Review on Dialogue System using Reinforcement Learning

2016-12-03

Kim, Youngsam

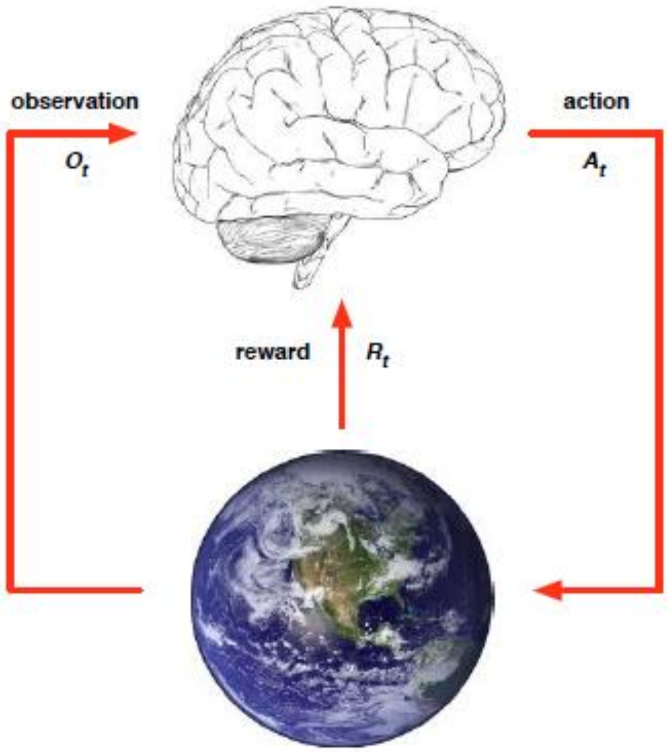
목표

- 현재 강화학습 맥락에서 연구되는 대화 시스템 연구 리뷰
- 리뷰 대상 연구
  - Deep Reinforcement Learning for Dialogue Generation (2016, arxiv) -> StanfordRL
  - Chapt. 4 of Reinforcement Learning for Adaptive Dialogue Systems (2011, Springer) -> SpringerRL
  - SimpleDS: A Simple Deep Reinforcement Learning Dialogue System (2016, arxiv)

Review Focus

- States
  - How state space is modelled?
- Reward
  - How reward function is designed?
- Action
  - How are dialogue acts defined?

General Framework of RL



- At each step t the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step

Components of RL

- A set of states S = {s_i}
- A set of actions A = {a_i}
- A state transition function T(s, a, s')
- A reward function R(s, a, s')
- A policy π: S → A
- The goal of RL agent is to select an action by maximizing its cumulative discounted reward defined as:
- $Q^*(s, a) = \max_{\pi} E(r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t = s, a_t = a)$

대화 시스템 분류

- 응용 목적에 따른 분류
  - QA system
  - Chatbot
  - 전화 자동안내 시스템
- 소통 방식에 따른 분류
  - 음성 기반
  - 텍스트 기반
  - 동작 기반
- 도메인에 따른 분류
  - Open domain / Close domain

Category of the three systems

	Application	Interface	Domain
StanfordRL	Chatbot	Text	Open
SpringerRL	QA or Assistant	Text	Close
SimpleDS	QA or Assistant	Text	Close

Algorithm for policy learning

	Algorithm
StanfordRL	Policy-gradient method with an encoder-decoder recurrent neural model
SpringerRL	Sarsa with linear function approximation
SimpleDS	Deep Q-Learning with experience replay

States: StanfordRL

- A state is denoted by the previous two dialogue turns $[p_i, q_i]$.
- The dialogue history is further transformed to a vector representation by feeding the concatenation of p_i and q_i into an LSTM encoder model.

States: SpringerRL

- Dialogue state contains 8 binary state variables
- Fill-slotN for whether each slot number N is filled (for $1 \leq N \leq 4$)
- Confirm-slotN for whether each slot number N is confirmed
- One DB variable for the number of DB hits (1~100)
- It resulted in $2^{10} \times 100 = 102{,}400$ distinct dialogue states.

States: SimpleDS

- 100-word binary feature vector
- It depends on the vocabulary of the SimpleDS agent in the restaurant domain.
- State transition
  - It is based on a numerical vector representing the last system and user responses.

Environment State	**Action**	**Verbalisation**
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	Salutation (greeting)	Hello!
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	Request (food,price,area)	What type of food, price range, and area are you looking for? [reasonably priced mexican ...food in the east of town]
0,0,0,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0.53,0,0,0,0,0.63,1,0,0,0,0,0,0,0.83,0,0,0,0,0,0,0,0,1,0,0, 0,0.78,0,0,0,0,0.31,0,0,0,1,0.36,1,0.25,0,0,0,0,0,0,0,0,0,0.57,0,0,0.82,1,0,0,0,0,1,0,0,0,0,0,1	ImpConfirm (food,price,area)	Okay, reasonably priced mexican food in the east.
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0, 0,0,0,1,0,0,0,0,0,1,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0	Retrieve (info)	Let me see.
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0, 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	Provide (known)	Restaurant X is an excellent choice. It is located in Y.
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0, 0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,0,0	AskFor (more)	Anything else? [no]
0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0.72,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	Salutation (closing)	Okay, talk to you soon. Bye!
0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1		

Table 1 Example dialogue using the policy from Fig.2, where states are numerical representations of the last system and noisy user inputs, actions are dialogue acts, and user resposes are in brackets

생각거리

- 어떤 상태 표상 방식이 대화상태 표상에 있어 가장 바람직한가?

