Reinforcement Learning in Natural Language Processing

Minlie Huang (黄民烈)

Dept. of Computer Science, Tsinghua University

aihuang@tsinghua.edu.cn

http://aihuang.org/p

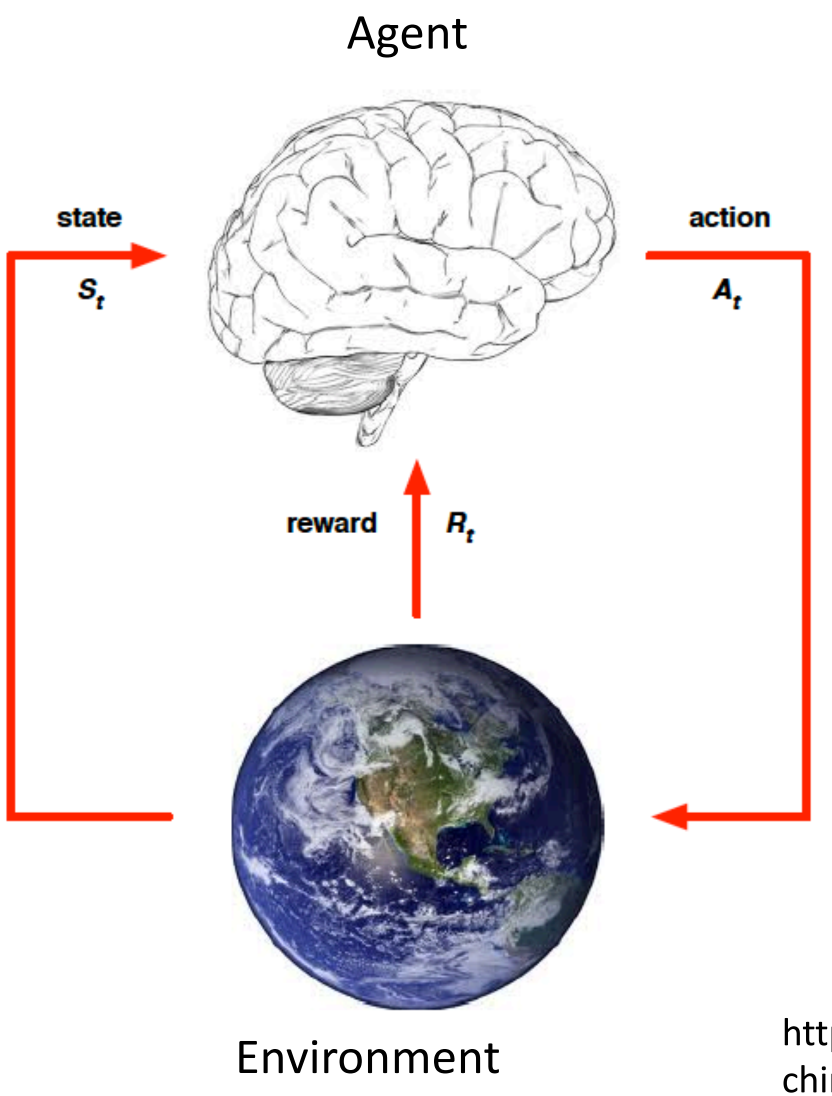
About Me (Minlie Huang)

- Associate Professor, CS Depart., Tsinghua University
- Homepage: http://aihuang.org/p
- Research Interests
  - Deep learning
  - Deep reinforcement learning
  - Generalized QA: QA, Read Comprehension, Story Comprehension
  - Dialogue systems: task-oriented, open-domain
  - Language generation
  - Sentiment/Emotion understanding

Our Recent Works on RL

- Brief Introduction to reinforcement learning (RL)
- Learning Structured Representation with RL (**AAAI 2018**)
  - Policy gradient
- Relation Classification from Noisy Data (**AAAI 2018**)
  - 入选**PaperWeekly** 2017年度最值得读的10篇NLP论文
  - Policy gradient
- Weakly Supervised Topic Labeling in Customer Dialogues (**IJCAI 2018**)
  - Policy gradient
- Learning to Collaborate: Joint Ranking Optimization (**WWW 2018**)
  - Multi-agent reinforcement learning; deterministic policy; actor-critic

Reinforcement Learning

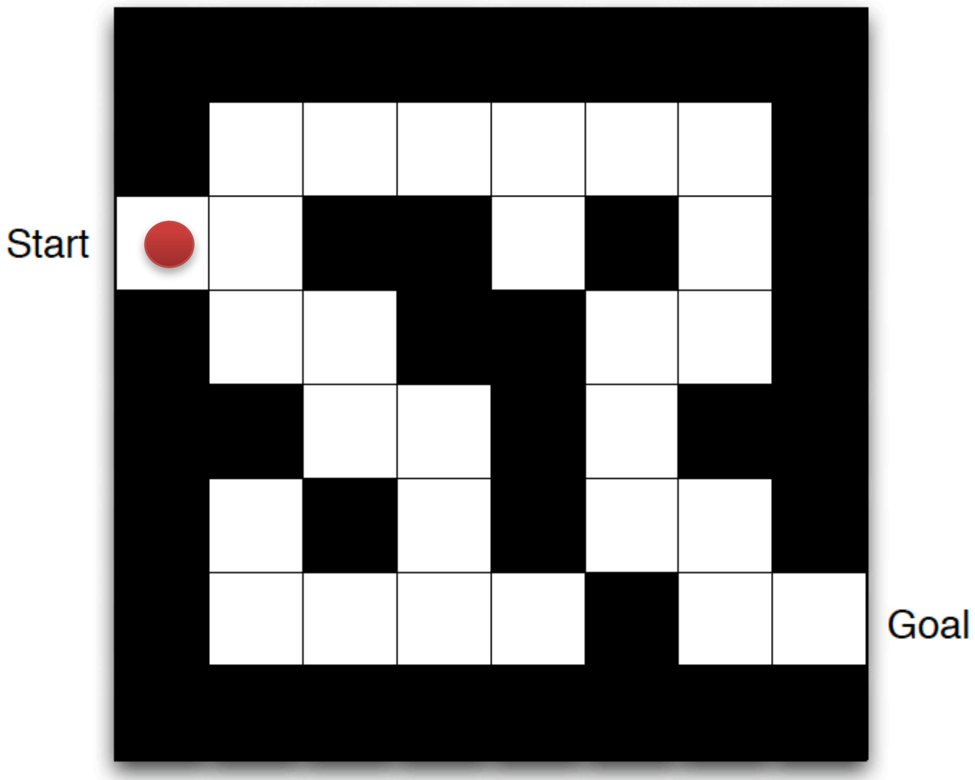


At each step t:

- The agent receives a **state** S_t from the environment
- The agent executes **action** A_t based on the received state
- The agent receives scalar **reward** R_t from the environment
- The environment transfers into a new state S_{t+1}

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/intro_RL.pdf

Maze Example

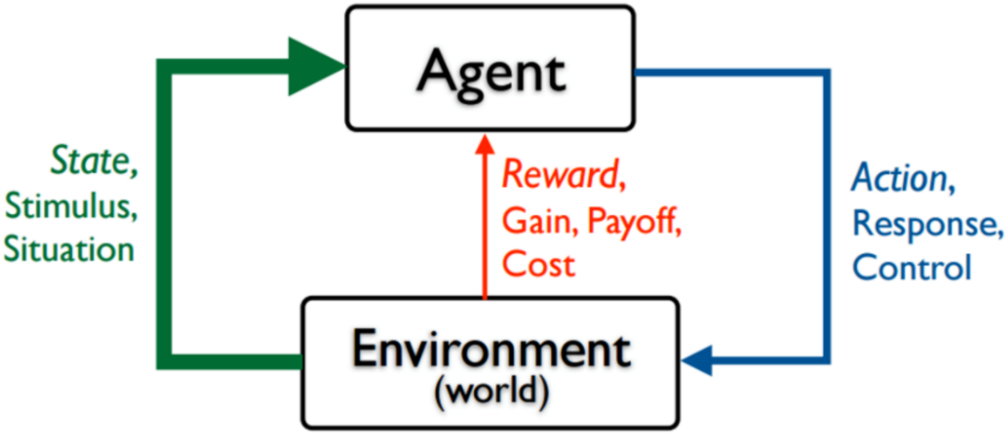


States: Agent's location
Actions: N, E, S, W
Rewards:

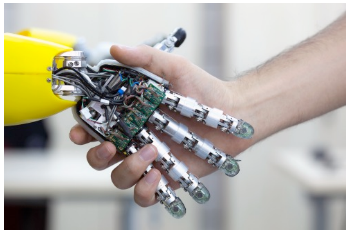
- 100 if reaching the goal
- -100 if reaching the dead end
- -1 per time-step

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/intro_RL.pdf

Deep Reinforcement Learning



Deep learning to represent states, actions, or policy functions



Robotics, control

Self-driving

Language interaction

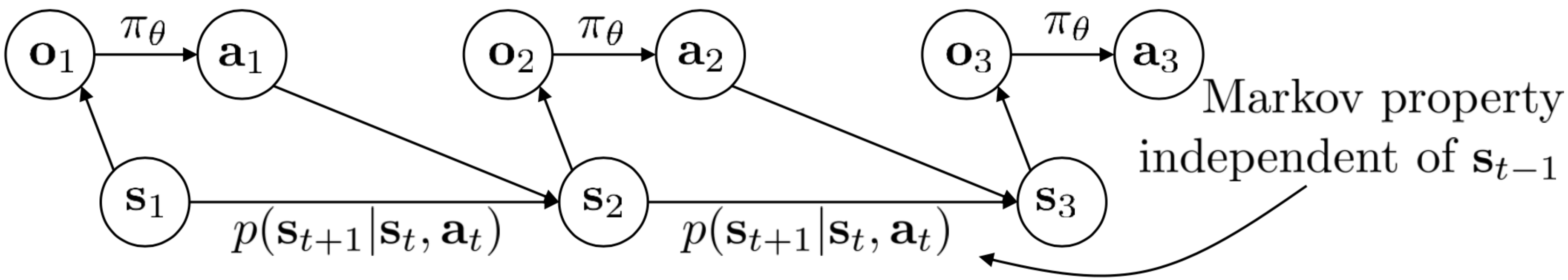
System operating

Reinforcement Learning

- Markov Decision Process

$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)



From ICML Tutorial by Sergey Levine and Chelsea Finn

Reinforcement Learning

$$\underbrace{p_\theta(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T)}_{\pi_\theta(\tau)} = p(\mathbf{s}_1) \prod_{t=1}^{T} \underbrace{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)}_{\text{Markov chain}}$$

$p_\theta(\mathbf{s}_t, \mathbf{a}_t)$ state-action marginal

$p_\theta(\mathbf{s}, \mathbf{a})$ stationary distribution

$$\theta^\star = \arg\max_\theta E_{(\mathbf{s},\mathbf{a})\sim p_\theta(\mathbf{s},\mathbf{a})}[r(\mathbf{s}, \mathbf{a})]$$

infinite horizon case

From ICML Tutorial by Sergey Levine and Chelsea Finn

Reinforcement Learning

$$\underbrace{p_\theta(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T)}_{\pi_\theta(\tau)} = p(\mathbf{s}_1) \prod_{t=1}^{T} \underbrace{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)}_{\text{Markov chain}}$$

