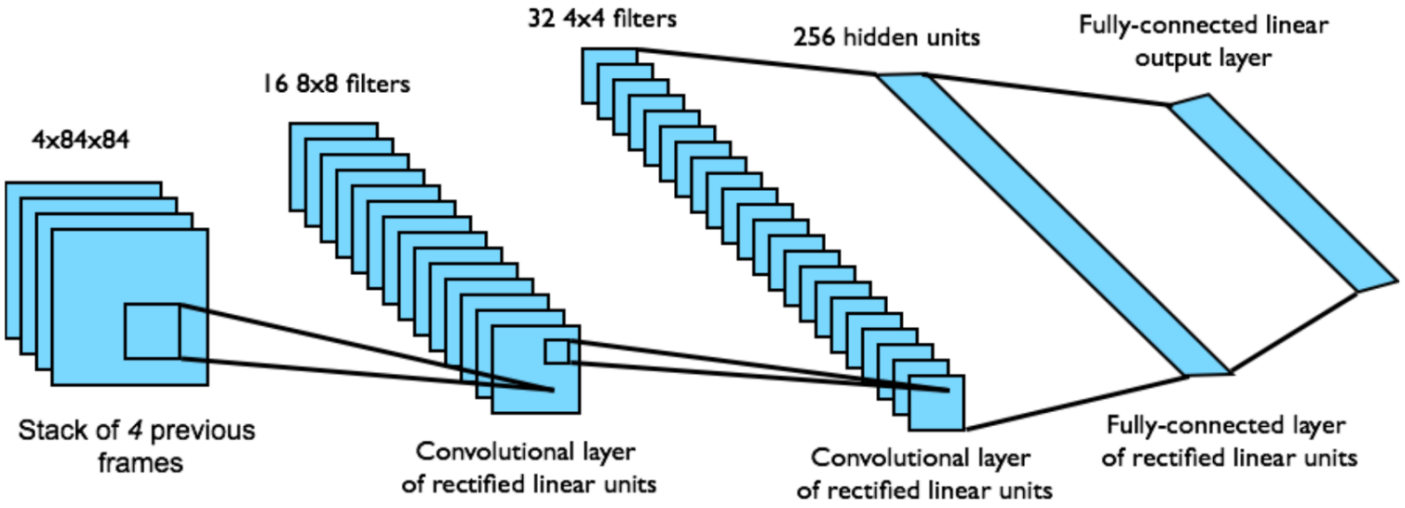
$\rightarrow \quad s, a, r, s'$

- Sample experiences from data-set and apply update

$$I = \left(r + \gamma \max_{a'} Q(s', a', \mathbf{w}^-) - Q(s, a, \mathbf{w}) \right)^2$$

