RL 논문리뷰 - 22.07.25

Language Understanding for Text-based Games using Deep Reinforcement Learning

Karthik Narasimhan, Tejas D Kulkarni, Regina Barzilay

(EMNLP 2015)

>> Table of contents

1

Task

1 Task >> Learning control polices for text-based strategy games

Text-based strategy games

Underlying state is not directly observable
→ Players should understand the text in order to act
→ Challenging for existing AI programs

ex. Fantasy world

- Mission : Quest of finding a secret tomb
- Current State : located on the old bridge
- Choose an action to “Go east”
- Next State (Results of an action) : arrive at ruined gatehouse

State 1: The old bridge
You are standing very close to the bridge’s eastern foundation. If you go east you will be back on solid ground ... The bridge sways in the wind.

Command: **Go east**

State 2: Ruined gatehouse
The old gatehouse is near collapse. Part of its northern wall has already fallen down ... East of the gatehouse leads out to a small open area surrounded by the remains of the castle. There is also a standing archway offering passage to a path along the old southern inner wall.
Exits: Standing archway, castle corner, Bridge over the abyss

2

Background

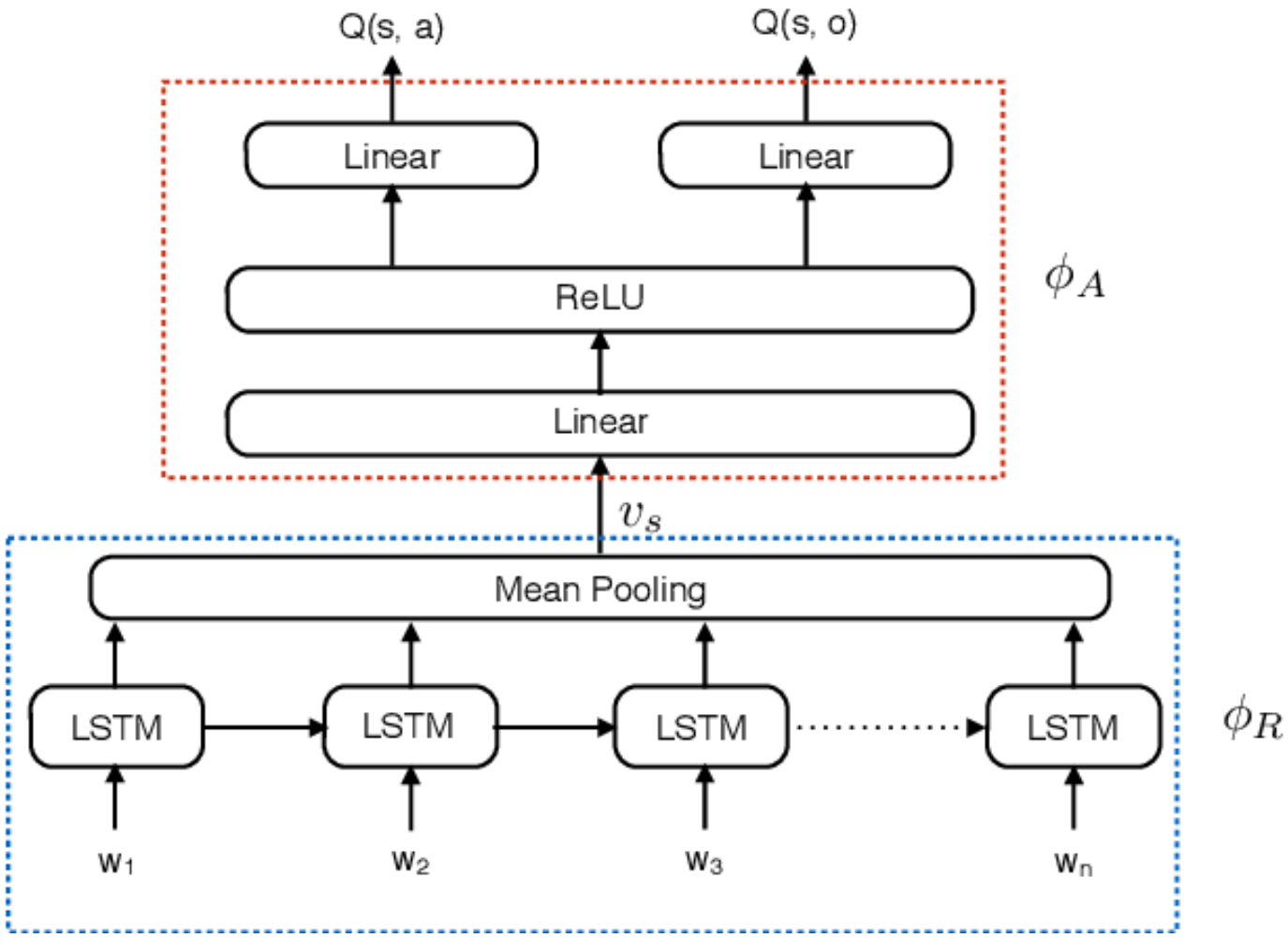
2 Background

- Reinforcement Learning
- Q-Learning
- Deep Q-Network
- Long Short-Term Memory

3

Architecture

Architecture >> LSTM-DQN



Action Scorer

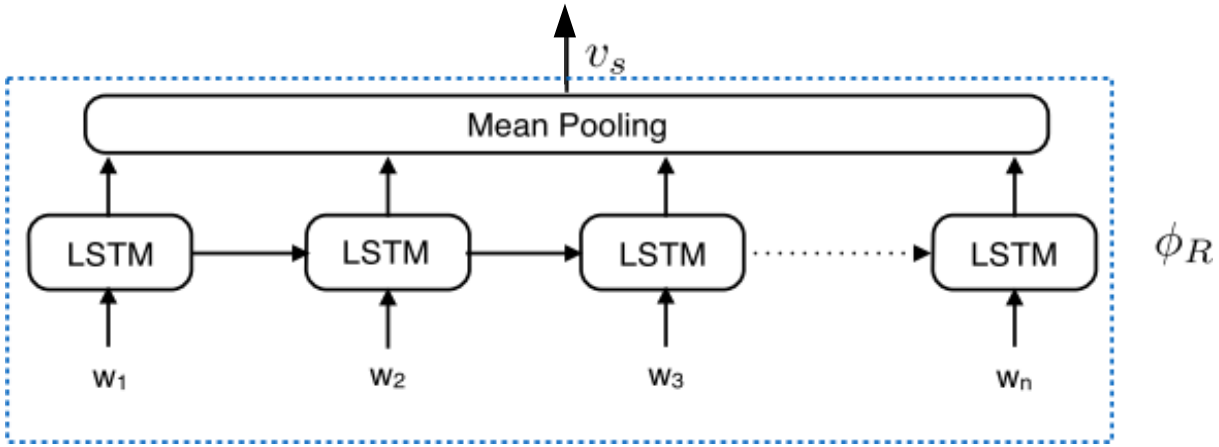
Representation Generator

3 Architecture >> Game Representation

- Represented by the tuple $< H, A, T, R, \Psi >$
- H : the set of all possible game states
- $A = \{(a, o)\}$: all commands (action-object pairs)
- $T(h' \mid h, a, o)$: stochastic transition function between game states
- $R(h, a, o)$: reward function
- $\Psi : H \rightarrow S$: converts game state into a textual description S , since the game state is hidden

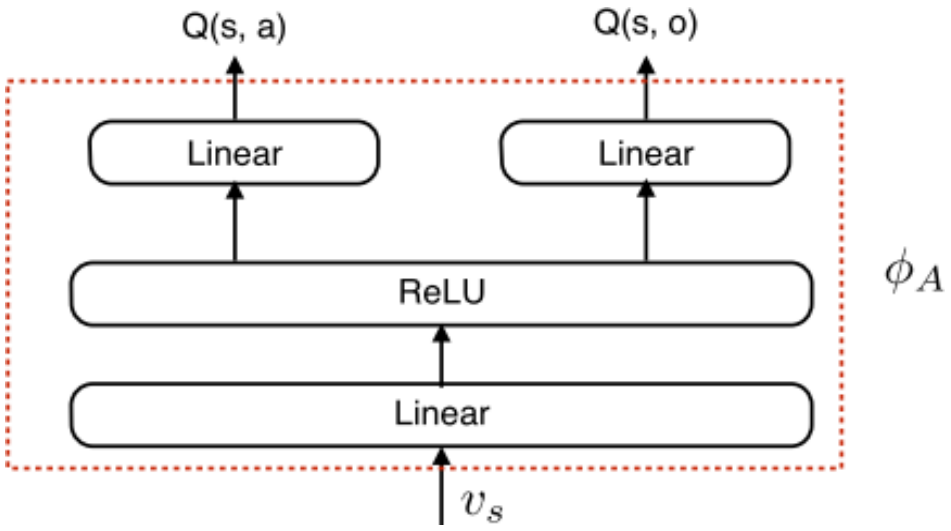
3 Architecture >> 1) Representation Generator ϕ_R