$p_\theta(\mathbf{s}_t, \mathbf{a}_t)$ state-action marginal

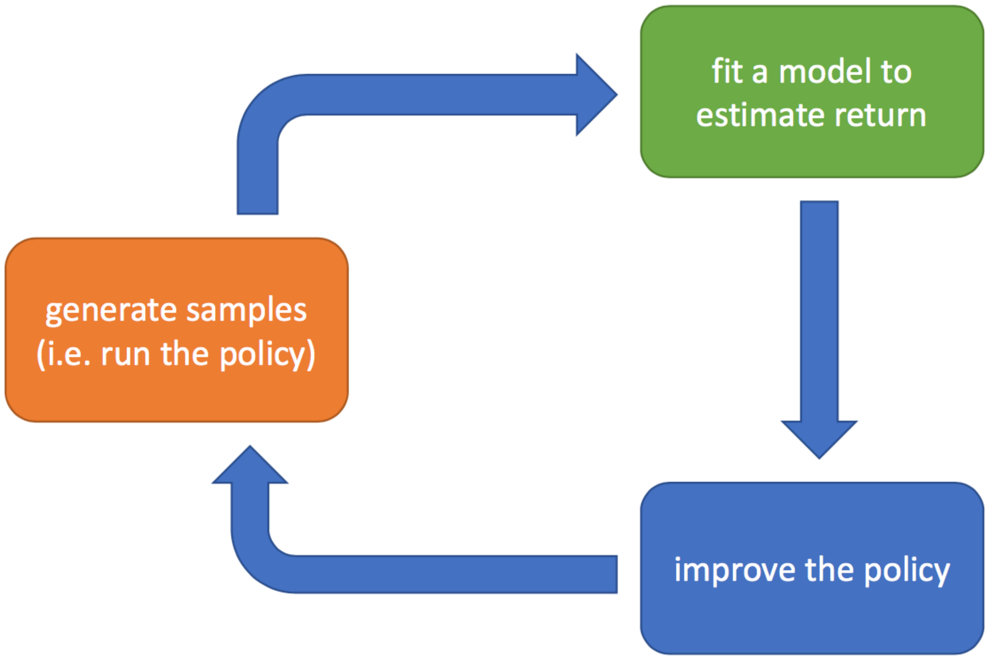
$p_\theta(\mathbf{s}, \mathbf{a})$ stationary distribution

$$\theta^\star = \arg\max_\theta \sum_{t=1}^{T} E_{(\mathbf{s}_t,\mathbf{a}_t)\sim p_\theta(\mathbf{s}_t,\mathbf{a}_t)}[r(\mathbf{s}_t, \mathbf{a}_t)]$$

finite horizon case

From ICML Tutorial by Sergey Levine and Chelsea Finn

Reinforcement Learning



compute $\hat{Q} = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$ (MC policy gradient)

fit $Q_\phi(\mathbf{s}, \mathbf{a})$ (actor-critic, Q-learning)

estimate $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ (model-based)

$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$ (policy gradient)

$\pi(\mathbf{s}) = \arg\max Q_\phi(\mathbf{s}, \mathbf{a})$ (Q-learning)

optimize $\pi_\theta(\mathbf{a}|\mathbf{s})$ (model-based)

From ICML Tutorial by Sergey Levine and Chelsea Finn

Policy Gradient

$$J(\theta) = E_{\tau \sim \pi_\theta(\tau)}[r(\tau)] = \int \pi_\theta(\tau) r(\tau) d\tau$$

$$\nabla_\theta J(\theta) = \int \nabla_\theta \pi_\theta(\tau) r(\tau) d\tau$$

$$= \int \pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau) r(\tau) d\tau$$

$$\pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau) = \pi_\theta(\tau) \frac{\nabla_\theta \pi_\theta(\tau)}{\pi_\theta(\tau)} = \nabla_\theta \pi_\theta(\tau)$$

Policy Gradient

$$\nabla_\theta J(\theta) = \int \underline{\nabla_\theta \pi_\theta(\tau)} r(\tau) d\tau$$

$$= \int \underline{\pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau) r(\tau) d\tau}$$

$$= E_{\tau \sim \pi_\theta(\tau)} [\nabla_\theta \log \underline{\pi_\theta(\tau)} r(\tau)]$$

$$\nabla_\theta J(\theta) = E_{\tau \sim \pi_\theta(\tau)} \left[\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_t | \mathbf{s}_t) \right) \left(\sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t) \right) \right]$$

Reinforcement Learning

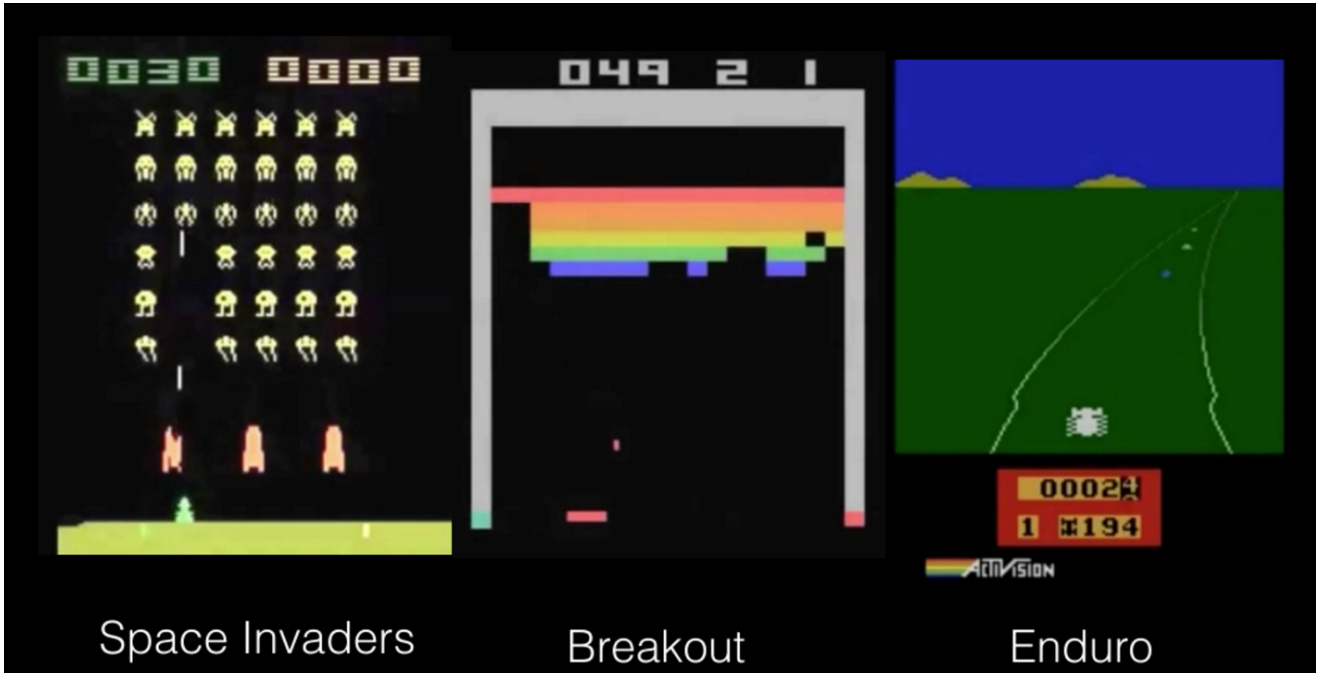
- Difference to supervised learning

- **Sequential decision:** current decision affects future decision
- **Trial-and-error**
- **Explore** (new possibilities) and **exploit** (with the current policy)

- **Future reward:** maximizing the future rewards instead of just the intermediate rewards

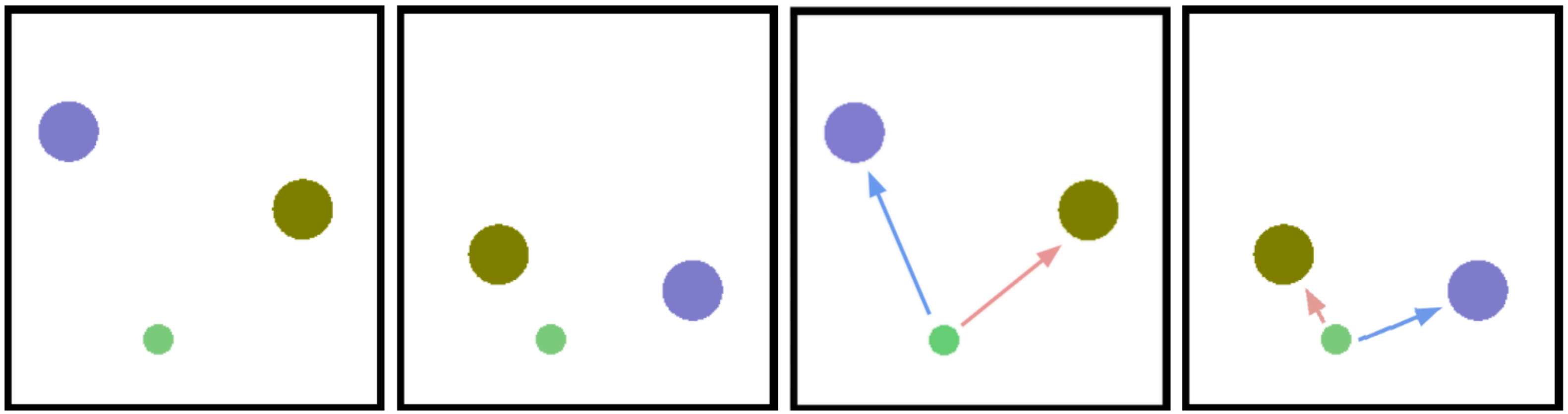
Difference to Supervised Learning

- Supervised learning: given a set of samples (x_i,y_i), estimate $f: X \rightarrow Y$



Difference to Supervised Learning

- You know what a true goal is, but do not know how to achieve that goal
- Through interactions with environment (trial-and-error)
- Many possible solutions (policies), which is optimal?



Applying RL in NLP

- **Challenges**
  - Sparse reward (few feedback when making decisions)
  - Difficulty in reward function design
  - High-dimensional action space
  - High variance in training RL algorithms
- **Strengthens of RL**
  - **Weak supervision** without explicit annotations
  - **Trial-and-error**: probabilistic exploring
  - **Accumulative rewards**: encoding expert/prior knowledge in rewards

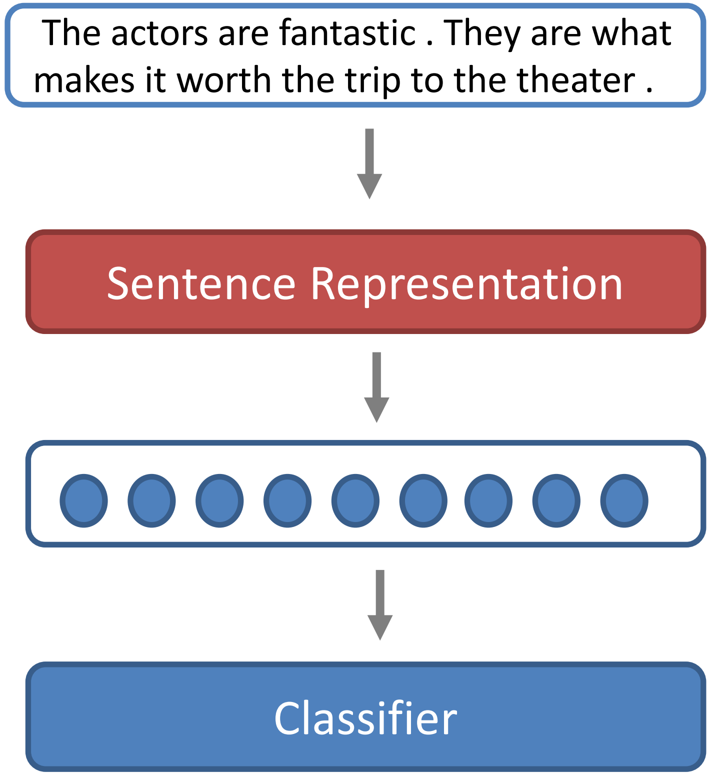
Learning Structured Representation for Text Classification via Reinforcement Learning