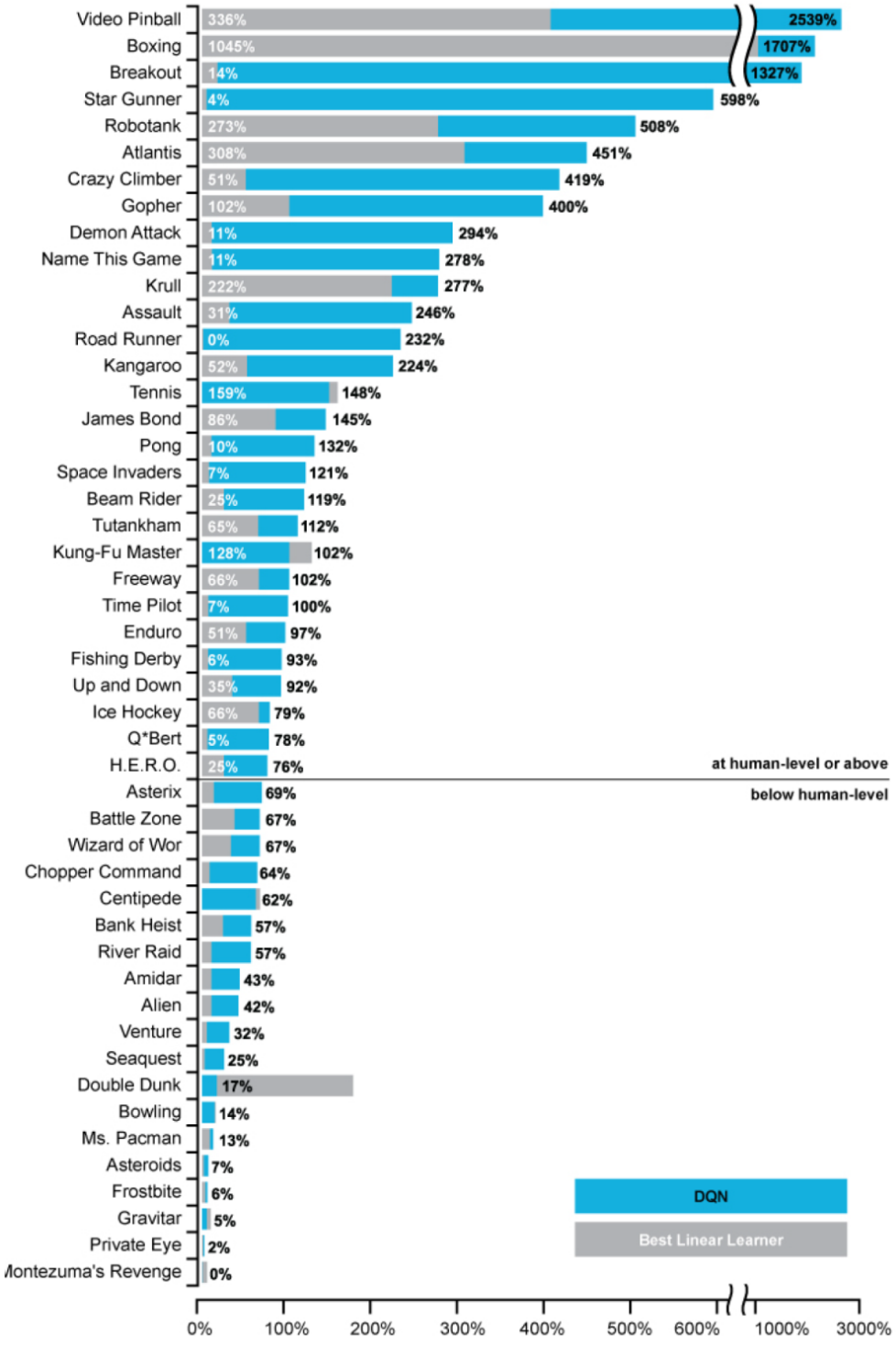
- To deal with non-stationarity, target parameters is fixed

Deep Q-Learning in Atari



Network architecture and hyperparameters fixed across all games

By David Silver



By David Silver

If you want to know more about RL, suggest to read:

Reinforcement Learning: An Introduction.
Richard S. Sutton and **Andrew G. Barto**
Second Edition, in progress
MIT Press, Cambridge, MA, 2017

RL in NLP

- Article summarization
- Question answering
- Dialogue generation
- Dialogue System
- Knowledge-based QA
- Machine Translation
- Text generation
  -
  -
  -

RL in NLP

- **Article summarization**
- Question answering
- Dialogue generation
- Dialogue System
- Knowledge-based QA
- Machine Translation
- Text generation
  -
  -
  -