- Reads displayed raw text $w_1, w_2, ..., w_n$
- converts it to a vector representation LSTM： $x_k = f(w_1, ..., w_k)$
- mean pooling layer： $v_s = \frac{1}{n} \sum x_k$



3 Architecture >> 2) Action Scorer ϕ_A

- multi-layered neural network
- given the current state representation v_s , produces scores for actions
- only consider commands with one action and object : (a, o)



3 Parameter Learning

- learning the parameter θ_R , θ_A by using stochastic gradient descent with RMSprop
- keep track of the agent's previous experiences in a memory D
- sample a random transition $(\hat{s}, \hat{a}, s'r)$ from D , and updating the parameters

$$L(\theta_i) = E_{\hat{s},\hat{a}}[(r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(\hat{s}, \hat{a}; \theta_i))^2]$$

- prioritized sampling to the transition with reward $r > 0$

3 Algorithm

Algorithm 1 Training Procedure for DQN with prioritized sampling

Initialize experience memory $\mathcal{D}$
Initialize parameters of representation generator (ϕ_R) and action scorer (ϕ_A) randomly
for $episode = 1, M$ **do**
    Initialize game and get start state description s_1
    for $t = 1, T$ **do**
        Convert s_t (text) to representation v_{s_t} using ϕ_R
        if $random() < \epsilon$ **then**
            Select a random action a_t
        else
            Compute $Q(s_t, a)$ for all actions using $\phi_A(v_{s_t})$
            Select $a_t = \text{argmax}\ Q(s_t, a)$
        Execute action a_t and observe reward r_t and new state s_{t+1}
        Set priority $p_t = 1$ if $r_t > 0$, else $p_t = 0$
        Store transition $(s_t, a_t, r_t, s_{t+1}, p_t)$ in $\mathcal{D}$
        Sample random mini batch of transitions $(s_j, a_j, r_j, s_{j+1}, p_j)$ from $\mathcal{D}$,
        with fraction ρ having $p_j = 1$
        Set $y_j = \begin{cases} r_j & \text{if } s_{j+1} \text{ is terminal} \\ r_j + \gamma \max_{a'} Q(s_{j+1}, a'; \theta) & \text{if } s_{j+1} \text{ is non-terminal} \end{cases}$
        Perform gradient descent step on the loss $\mathcal{L}(\theta) = (y_j - Q(s_j, a_j; \theta))^2$

4

Experiment & Result

4 Experiment & Result

Two World

- Home world : hand-created simple testcase
- Fantasy World : real MUD game testcase