Tianyang Zhang, **Minlie Huang**, Li Zhao

AAAI 2018

Background

- **Non-structure model**
  - CNN, RNN, LSTM
  - Bag-of-words models (BM、AE)
- **Using parsing structures**
  - Recursive autoencoders
  - Tree-structured LSTM
- **Auto-learned structure**
  - Binary tree, overly deep



The Problem …

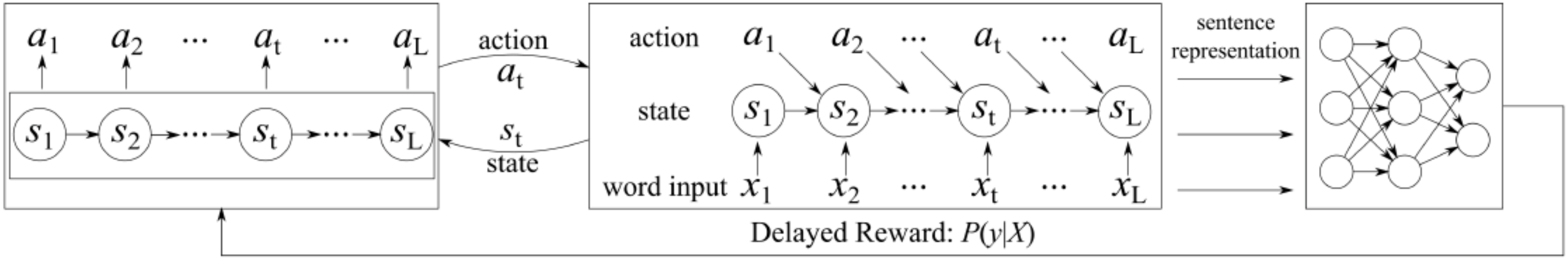
- **How can we identify task-relevant structures without explicit annotations on structure?**

Origin text	Cho continues her exploration of the outer limits of raunch with considerable brio .
ID-LSTM	**Cho continues** ~~her~~ **exploration** ~~of the outer limits of~~ **raunch** ~~with~~ **considerable brio .**
HS-LSTM	Cho | continues her exploration | of the outer limits of raunch | with considerable | brio .
Origin text	Much smarter and more attentive than it first sets out to be .
ID-LSTM	**Much smarter** ~~and more~~ **attentive** ~~than it first sets out to be~~ .
HS-LSTM	Much smarter | and more attentive | than it first sets out to be .
Origin text	Offers an interesting look at the rapidly changing face of Beijing .
ID-LSTM	**Offers** ~~an~~ **interesting look** ~~at the~~ **rapidly changing face** ~~of~~ **Beijing .**
HS-LSTM	Offers | an interesting look | at the rapidly changing | face of Beijing | .

- Challenges
  - **NO explicit** annotations on structure-**weak supervision**
  - **Trial-and-error**, measured by **delayed rewards**

Model Structure

Policy Network(PNet) **Structured Representation Model** **Classification Network(CNet)**



- **Policy Network**:
  - Samples an action at each state
  - Two models: **Information Distilled LSTM**, **Hierarchically Structured LSTM**
- **Structured Representation Model**: transfer action sequence to representation
- **Classification Network**: provide reward signals

Policy Network (PNet)

- **State s_t**
  - Encodes the current input and previous contexts
  - Provided by different representation models
- **Action a_t**
  - {**Retain, Delete**} in **Information Distilled LSTM**
  - {**Inside, End**} in **Hierarchically Structured LSTM**
  - $\pi(a_t|s_t;\Theta) = \sigma(W * s_t + b)$
- **Reward r_t**
  - Calculated from the classification likelihood
  - A factor considering the tendency of structure selection

Policy Network (PNet)

- Maximize the expected reward:

$$
\begin{aligned}
J(\Theta) &= \mathbb{E}_{(\mathbf{s_t},a_t)\sim P_\Theta(\mathbf{s_t},a_t)} r(\mathbf{s_1}a_1\cdots\mathbf{s_L}a_L) \\
&= \sum_{\mathbf{s_1}a_1\cdots\mathbf{s_L}a_L} P_\Theta(\mathbf{s_1}a_1\cdots\mathbf{s_L}a_L)R_L \\
&= \sum_{\mathbf{s_1}a_1\cdots\mathbf{s_L}a_L} p(\mathbf{s_1})\prod_t \pi_\Theta(a_t|\mathbf{s_t})p(\mathbf{s_{t+1}}|\mathbf{s_t},a_t)R_L \\
&= \sum_{\mathbf{s_1}a_1\cdots\mathbf{s_L}a_L} \prod_t \pi_\Theta(a_t|\mathbf{s_t})R_L.
\end{aligned}
$$

- Update the policy network with policy gradient:

$$
\nabla_\Theta J(\Theta) = \sum_{t=1}^{L} R_L \nabla_\Theta \log \pi_\Theta(a_t|\mathbf{s_t})
$$

Classification Network (CNet)

- Cnet is trained via cross entropy (loss function):

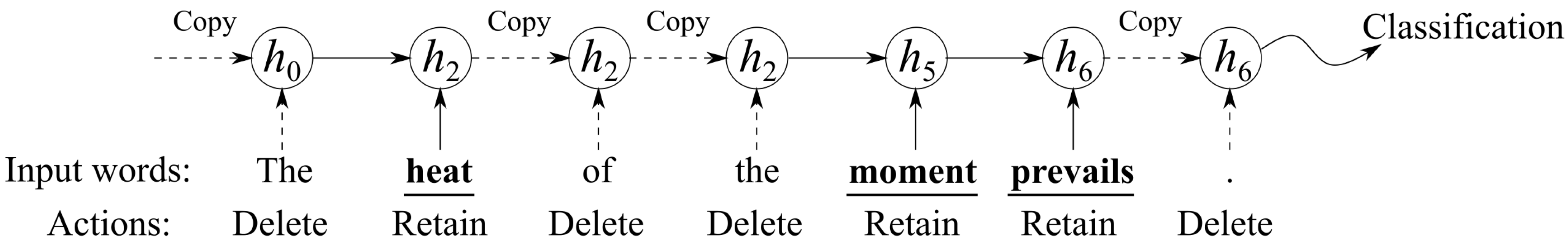
$$P(y|X) = softmax(\mathbf{W_s h_L} + \mathbf{b_s}),$$

$$\mathcal{L} = \sum_{X \in \mathcal{D}} - \sum_{y=1}^{K} \hat{p}(y, X) \log P(y|X)$$

Information Distilled LSTM (ID-LSTM)

- Distill the most important words and remove irrelevant words
- Sentence representation: the last hidden state of ID-LSTM

$$P(y|X) = softmax(\mathbf{W_s h_L} + \mathbf{b_s})$$



Information Distilled LSTM (ID-LSTM)

- Action: {**Retain**, **Delete**}
- States:

$$\mathbf{s_t} = \mathbf{c_{t-1}} \oplus \mathbf{h_{t-1}} \oplus \mathbf{x_t},$$

$$\mathbf{c_t}, \mathbf{h_t} = \begin{cases} \mathbf{c_{t-1}}, \mathbf{h_{t-1}}, & a_t = Delete \\ \Phi(\mathbf{c_{t-1}}, \mathbf{h_{t-1}}, \mathbf{x_t}), & a_t = Retain \end{cases}$$

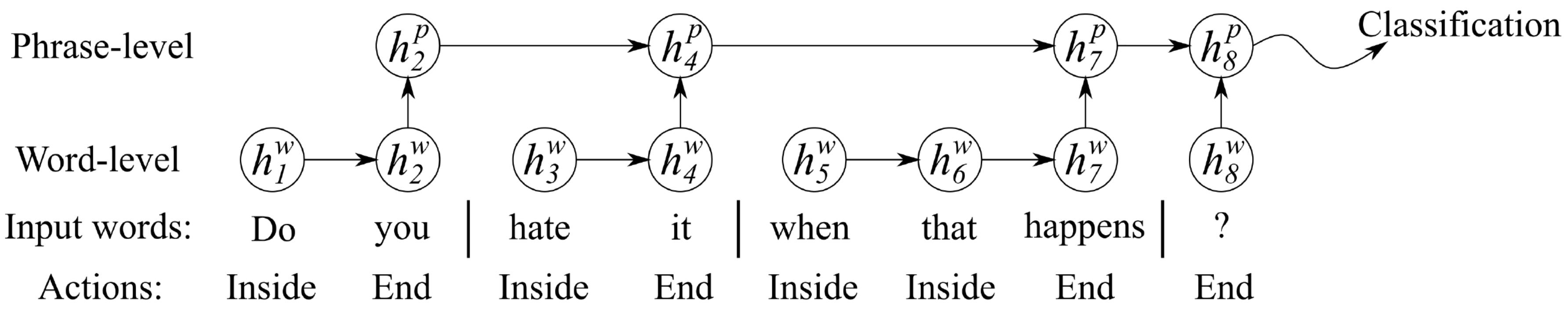
- Rewards:

$$R_L = \log P(c_g|X) + \boxed{\gamma L'/L}$$

the proportion of the number of deleted words to the sentence length

Hierarchically Structured LSTM(HS-LSTM)

- Build a structured representation by discovering hierarchical structures in a sentence
- Two-level structure:
  - Word-level LSTM + phrase-level LSTM
  - Sentence representation: the last hidden state of phrase-level LSTM



Hierarchically Structured LSTM(HS-LSTM)

- Action: {**Inside, End**}

a_{t-1}	a_t	Structure Selection
Inside	Inside	A phrase continues at x_t.
Inside	End	A old phrase ends at x_t.
End	Inside	A new phrase begins at x_t.
End	End	x_t is a single-word phrase.

- States: $\mathbf{s_t} = \mathbf{c^p_{t-1}} \oplus \mathbf{h^p_{t-1}} \oplus \mathbf{c^w_t} \oplus \mathbf{h^w_t}$

Word-level LSTM $\mathbf{c^w_t}, \mathbf{h^w_t} = \begin{cases} \Phi^w(\mathbf{0}, \mathbf{0}, \mathbf{x_t}), & a_{t-1} = End \\ \Phi^w(\mathbf{c^w_{t-1}}, \mathbf{h^w_{t-1}}, \mathbf{x_t}), & a_{t-1} = Inside \end{cases}$

Phrase-level LSTM $\mathbf{c^p_t}, \mathbf{h^p_t} = \begin{cases} \Phi^p(\mathbf{c^p_{t-1}}, \mathbf{h^p_{t-1}}, \mathbf{h^w_t}), & a_t = End \\ \mathbf{c^p_{t-1}}, \mathbf{h^p_{t-1}}, & a_t = Inside \end{cases}$

- Rewards:

$$R_L = \log P(c_g|X) - \boxed{\gamma(L'/L + 0.1L/L')}$$

a unimodal function of the number of phrases (a good phrase structure should contain neither too many nor too few phrases)

Experiment

- Dataset
  - MR: movie reviews (Pang and Lee 2005)
  - SST: Stanford Sentiment Treebank, a public sentiment analysis dataset with five classes (Socher et al. 2013)
  - Subj: subjective or objective sentence for subjectivity classification (Pang and Lee 2004)
  - AG: AG's news corpus, a large topic classification dataset constructed by (Zhang, Zhao, and LeCun 2015)

Experiment

- Classification Results

Models	MR	SST	Subj	AG
LSTM	77.4*	46.4*	92.2	90.9
biLSTM	79.7*	49.1*	92.8	91.6
CNN	81.5*	48.0*	93.4*	91.6
RAE	76.2*	47.8	92.8	90.3
Tree-LSTM	80.7*	**50.1**	93.2	91.8
Self-Attentive	80.1	47.2	92.5	91.1
ID-LSTM	81.6	50.0	93.5	92.2
HS-LSTM	**82.1**	49.8	**93.7**	**92.5**

- Examples by ID-LSTM/HS-LSTM

<table>
<tr><td>Origin text</td><td>Cho continues her exploration of the outer limits of raunch with considerable brio .</td></tr>
<tr><td>ID-LSTM</td><td>Cho continues <s>her</s> exploration <s>of the outer limits of</s> raunch <s>with</s> considerable brio .</td></tr>
<tr><td>HS-LSTM</td><td>Cho | continues her exploration | of the outer limits of raunch | with considerable | brio .</td></tr>
<tr><td>Origin text</td><td>Much smarter and more attentive than it first sets out to be .</td></tr>
<tr><td>ID-LSTM</td><td>Much smarter <s>and more</s> attentive <s>than it first sets out to be</s> .</td></tr>
<tr><td>HS-LSTM</td><td>Much smarter | and more attentive | than it first sets out to be .</td></tr>
<tr><td>Origin text</td><td>Offers an interesting look at the rapidly changing face of Beijing .</td></tr>
<tr><td>ID-LSTM</td><td>Offers <s>an</s> interesting look <s>at the</s> rapidly changing face <s>of</s> Beijing .</td></tr>
<tr><td>HS-LSTM</td><td>Offers | an interesting look | at the rapidly changing | face of Beijing | .</td></tr>
</table>

Results of ID-LSTM

Dataset	Length	Distilled Length	Removed
MR	21.25	11.57	9.68
SST	19.16	11.71	7.45
Subj	24.73	9.17	15.56
AG	35.12	13.05	22.07

Table 4: The original average length and distilled average length by ID-LSTM in the test set of each dataset.