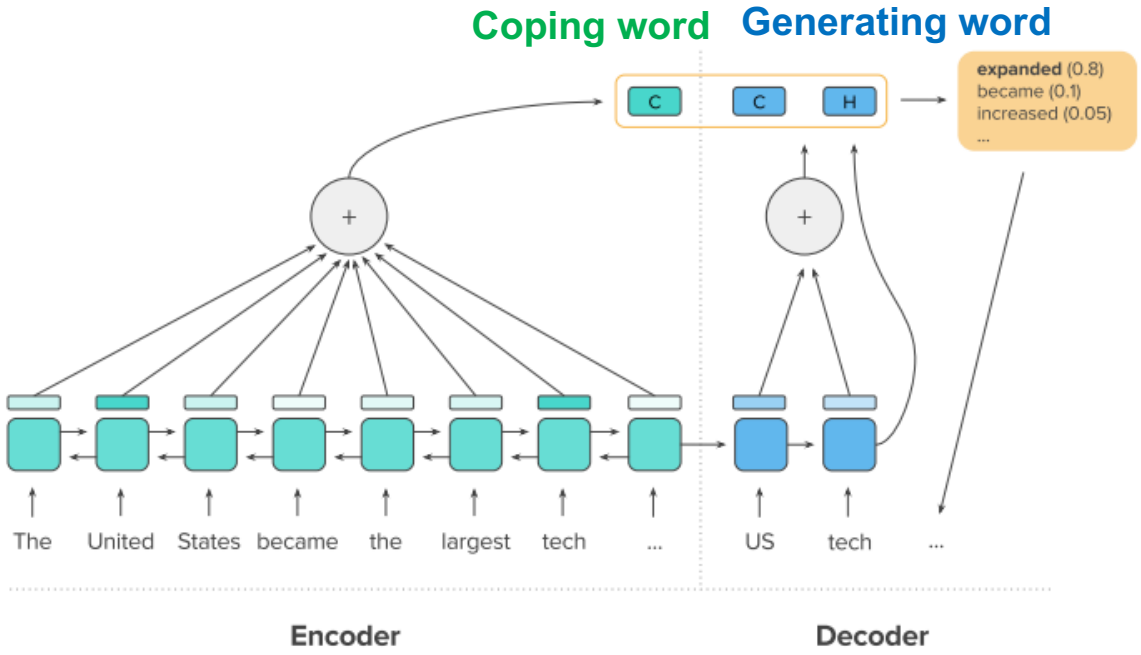
Article Summarization

Text summarization is the process of automatically generating natural language summaries from an input document while retaining the important points.

- extractive summarization
- abstractive summarization

A Deep Reinforced Model for Abstractive Summarization

Given $\mathrm{x} = \{x_1, x_2, \cdots, x_n\}$ represents the sequence of input (article) tokens, $y = \{y_1, y_2, \cdots, y_m\}$, the sequence of output (summary) tokens



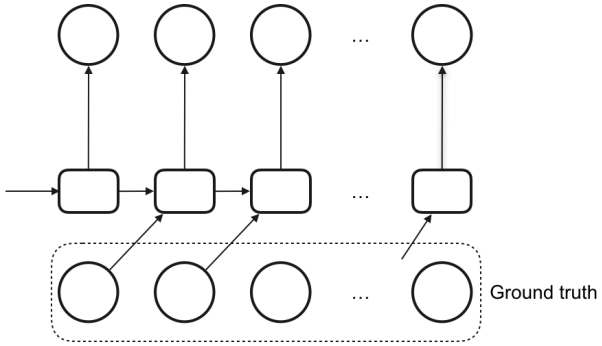
Paulus et. al.

A Deep Reinforced Model for Abstractive Summarization

The maximum-likelihood training objective:

$$L_{ml} = -\sum_{t=1}^{n'} \log p(y_t^* | y_1^*, \ldots, y_{t-1}^*, x)$$

Training with teacher forcing algorithm.



Paulus et. al.

A Deep Reinforced Model for Abstractive Summarization

There is discrepancy between training and test performance, because

- exposure bias
- potentially valid summaries
- metric difference

Paulus et. al.

A Deep Reinforced Model for Abstractive Summarization

Using reinforcement learning framework, learn a policy that maximizes a specific discrete metric.