Evaluation

- the cumulative reward
- the fraction of quests completed by the agent

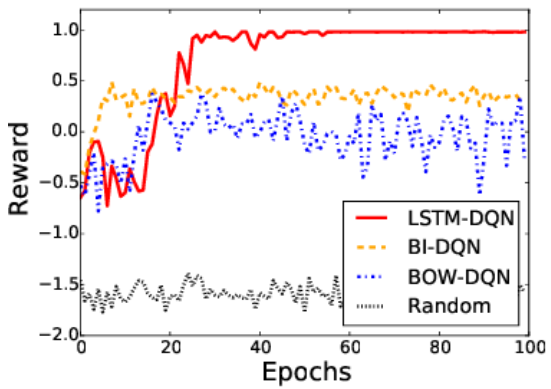
Baseline

- Random agent
- Bag-of-words
- bigram

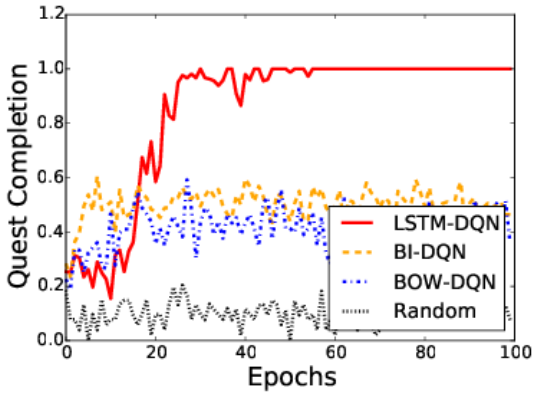
4 Experiment & Result

Stats	Home World	Fantasy World
Vocabulary size	84	1340
Avg. words / description	10.5	65.21
Max descriptions / room	3	100
# diff. quest descriptions	12	-
State transitions	Deterministic	Stochastic
# states (underlying)	16	≥ 56
Branching factor (# commands / state)	40	222

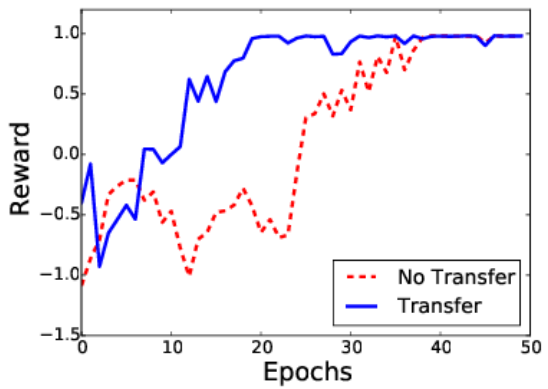
4 Experiment & Result >> Reward, Quest Completion



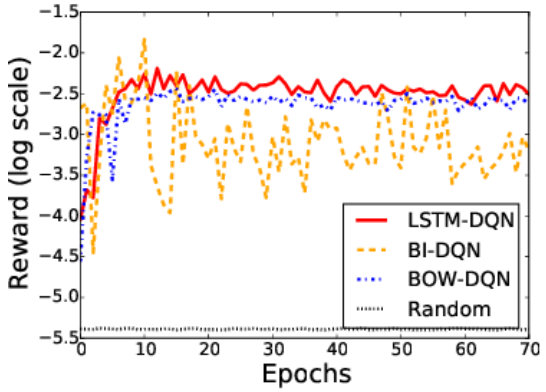
Reward (Home)



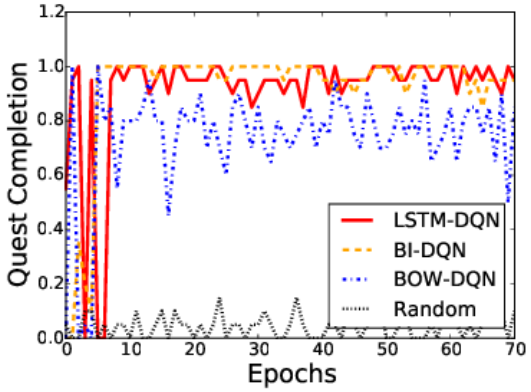
Quest completion (Home)



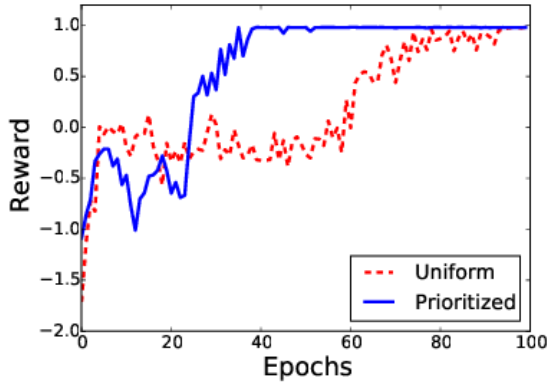
Transfer Learning (Home)



Reward (Fantasy)