Word	Count	Deleted	Percentage
of	1,074	947	88.18%
by	161	140	86.96%
the	1,846	1558	84.40%
's	649	538	82.90%
but	320	25	7.81%
not	146	0	0.00%
no	73	0	0.00%
good	70	0	0.00%
interesting	25	0	0.00%

Table 5: The most/least deleted words in the test set of SST.

Results of HS-LSTM

Models	SST-binary	AG's News
RAE	85.7	90.3
Tree-LSTM	87.0	91.8
Com-Tree-LSTM	86.5*	—
Par-HLSTM	86.5	91.7
HS-LSTM	**87.8**	**92.5**

Table 8: Classification accuracy from structured models. The result marked with * is re-printed from (Yogatama et al. 2017).

Dataset	Length	#Phrases	#Words per phrase
MR	21.25	4.59	4.63
SST	19.16	4.76	4.03
Subj	24.73	4.42	5.60
AG	35.12	8.58	4.09

Table 9: Statistics of structures discovered by HS-LSTM in the test set of each dataset.

Summary

- A reinforcement learning method which learns sentence representation by discovering task-relevant structure.
- Two representation models: ID-LSTM and HS-LSTM
- State-of-the-art performance & interesting task-relevant structures
- **No direct supervision on structure → trial-and-error!**
  - **Policy gradient**

Reinforcement Learning for Relation Classification from Noisy Data

Jun Feng, **Minlie Huang**, Li Zhao,
Yang Yang, Xiaoyan Zhu

AAAI 2018

Introduction to Relation Classification

- Relation Classification (or extraction)

[**Obama**]$_{e1}$ was born in the [**United States**]$_{e2}$.

↓

Relation: *BornIn*

- Distant Supervision (**noisy labeling problem**)

[**Barack Obama**]$_{e1}$ is **~~the 44th President of~~** the [**United States**]$_{e2}$.

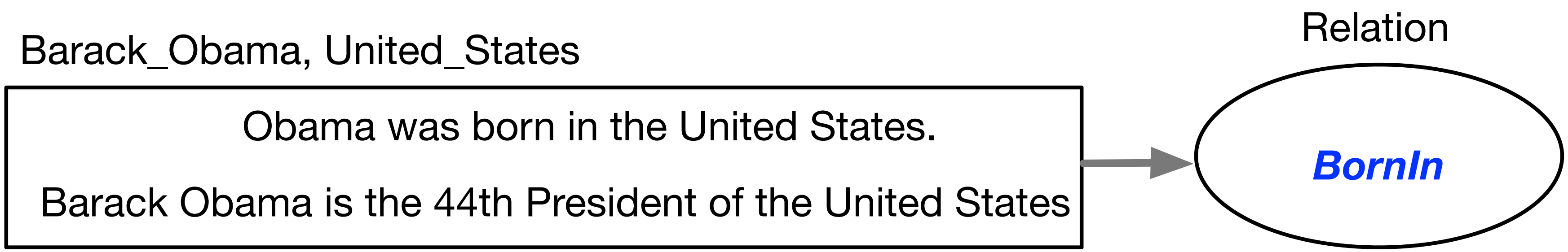
Triple in knowledge base:<Barack_Obama, ***BornIn***, United_States>

↓

Relation: ***BornIn***

The Problem …

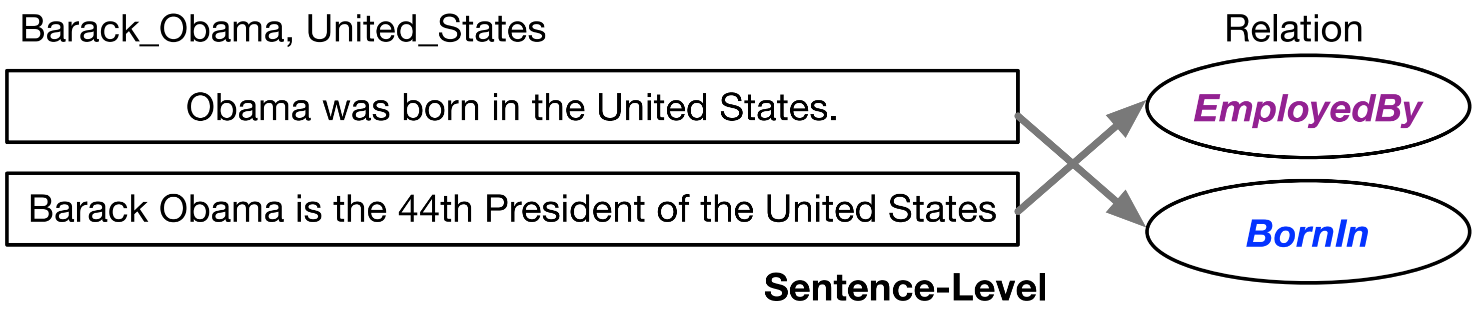
- Previous studies adopt multi-instance learning to consider the instance noises



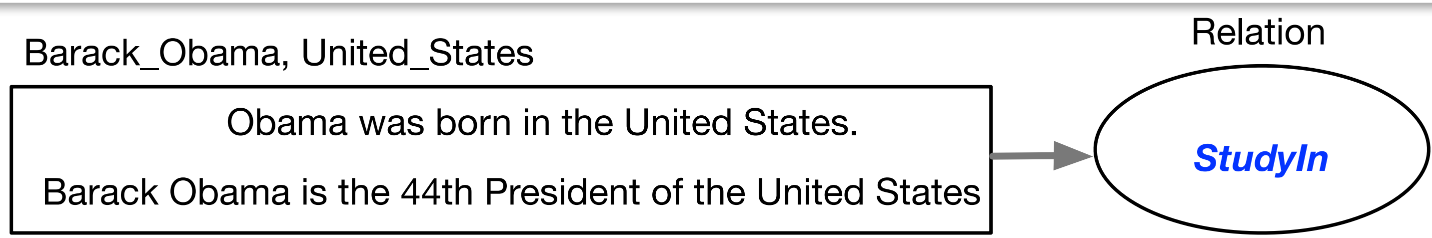
Bag-Level

Motivation

- Two limitations of previous works:
  - Unable to handle the **sentence-level prediction**

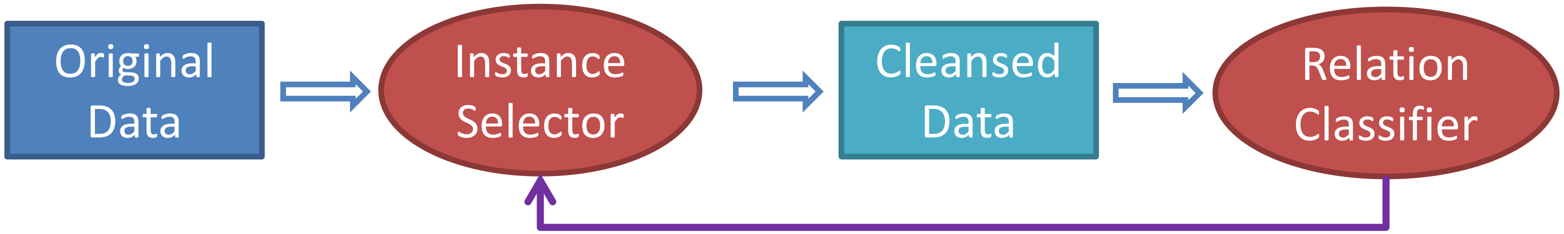


How can we remove noisy data to improve relation extraction without explicit annotations?



Model Structure

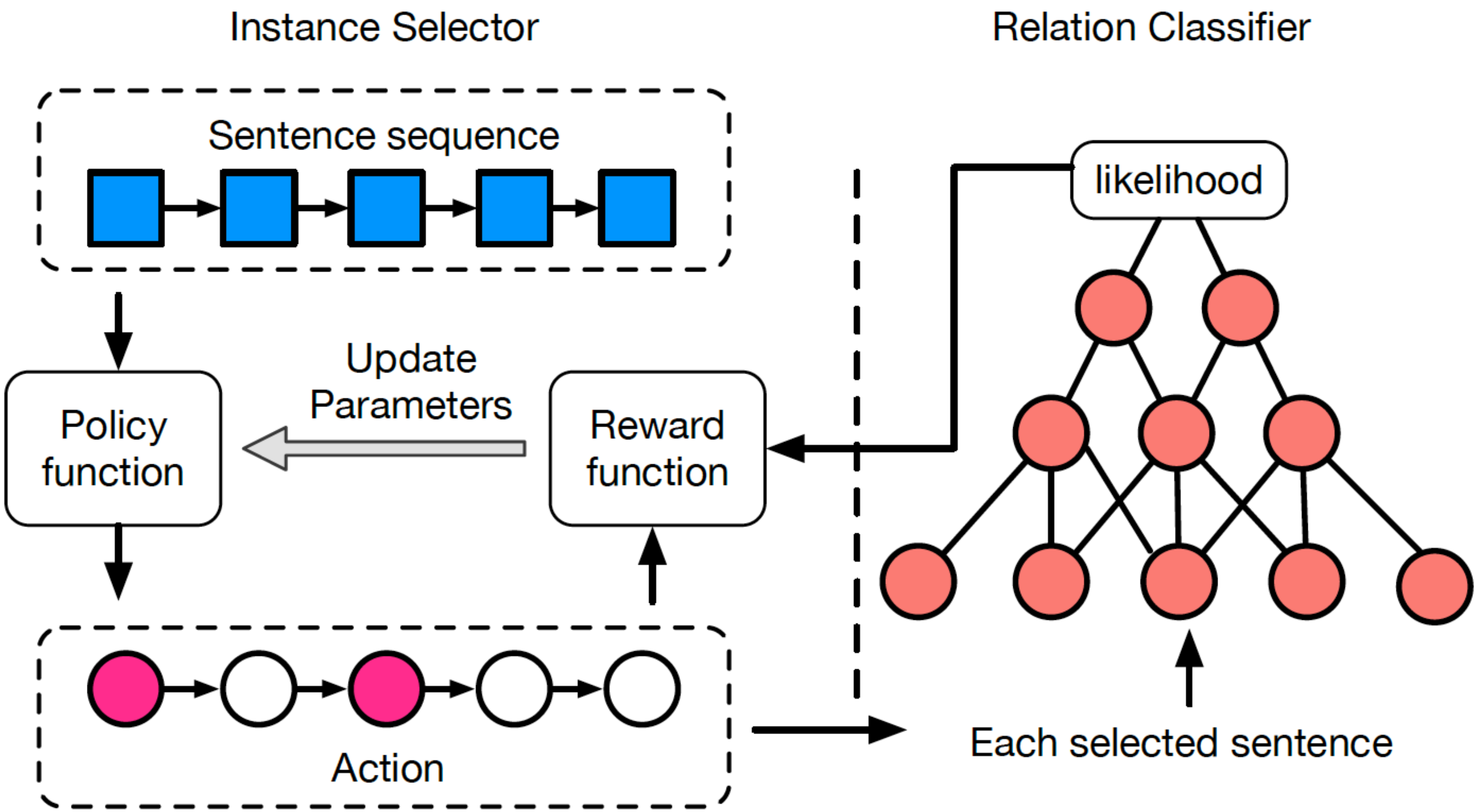
- The model consists of an **instance selector** and **a relation classifier**