Action: $u_t \in [copy, generate]$ and word y_t^s

State: hidden states of encoder and previous outputs

Reward: ROUGH score

$$L_{rl} = (r(\hat{y}) - r(y^s)) \sum_{t=1}^{n'} \log p(y_t^s | y_1^s, \ldots, y_{t-1}^s, x)$$

Where $p(y_t^s | y_1^s, \cdots, y_{t-1}^s, x) = p(u_t = copy) p(y_t^s | y_1^s, \cdots, y_{t-1}^s, x, u_t = copy)$
$+ p(u_t = generate) p(y_t^s | y_1^s, \cdots, y_{t-1}^s, x, u_t = generate)$

Paulus et. al.

A Deep Reinforced Model for Abstractive Summarization

Model	ROUGE-1	ROUGE-2	ROUGE-L
words-lvt2k-temp-att (Nallapati et al., 2016)	35.46	13.30	32.65
ML, no intra-attention	37.86	14.69	34.99
ML, with intra-attention	38.30	14.81	35.49
RL, with intra-attention	**41.16**	15.75	**39.08**
ML+RL, with intra-attention	39.87	**15.82**	36.90

Table 1: Quantitative results for various models on the CNN/Daily Mail test dataset

Model	ROUGE-1	ROUGE-2	ROUGE-L
ML, no intra-attention	44.26	27.43	40.41
ML, with intra-attention	43.86	27.10	40.11
RL, no intra-attention	**47.22**	30.51	**43.27**
ML+RL, no intra-attention	47.03	**30.72**	43.10

Table 2: Quantitative results for various models on the New York Times test dataset

Paulus et. al.

A Deep Reinforced Model for Abstractive Summarization

Human readability scores on a random subset of the CNN/Daily Mail test dataset

Model	Readability	Relevance
ML	6.76	7.14
RL	4.18	6.32
ML+RL	**7.04**	**7.45**

Paulus et. al.

RL in NLP

- Article summarization
- **Question answering**
- Dialogue generation
- Dialogue System
- Knowledge-based QA
- Machine Translation
- Text generation
  -
  -
  -

Text Question Answering

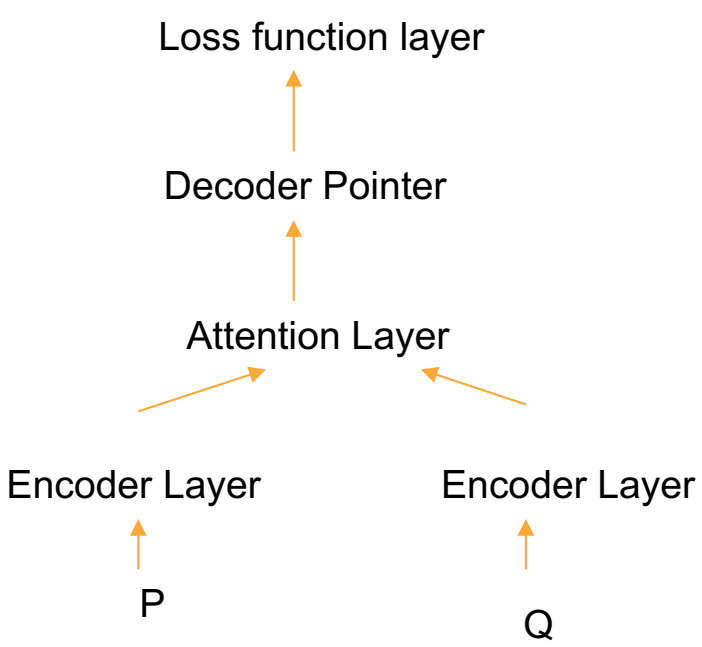
Passage: Tesla later approached Morgan to ask for more funds to build a more powerful transmitter. **When asked where all the money had gone, Tesla responded by saying that he was affected by the Panic of 1901** which he (Morgan) had caused. Morgan was shocked by the reminder of his part in the stock market crash and by Tesla's breach of contract by asking for more funds. Tesla wrote another plea to Morgan, but it was also fruitless. Morgan still owed Tesla money on the original agreement, and Tesla had been facing foreclosure even before construction of the tower began.