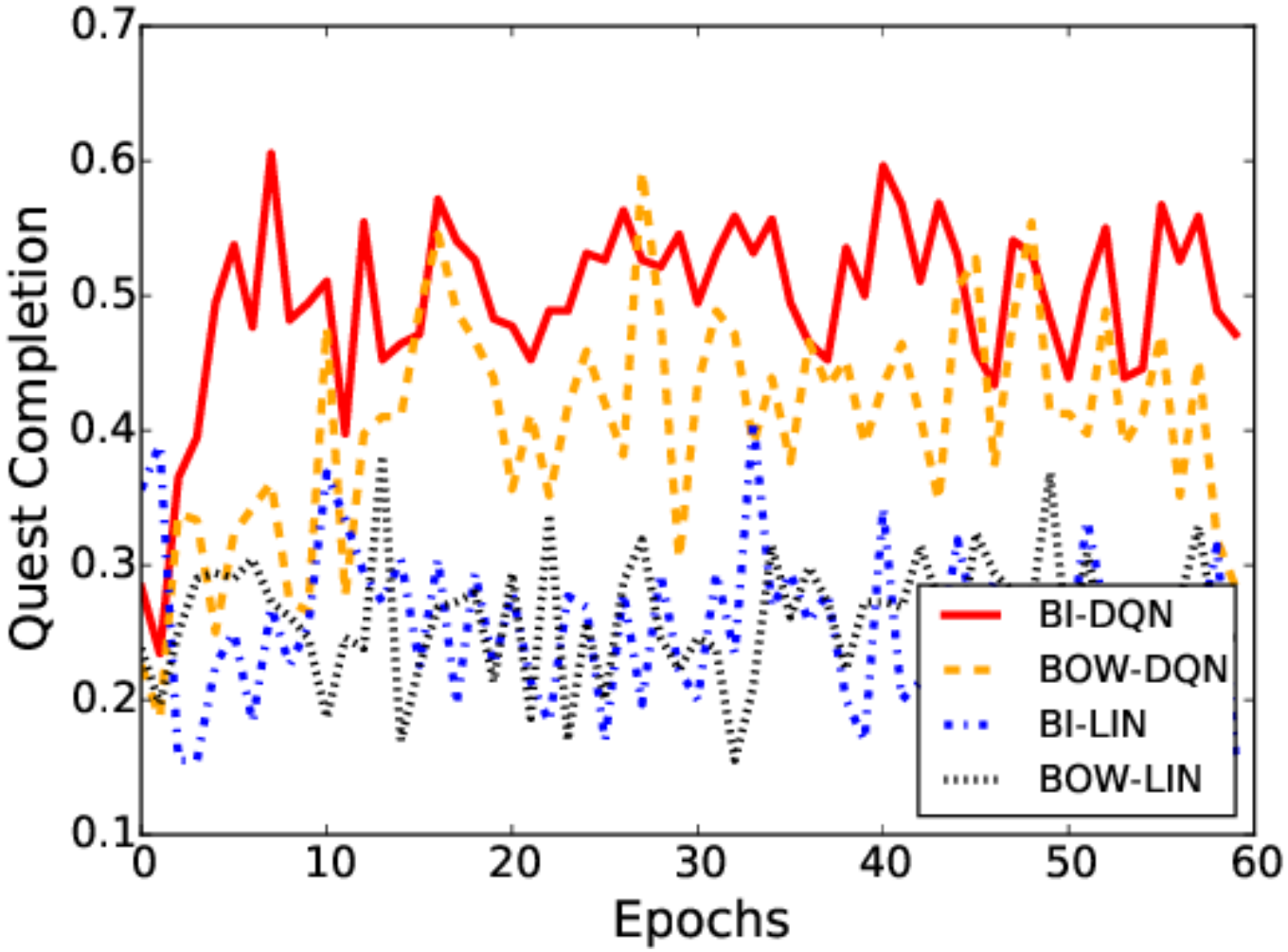


Quest completion (Fantasy)