- Challenges:
  - Instance selector has no explicit knowledge about which sentences are labeled incorrectly
    - Weak supervision -> delayed reward
    - Trail-and-error search

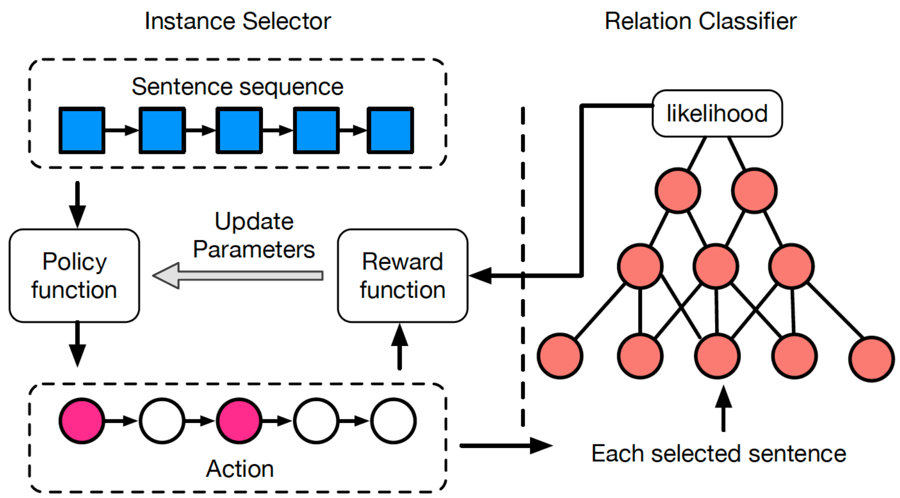
    → Reinforcement Learning
  - How to train the two modules jointly

Model Structure



The Logic Why it Works

- Start from noisy data to pretrain relation classifier and instance selector
- Remove noisy data
- Train better classifier to obtain better reward estimator
- Train better policy with more accurate reward estimator
- Remove noisy data more accurately



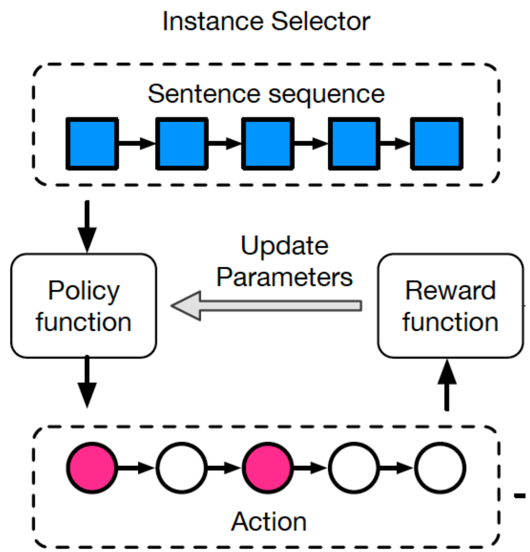
Instance Selector

- Instance selection as a reinforcement learning problem
  - **State**: $\boldsymbol{F(s_i)}$ the current sentence, the already selected sentences, and the entity pair
  - **Action**: {**0,1**}, select the current sentence or not

$$
\begin{aligned}
\pi_\Theta(s_i, a_i) &= P_\Theta(a_i|s_i) \\
&= a_i\sigma(\boldsymbol{W} * \boldsymbol{F}(s_i) + \boldsymbol{b}) \\
&+ (1 - a_i)(1 - \sigma(\boldsymbol{W} * \boldsymbol{F}(s_i) + \boldsymbol{b}))
\end{aligned}
$$

  - **Reward:** the total likelihood of the sent. bag

$$
r(s_i|B) = \begin{cases} 0 & i < |B| + 1 \\ \frac{1}{|\hat{B}|} \sum_{x_j \in \hat{B}} \log p(r|x_j) & i = |B| + 1 \end{cases}
$$



Instance Selector

- Optimization:
  - Maximize the expected total rewards

$$J(\Theta) = V_\Theta(s_1|B)$$

$$= E_{s_1,a_1,s_2,\ldots,s_i,a_i,s_{i+1}\ldots}[\sum_{i=0}^{|B|+1} r(s_i|B)]$$

  - Update parameters with the **REINFORCE** algorithm

$$\Theta \leftarrow \Theta + \alpha \sum_{i=1}^{|B|} v_i \nabla_\Theta \log \pi_\Theta(s_i, a_i)$$

Relation Classifier

- A CNN architecture to classify relations

$$\mathbf{L} = \mathrm{CNN}(\mathbf{x})$$

$$p(r|x;\boldsymbol{\Phi}) = softmax(\mathbf{W}_r * tanh(\boldsymbol{L}) + \mathbf{b}_r)$$

- Optimization: cross-entropy as the objective function

$$\mathcal{J}(\Phi) = -\frac{1}{|\hat{X}|}\sum_{i=1}^{|\hat{X}|}\log p(r_i|x_i;\Phi)$$

Training Procedure

- Overall Training Procedure
  1. Pre-train the CNN model of the relation classifier
  2. Pre-train the policy network of the instance selector with the CNN model fixed
  3. Jointly train the CNN model and the policy network

Experiment

- Dataset
  - NYT and developed by (Riedel, Yao, and McCallum 2010)
- Baselines
  - CNN: is a sentence-level classification model. It does not consider the noisy labeling problem.
  - CNN+Max: assumes that there is one sentence describing the relation in a bag and chooses the most correct sentence in each bag.
  - CNN+ATT: adopts a sentence-level attention over the sentences in a bag and thus can down weight noisy sentences in a bag.

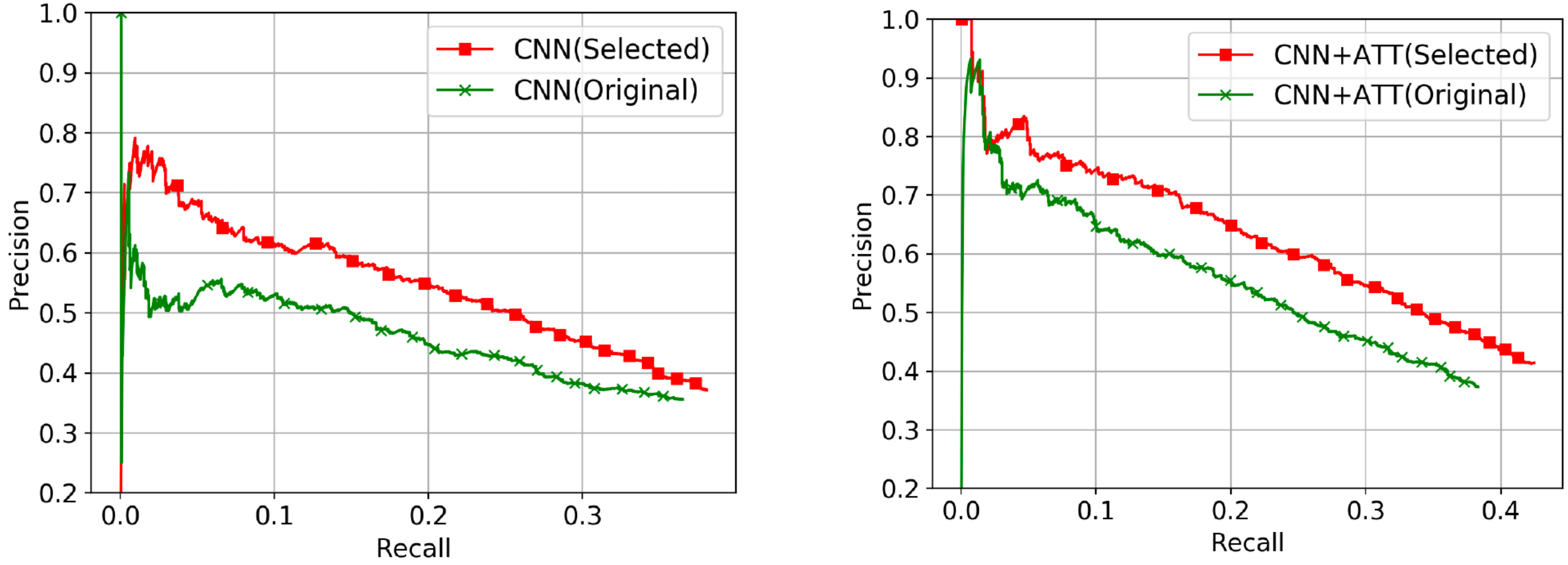
Experiment

- Sentence-Level Relation Classification

Method	Macro F_1	Accuracy
CNN	0.40	0.60
CNN+Max	0.06	0.34
CNN+ATT	0.29	0.56
CNN+RL(ours)	**0.42**	**0.64**