Action: StanfordRL

- An action is the dialogue utterance to generate.
- The action space is infinite since arbitrary-length sequences can be generated.

Action: SpringerRL

- Pre-defined action sets
  - e.g., ‘greet’, ‘ask a slot’, etc.

- `greet` e.g. “How may I help you?”
- `ask a slot` (`AskASlot`), e.g. “What kind of music would you like?”
- `explicit confirm` (`explicitConf`), e.g. “Did you say Jazz?”
- `implicit confirm and ask a slot` (`implConf-AskASlot`) e.g. “OK, Jazz music. Which artist?
- `close and present information` (`presentList`) e.g. “The following items match your query ...”

Action: SimpleDS

- Action space includes 35 dialogue acts in restaurant domain.
- 2 salutations, 9 requests, 7 apologies, 7 explicit confirmations, 7 implicit confirmation, 1 retrieve info, 2 provide info

[2] Actions: Salutation(greeting), Request(hmihy), Request(food), Request(price), Request(area), Request(food, price), Request(food, area), Request(price, area), Request(food, price, area), Ask-For(more), Apology(food), Apology(price), Apology(area), Apology(food, price), Apology(food, area), Apology(price, area), Apology(food, price, area), ExpConfirm(food), ExpConfirm(price), ExpConfirm(area), ExpConfirm(food, price), ExpConfirm(food, area), ExpConfirm(price, area), ExpConfirm(food, price, area), ImpConfirm(food), ImpConfirm(price), ImpConfirm(area), Imp-Confirm(food,price), ImpConfirm(food, area), ImpConfirm(price, area), ImpConfirm(food, price, area), Retrieve(info), Provide(unknown), Provide(known), Salutation(closing).

생각거리

- 어떤 행위 표상 방식이 가장 바람직한가?

Reward: StanfordRL

- Easy of Answering

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

- Information Flow

$$r_2 = -\log \cos(h_{p_i}, h_{p_{i+1}}) = -\log \cos \frac{h_{p_i} \cdot h_{p_{i+1}}}{\|h_{p_i}\| \|h_{p_{i+1}}\|} \qquad (2)$$

- Semantic Coherence

$$r_3 = \frac{1}{N_a} \log p_{\text{seq2seq}}(a|q_i, p_i) + \frac{1}{N_{q_i}} \log p^{\text{backward}}_{\text{seq2seq}}(q_i|a) \qquad (3)$$

$$r(a, [p_i, q_i]) = \lambda_1 r_1 + \lambda_2 r_2 + \lambda_3 r_3 \qquad (4)$$

Reward: SpringerRL

$$FinalReward = completionValue - dialogueLengthPenalty \quad (4.3)$$

$$-DBhitsPenalty; \quad (4.4)$$

dialogue is being conducted. Thus, where P_c is the probability of a confirmed slot being correct, and P_f is the probability of a filled slot being correct, where C and F are the number of confirmed slots and filled (but not confirmed) slots respectively,

$$completionValue = 100 \times (P_c)^C \times (P_f)^F$$

Reward: SimpleDS

- Human-machine dialogues should confirm the information required and that interactions should be human-like.
- $R(s,a,s') = (CR \times w) + (DR \times (1-w)) - DL$
- CR: Number of positively confirmed slots divided by the slots to confirm.
- W = the weight over the CR (let w = 0.5)
- DR is a data-like probability of having observed action a in state s. $\rightarrow p(a|s)$
- DL is used to encourage efficient interactions (0.1 used)

생각거리

- Reward 함수는 시스템마다 매우 상이해 보인다.
- Open-domain 시스템에 가장 알맞은 reward 함수는 어떤 형태일까?

Summary

- 상태 표상 방법
  - StanfordRL: 대화의 연속적 양상을 neural embedding으로 모델링함
  - SpringerRL: Slot-templat에 기반하여 대화상태를 표현
  - SimpleDS: Bag-of-Word 벡터모형으로 상태를 표상
- 행위 함수 방법
  - StanfordRL: Generation 모형으로 대체함
  - SpringerRL: event template 형태로 표상함
  - SimpleDS: SpringerRL과 유사함
- 보상 함수 방법
  - StanfordRL: 세 종류의 비지도 함수를 가중합산한 방식
  - SpringerRL: 슬롯완충률에서 감점을 적용하는 방식
  - SimpleDS: 슬롯확인률과 행위 관찰 확률을 같이 고려