Question: On what did Tesla blame for the loss of the initial money?

Answer: Panic of 1901

Example from SQuaD dataset

Text Question Answering



Cross Entropy

LSTM + MLP
GRU + MLP

Self-attention
biAttention
Coattention

LSTM,
GRU

DCN+: MIXED OBJECTIVE AND DEEP RESIDUAL COATTENTION FOR QUESTION ANSWERING

Constraints of Cross-Entropy loss:

P: “Some believe **that the Golden State Warriors team of 2017** is one of the greatest teams in NBA history,…”

Q: “which team is considered to be one of the greatest teams in NBA history”

GT: **“the Golden State Warriors team of 2017”**

Ans1: “Warriors”
Ans2: “history”

Xiong et. al.

DCN+: MIXED OBJECTIVE AND DEEP RESIDUAL COATTENTION FOR QUESTION ANSWERING

To address this, we introduce F1 score as extra objective combining with traditional cross entropy loss:

$$
\begin{aligned}
l_{rl}(\Theta) &= -\mathbb{E}_{\hat{\tau}\sim p_\tau}\left[R\left(s, e, \hat{s}_T, \hat{e}_T; \Theta\right)\right] \\
&\approx -\mathbb{E}_{\hat{\tau}\sim p_\tau}\left[F_1\left(\text{ans}\left(\hat{\text{s}}_\text{T}, \hat{\text{e}}_\text{T}\right), \text{ans}\left(\text{s}, \text{e}\right)\right) - F_1\left(\text{ans}\left(\text{s}_\text{T}, \text{e}_\text{T}\right), \text{ans}\left(\text{s}, \text{e}\right)\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
\nabla_\Theta l_{rl}(\Theta) &= -\nabla_\Theta\left(\mathbb{E}_{\hat{\tau}\sim p_\tau}\left[R\right]\right) \\
&= -\mathbb{E}_{\hat{\tau}\sim p_\tau}\left[R\nabla_\Theta \log p_\tau\left(\tau; \Theta\right)\right]
\end{aligned}
$$

Not necessary for variable length.

Xiong et. al.

RL in NLP

- Article summarization
- Question answering
- **Dialogue generation**
- Dialogue System
- Knowledge-based QA
- Machine Translation
- Text generation
  -
  -
  -

Deep Reinforcement Learning for Dialogue Generation

To generate responses for conversational agents.

A: Where are you going? (1)
B: I'm going to the restroom. (2)

A: how old are you? (1)
B: I'm 16. (2)

The LSTM sequence-to-sequence (SEQ2SEQ) model is one type of neural generation model that maximizes the probability of generating a response given the previous dialogue turn. However,

- One concrete example is that SEQ2SEQ models tend to generate highly generic responses
- stuck in an infinite loop of repetitive responses

Li et. al.

Deep Reinforcement Learning for Dialogue Generation

Baseline mutual information model (Li et al. 2015)
A: Where are you going? (1)
B: I'm going to the restroom. (2)
A: See you later. (3)
B: See you later. (4)
A: See you later. (5)
B: See you later. (6)
...
...

A: how old are you? (1)
B: I'm 16. (2)
A: 16? (3)
B: I don't know what you are talking about. (4)
A: You don't know what you are saying. (5)
B: I don't know what you are talking about . (6)
A: You don't know what you are saying. (7)
...

To solve these, the model needs:

- integrate developer-defined rewards that better mimic the true goal of chatbot development
- model the long term influence of a generated response in an ongoing dialogue

Li et. al.