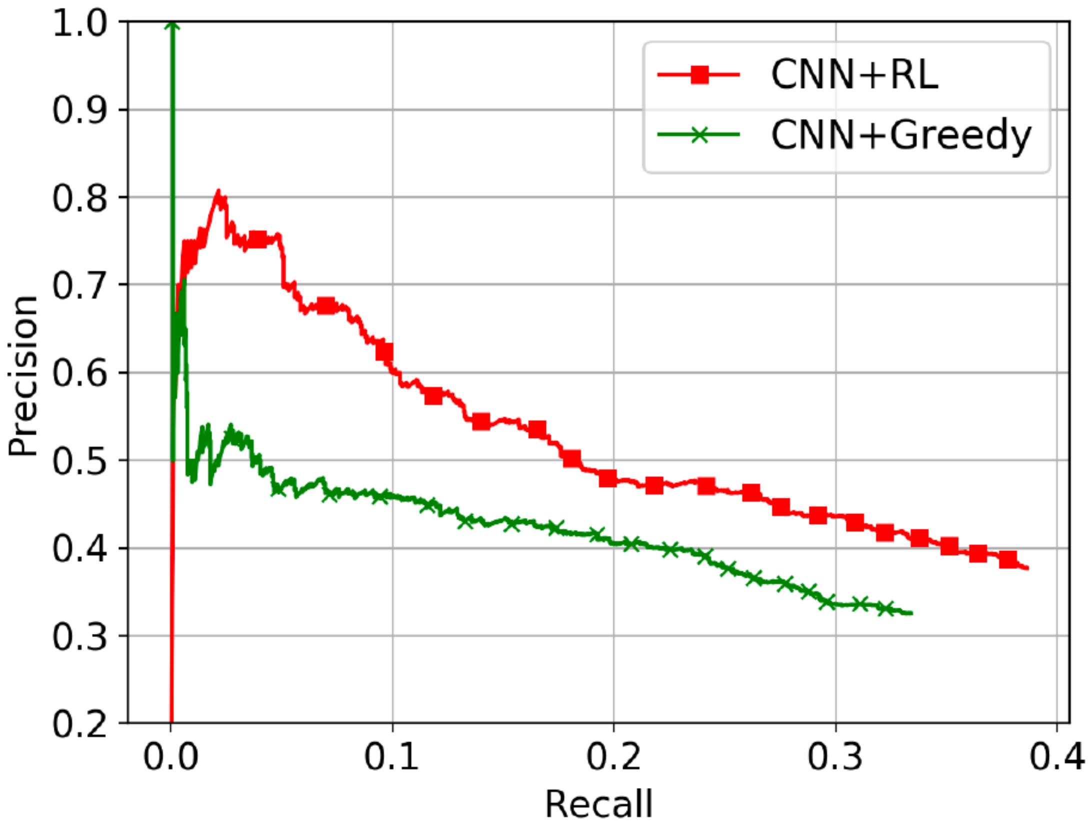
Experiment

- The performance of the instance selector



Experiment

- The performance of the instance selector



Case Study

Bag I (**Entity Pair**: fabrice_santor, france; **Relation**:/people/person/nationality)	CNN+RL	CNN+ATT	CNN+Max
though not without some struggle, federer, the world 's top-ranked player, advanced to the fourth round with a thrilling, victory over the crafty **fabrice_santoro** of **france**, who is ranked 76th.	1	0.60	0
in his quarterfinal , nalbandian overwhelmed unseeded **fabrice_santoro** of **france**	1	0.39	1
fabrice_santoro, 33 , of **france** finally reached the quarterfinals in a major on his 54th attempt by defeating the 11th-seeded spaniard david ferrer	1	0.01	0
Bag II (**Entity Pair**: jonathan_littel, france; **Relation**:/people/person/nationality)			
jonathan_littell, a new york-born writer whose french-language novel about a murderous and degenerate officer has been the sensation of the french publishing season, on monday became the first american to win **france**'s most prestigious literary award, the prix goncourt	0	0.89	1
after a languid intercontinental auction that stretched for more than a week, the american rights to **jonathan_littell**'s novel les bienveillantes, which became a publishing sensation in **france**, have been sold to harpercollins, the publisher confirmed yesterday.	0	0.11	0

Summary

- A new model to extract relations from noisy data.
- Merely with a **weak supervision signal** from the relation classifier.
- The idea for **instance selection** can be generalized to other tasks that employ noisy data or distant supervision.
- **Weak supervision: no annotation on which sentence is noisy!**

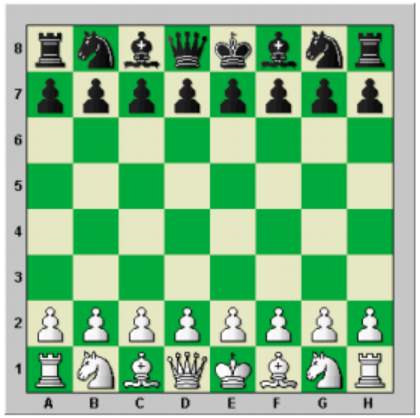
Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning

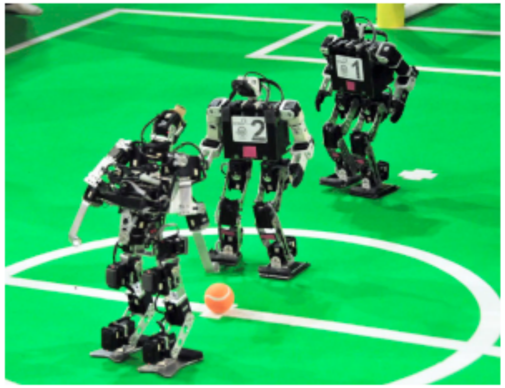
Jun Feng, Heng Li, **Minlie Huang**, Shichen Liu, Wenwu Ou, Zhirong Wang and Xiaoyan Zhu

WWW 2018

Background

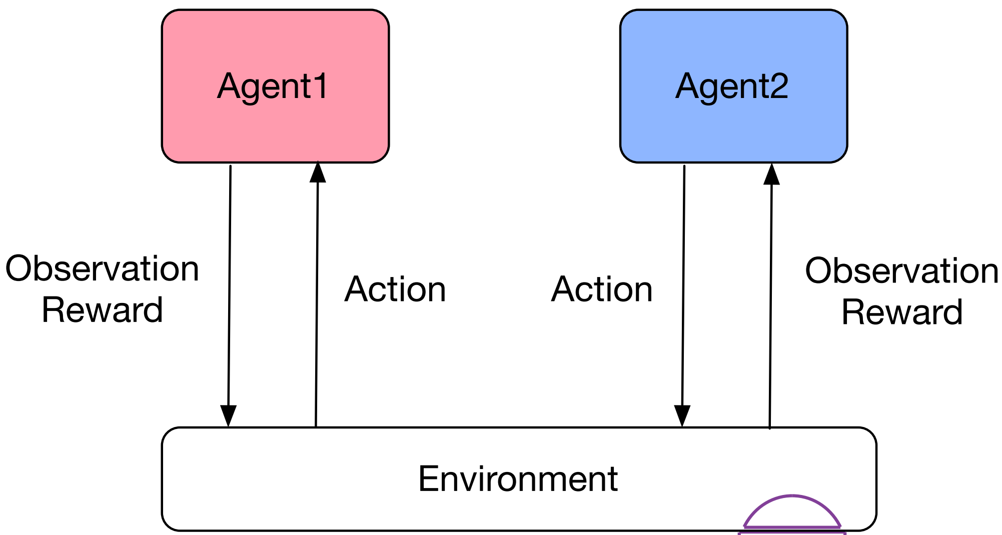
- Examples of multi-agent reinforcement learning problems





Background

- Multi-Agent Reinforcement Learning
  - N agents $A^1, A^2, \ldots, A^N$ interact in a common environment
  - The state s_t is global
  - At time step t, each agent has:
    - its own observation o_t^i
    - its own action a_t^i
    - its own reward $r_i^t = r(s_t, a_t^i)$



Background

- Types of multi-agent reinforcement learning
  - Fully cooperative
    - All the agents have the same goal, maximizing the same objective function
  - Fully competitive
    - Two agents have opposite goals
    - Maximize one's benefit under the worst-case assumption that the opponent will always endeavor to minimize it
  - Mixed

Background

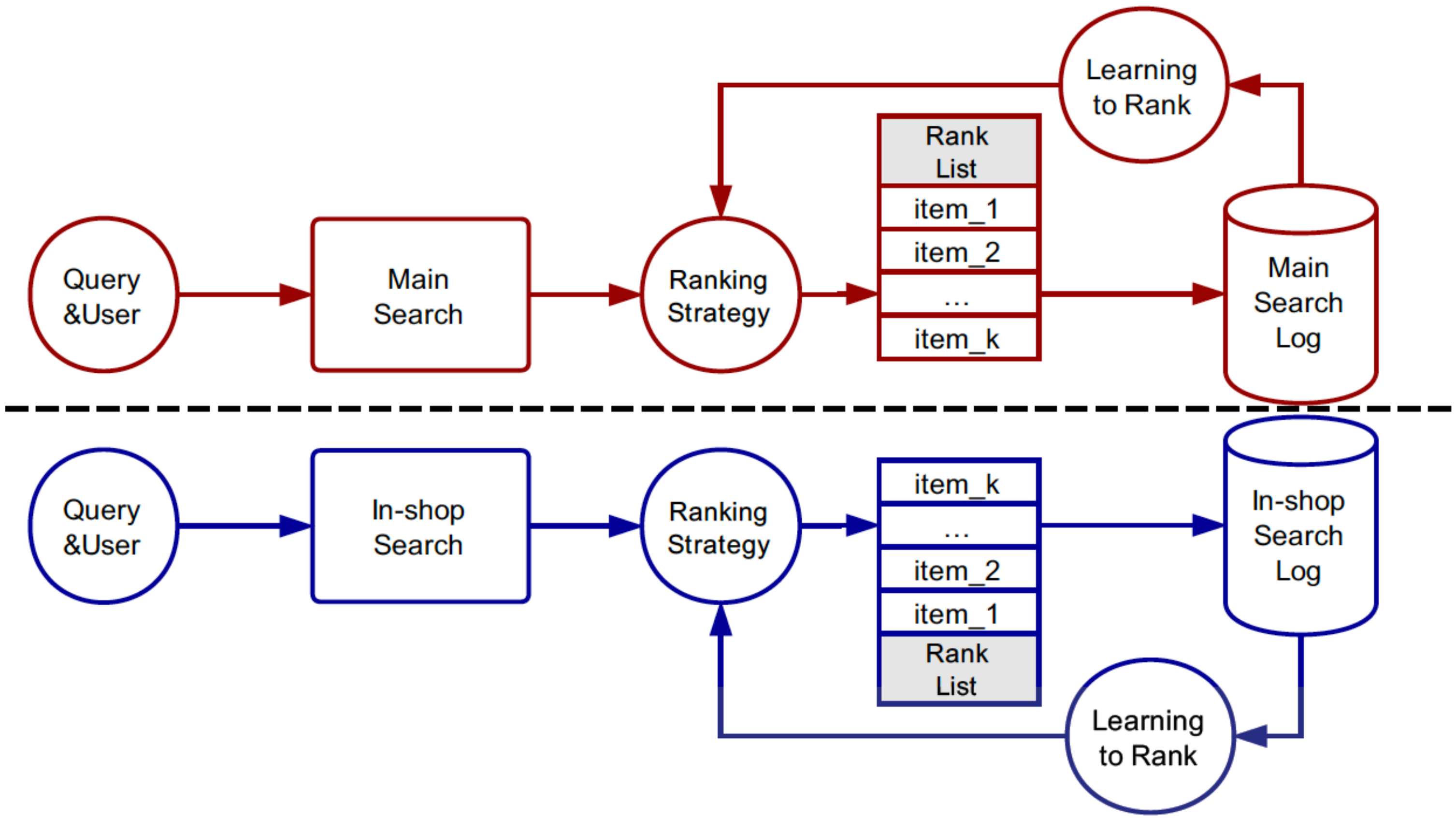
- **Ranking** is a fundamental and widely studied problem
  - Search, advertising and recommendation

Background

- **Multi-scenario Ranking:** most large-scale online platforms or mobile Apps have multiple scenarios

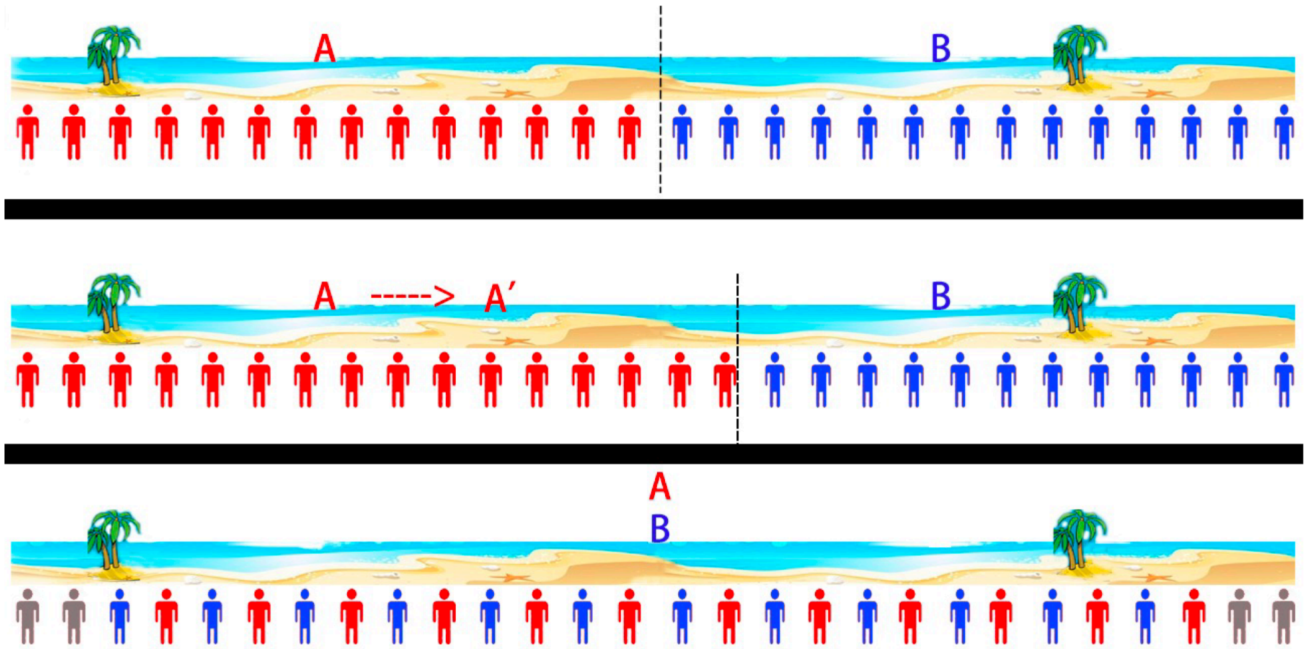
Motivation

- Previous methods separately optimized each individual ranking strategy in each scenario