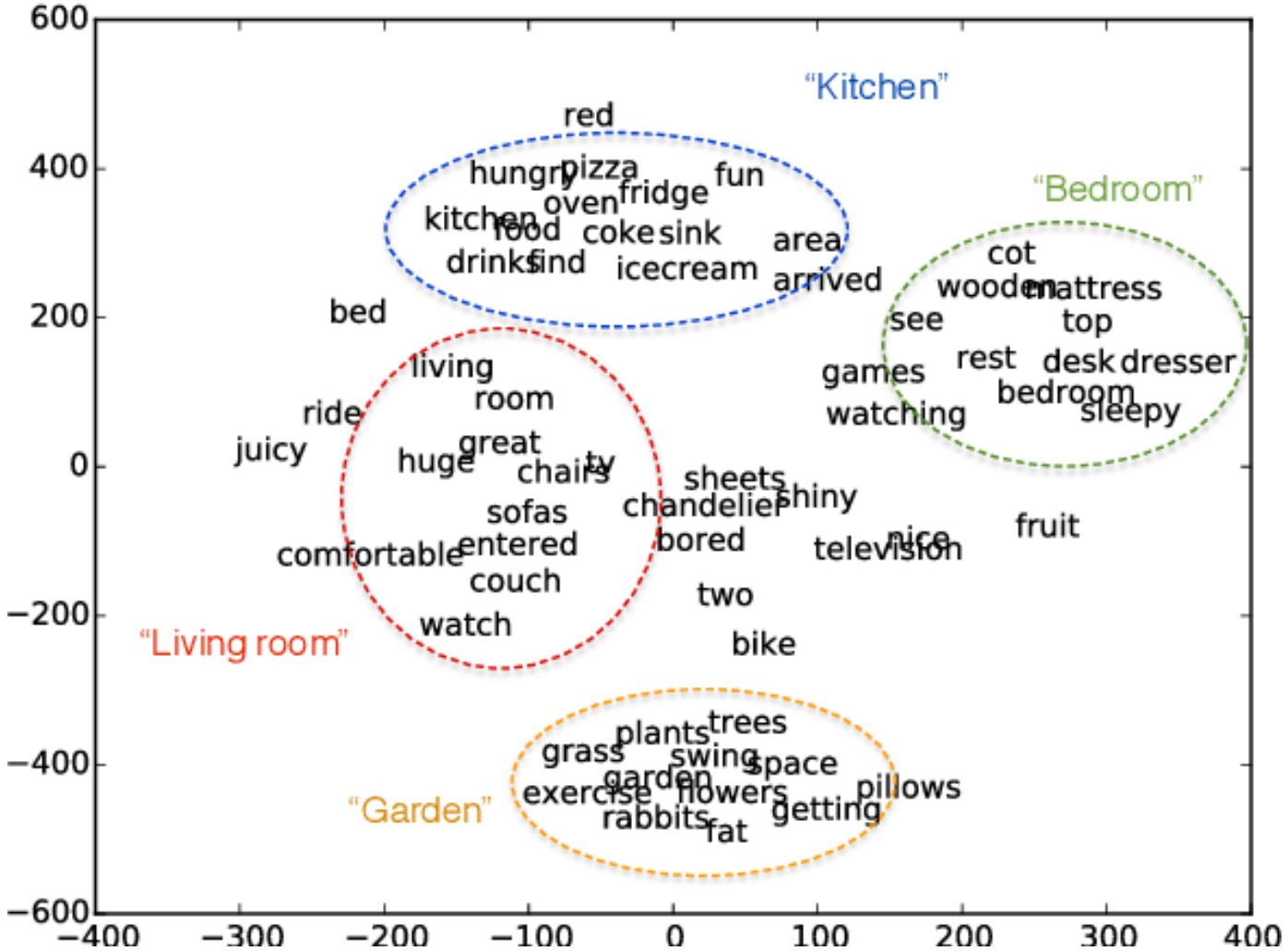


Prioritized Sampling (Home)

4 Experiment & Result



4 Experiment & Result



5

Conclusion & Discussion

5 Conclusion

- a deep reinforcement learning framework
- jointly learn state representations and action policies using game rewards as feedback.
- mapping text descriptions into vectors that capture the semantics of the game states.

5 Discussion

- Text Representation 방법, Action Scoring 방법, optimizer 등 개선 방법?
- 기존 Task에 강화학습 적용 사례? (NLP + RL, Recommendation + RL 등)


Q&A