Deep Reinforcement Learning for Dialogue Generation

Definitions:

Action: infinite since arbitrary-length sequences can be generated.

State: A state is denoted by the previous two dialogue turns $[p_i, q_i]$.

Reward: Ease of answering, Information Flow and Semantic Coherence

Li et. al.

Deep Reinforcement Learning for Dialogue Generation

- Ease of answering: avoid utterance with a dull response.

$$r_1 = -\frac{1}{N_{\mathbb{S}}} \sum_{s \in \mathbb{S}} \frac{1}{N_s} \log p_{\text{seq2seq}}(s|a)$$

The $\mathbb{S}$ is a list of dull responses such as “I don’t know what you are talking about”, “I have no idea”, etc.

Li et. al.

Deep Reinforcement Learning for Dialogue Generation

- Information Flow: penalize semantic similarity between consecutive turns from the same agent.

$$r_2 = -\log\cos(h_{p_i}, h_{p_{i+1}}) = -\log\cos\frac{h_{p_i} \cdot h_{p_{i+1}}}{\|h_{p_i}\|\|h_{p_{i+1}}\|}$$

Where h_{p_i} and $h_{p_{i+1}}$ denote representations obtained from the encoder for two consecutive turns p_i and p_{i+1}

Li et. al.

Deep Reinforcement Learning for Dialogue Generation

- Semantic Coherence: avoid situations in which the generated replies are highly rewarded but are ungrammatical or not coherent

$$r_3 = \frac{1}{N_a} \log p_{\text{seq2seq}}(a|q_i, p_i) + \frac{1}{N_{q_i}} \log p^{\text{backward}}_{\text{seq2seq}}(q_i|a)$$

- The final reward for action a is a weighted sum of the rewards

$$r(a, [p_i, q_i]) = \lambda_1 r_1 + \lambda_2 r_2 + \lambda_3 r_3$$

Li et. al.

Deep Reinforcement Learning for Dialogue Generation

- Simulation of two agents taking turns that explore state-action space and learning a policy
  - **Supervised learning for Seq2Seq models**
  - Mutual Information for pretraining policy model
  - Dialogue Simulation between Two Agents

Li et. al.

Deep Reinforcement Learning for Dialogue Generation

- Simulation of two agents taking turns that explore state-action space and learning a policy
  - Supervised learning for Seq2Seq models
  - **Mutual Information for pretraining policy model**
  - Dialogue Simulation between Two Agents

Li et. al.

Deep Reinforcement Learning for Dialogue Generation

- Mutual Information for previous sequence S and response T

$$\log \frac{p(S,T)}{p(S)p(T)}$$

- MMI objective

$$\hat{T} = \arg\max_{T} \left\{ \log p(T|S) - \log p(T) \right\}$$

$$\hat{T} = \arg\max_{T} \left\{ \log p(T|S) - \lambda \log p(T) \right\}$$

λ : controls the penalization for generic response

Li et. al.

Deep Reinforcement Learning for Dialogue Generation

$$\hat{T} = \arg\max_{T} \{ \log p(T|S) - \lambda \log p(T) \}$$

$$\log p(T) = \log p(T|S) + \log p(S) - \log p(S|T)$$

$$\begin{aligned}\hat{T} &= \arg\max_{T} \{ (1-\lambda) \log p(T|S) \\ &\qquad + \lambda \log p(S|T) - \lambda \log p(S) \} \\ &= \arg\max_{T} \{ (1-\lambda) \log p(T|S) + \lambda \log p(S|T) \}\end{aligned}$$

Consider S as $(\boldsymbol{q_i}, \boldsymbol{p_i})$, T as a, we can have

$$r_3 = \frac{1}{N_a} \log p_{\text{seq2seq}}(a|q_i, p_i) + \frac{1}{N_{q_i}} \log p^{\text{backward}}_{\text{seq2seq}}(q_i|a)$$

Li et. al.