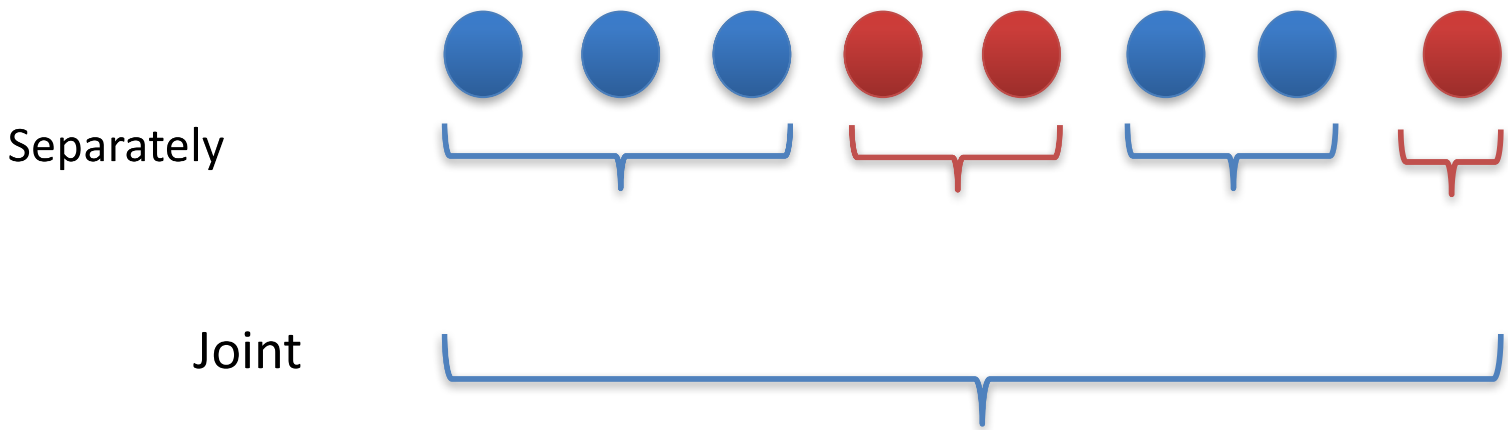
Motivation

- Separately optimization has two main limitations:
  - **Lack of collaboration between scenarios:**
    maximizing one's own objective but ignoring the goals of other strategies leads to a suboptimal overall performance



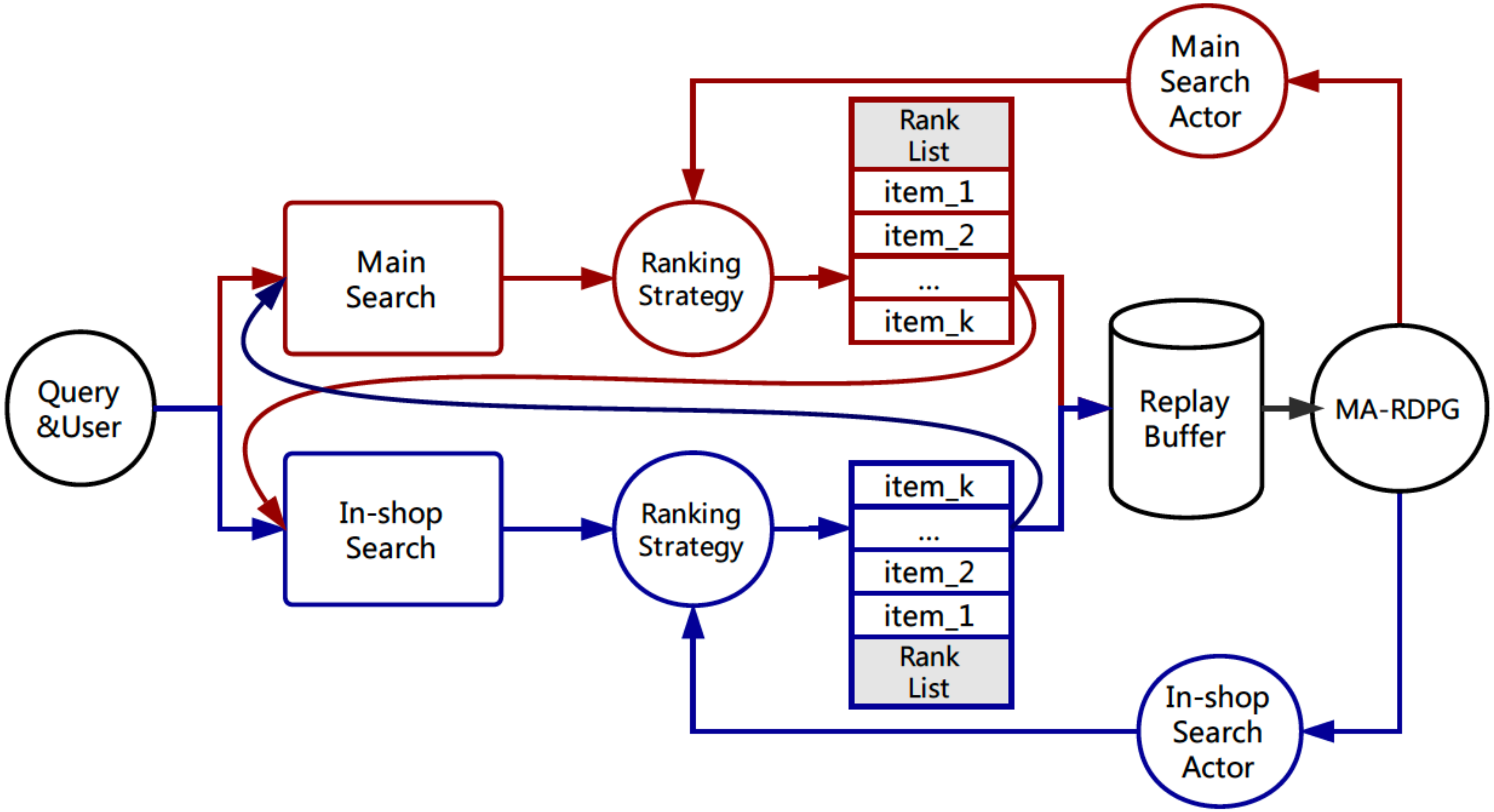
Motivation

- Separately optimization has two main limitations:
  - **Lack of collaboration between scenarios**
  - **Inability to model the correlation between scenarios:** optimization in one scenario only uses its own user data but ignores the context in other scenarios.



Problem Description

- Joint Optimization of Multi-scenario Ranking



Problem Description

- Joint Optimization of Multi-scenario Ranking
  - Multiple ranking strategies for different scenarios in a system
  - Users sequentially interact with the system, and the scenarios sequentially interact with the users
  - Ranking strategies for different scenarios maximize a shared metric
  - Each ranking strategies receive the information of its own scenario

Problem Description

- Joint Optimization of Multi-scenario Ranking
  - Multiple ranking strategies for different scenarios in a system

    Multi-Agent
  - Users sequentially interact with the system, and the scenarios sequentially interact with the users

    Sequential Decision
  - Ranking strategies for different scenarios maximize a shared metric

    Fully Cooperative
  - Each ranking strategies receive the information of its own scenario

    Partially Observable

Problem Description

- Joint Optimization of Multi-scenario Ranking
  - Multiple ranking strategies for different scenarios in a system

Multi-Agent

  - Users sequentially interact with the system, and the scenarios sequentially interact with the users

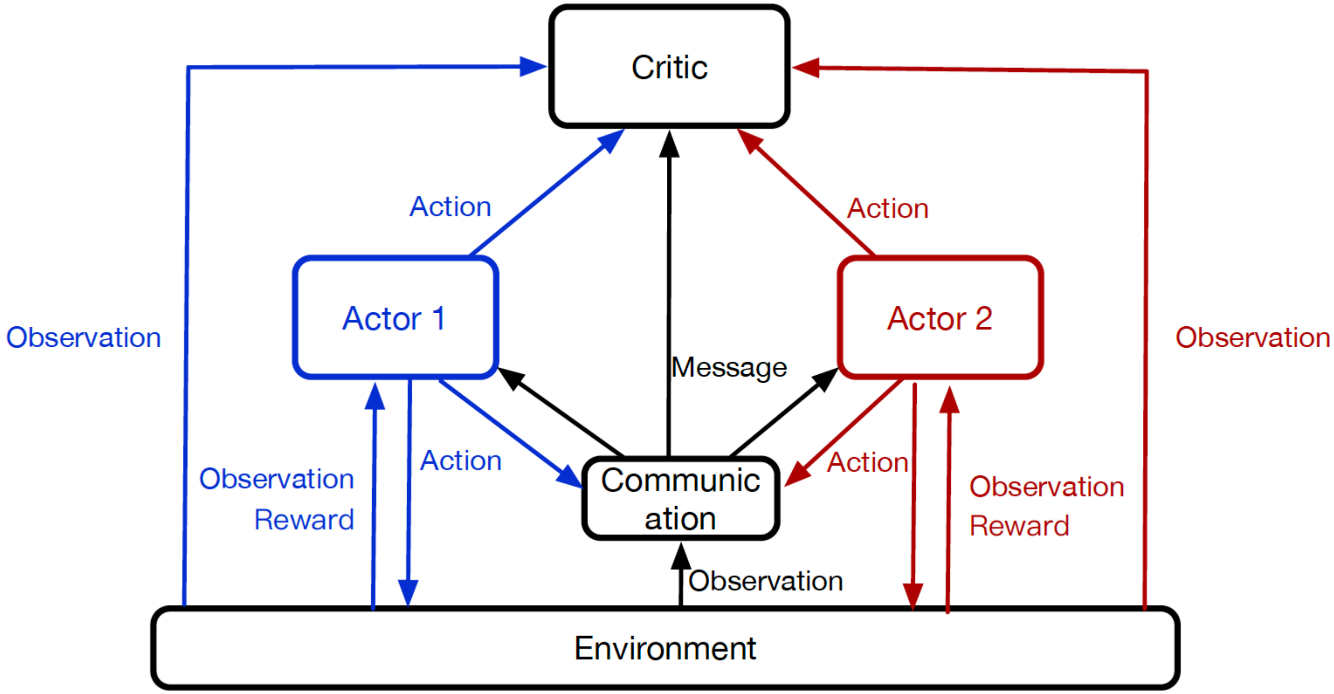
  - a **fully cooperative, partially observable, multi-agent sequential decision** problem