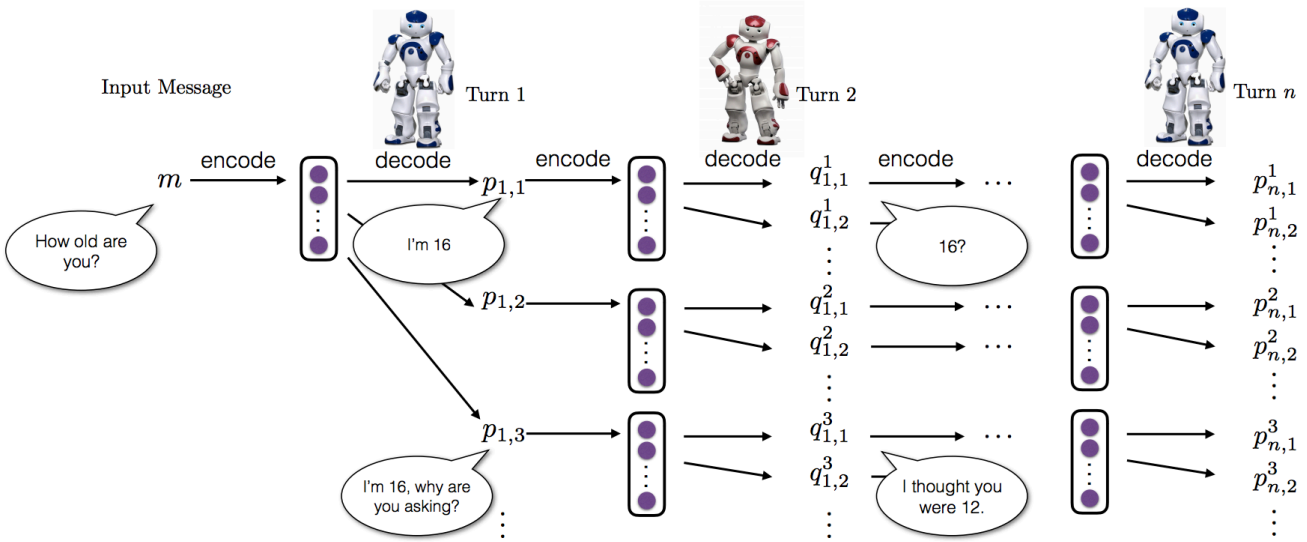
Deep Reinforcement Learning for Dialogue Generation

- Simulation
  - Supervised learning for Seq2Seq models
  - Mutual Information for pretraining policy model
  - **Dialogue Simulation between Two Agents**

Li et. al.

Deep Reinforcement Learning for Dialogue Generation

- Dialogue Simulation between Two Agents



- Using the simulated turns and reward, maximize the expected future reward.
- Training trick: Curriculum Learning

Li et. al.

Deep Reinforcement Learning for Dialogue Generation

Model	# of simulated turns
SEQ2SEQ	2.68
mutual information	3.40
RL	4.48

Input Message	Mutual Information Model	RL
How old are you?	I'm 16.	I'm 16, why are you asking?
What's your full name?	I have no idea.	What's yours?
I don't want to go home tonight.	Really?	Why?
Do you have any feelings for me?	I don't know what you are talking about.	I don't want to hurt your feelings.
How much time do you have here?	Not long enough. Sorry, sir.	Ten seconds.
Shall we get started?	Of course!	Yes. We've got a lot of work to do here.
Do you play football?	No, i don't	Yes. I love football!
We'd have to talk to him.	I mean, he's a good guy	About what ?
How come you never say it?	Because I don't want to hurt you.	I don't think it's a good idea to say it.

Li et. al.

Summary

- The introduction of Reinforcement Learning
- Deep Policy Learning
- Deep Q-Learning
- Applications on NLP
  - Article summarization
  - Question answering
  - Dialogue generation