  - Each ranking strategies receive the information of its own scenario

Partially Observable

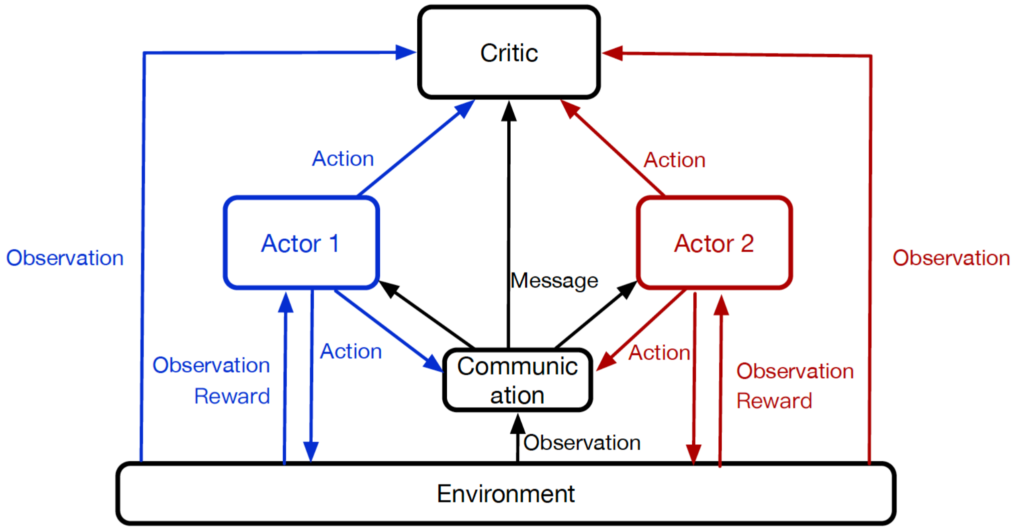
Model Overview

- Multi-Agent Recurrent Deterministic Policy Gradient (**MA-RDPG**)



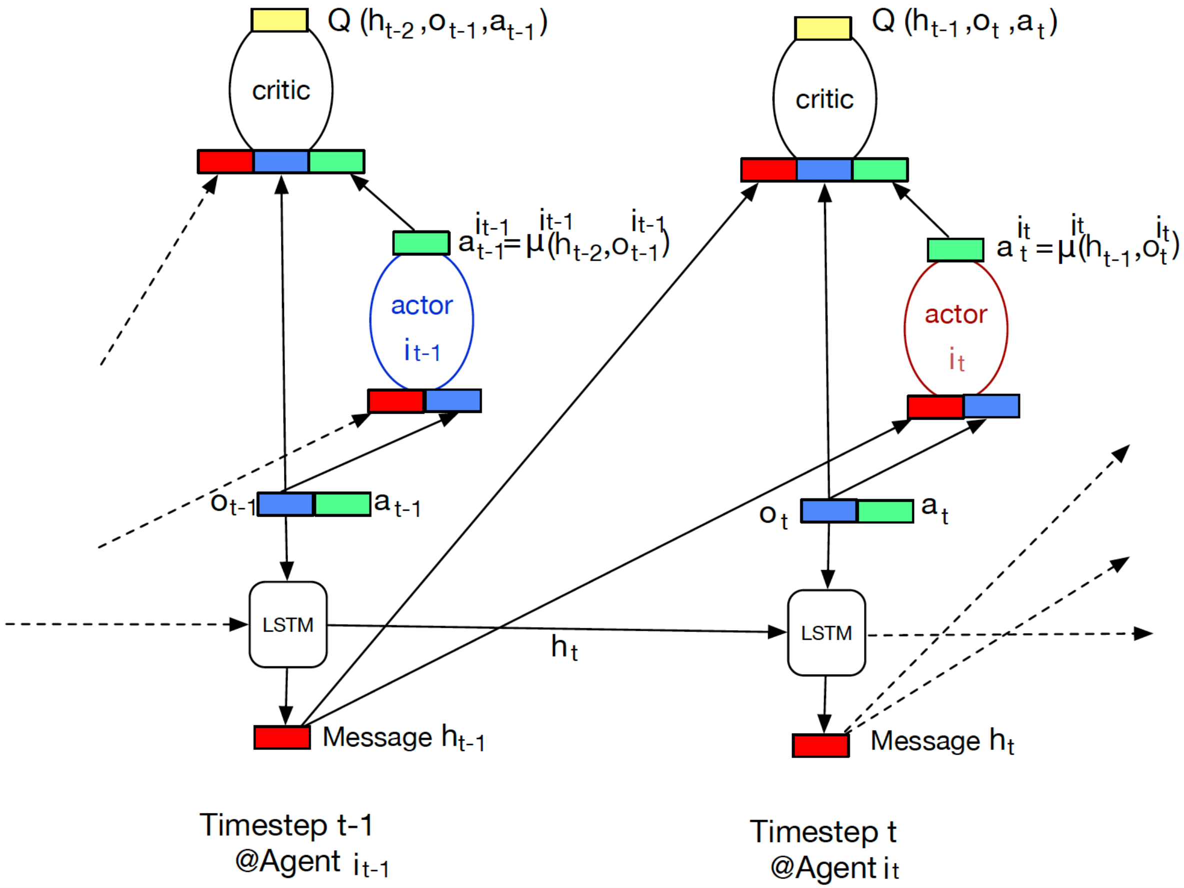
Model Overview

- Multi-Agent Recurrent Deterministic Policy Gradient (**MA-RDPG**)
  - Communication Component
    - partial observation, fully cooperative
  - Private Actor
    - partial observation
  - Centralized Critic
    - fully cooperative



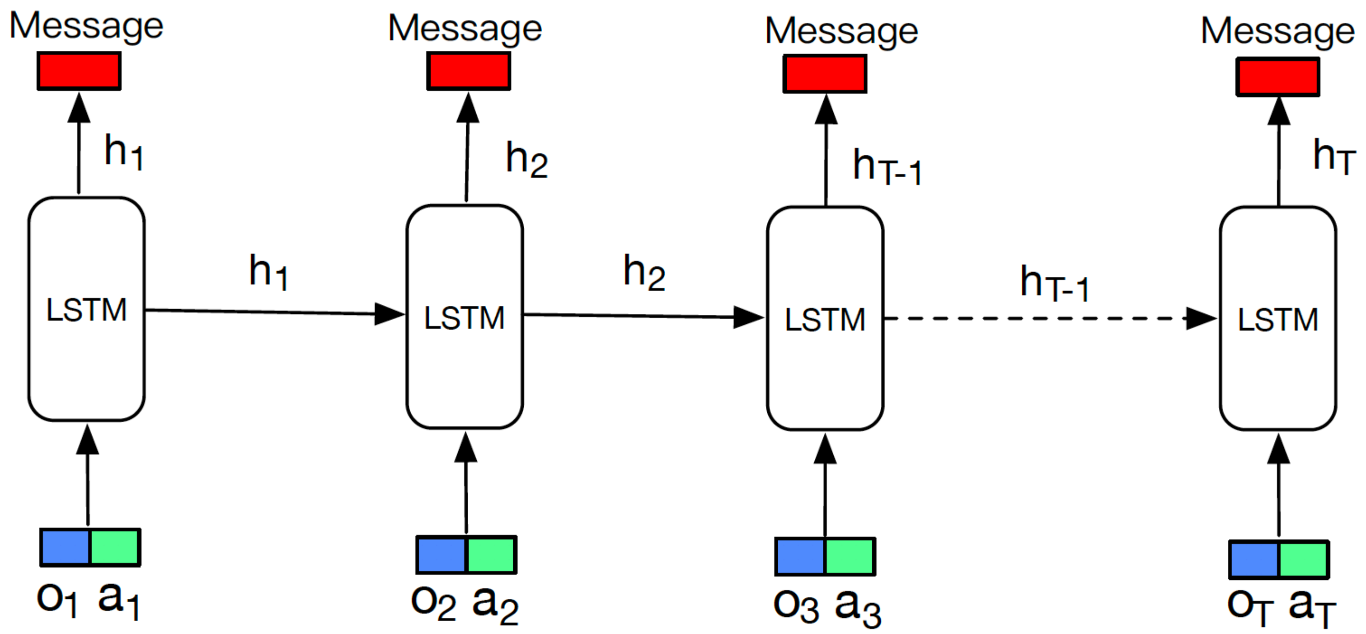
Model Structure

- Multi-Agent Recurrent Deterministic Policy Gradient (**MA-RDPG**)



Model Structure

- **Communication Component**: make the agents collaborate better with each other by sending messages



$$h_{t-1} = LSTM(h_{t-2}, [o_{t-1}; a_{t-1}]; \psi)$$

Model Structure

- **Private Actor**. Each agent has a private actor which receives local observations and shared messages, and makes its own actions.

$$a_t^{i_t} = \mu^{i_t}(s_t; \theta^{i_t}) \approx \mu^{i_t}(h_{t-1}, o_t^{i_t}; \theta^{i_t})$$

- **Centralized Critic**: an action-value function to approximate the future overall rewards obtained by all the agents

$$\begin{aligned} & Q(s_t, a_t^1, a_t^2, \ldots, a_t^N; \phi) \\ = \; & r_t + Q(s_{t+1}, a_{t+1}^1, a_{t+2}^2, \ldots, a_{t+1}^N; \phi) \end{aligned}$$

Training Procedure

- The centralized critic is trained using the Bellman equation

$$L(\phi) = \mathbb{E}_{h_{t-1},o_t}[(Q(h_{t-1}, o_t, a_t; \phi) - y_t)^2]$$

$$y_t = r_t + \gamma Q(h_t, o_{t+1}, \mu^{i_{t+1}}(h_t, o_{t+1}); \phi)$$

- The private actor is updated by maximizing the expected total rewards with respect to the actor's parameters

$$J(\theta^{i_t}) = \mathbb{E}_{h_{t-1},o_t}[Q(h_{t-1}, o_t, a; \phi)|_{a=\mu^{i_t}(h_{t-1},o_t;\theta^{i_t})}]$$

Training Procedure

ALGORITHM 1: MA-RDPG

Initialize the parameters $\theta = \{\theta^1, \ldots, \theta^N\}$ for the N actor networks and ϕ for the centralized critic network.
Initialized the replay buffer R
for *each training step e* **do**
  for *i = 1 to M* **do**
    h_0 = initial message, $t = 1$
    while $t < T$ *and* $o_t \neq$ *terminal* **do**
      Select the action $a_t = \mu^{i_t}(h_{t-1}, o_t) + \mathcal{N}_t$ for the active agent i_t with an exploration noise
      Receive reward r_t and the new observation o_{t+1}
      Generate the message $h_t = LSTM(h_{t-1}, [o_t; a_t])$
      $t = t + 1$
    end
    Store episode $\{h_0, o_1, a_1, r_1, h_1, o_2, r_2, h3, o3, \ldots\}$ in R
  end
  Sample a random minibatch of episodes B from replay buffer R
  foreach *episode in B* **do**
    for *t = T downto 1* **do**
      Update the critic by minimizing the loss:
      $L(\phi) = (Q(h_{t-1}, o_t, a_t; \phi) - y_t)^2$, where
      $y_t = r_t + \gamma Q(h_t, o_{t+1}, \mu^{i_{t+1}}(h_t, o_{t+1}); \phi)$
      Update the i_t-th actor by maximizing the critic:
      $J(\theta^{i_t}) = Q(h_{t-1}, o_t, a; \phi)|_{a=\mu^{i_t}(h_{t-1}, o_t; \theta^{i_t})}$
      Update the communication component.
    end
  end
end

→ Generate new episode

→ Update the replay buffer

→ Sample training batch from replay buffer

→ Update the parameters of:

- Centralized Critic
- Private actor
- Communication Component

Application in Search

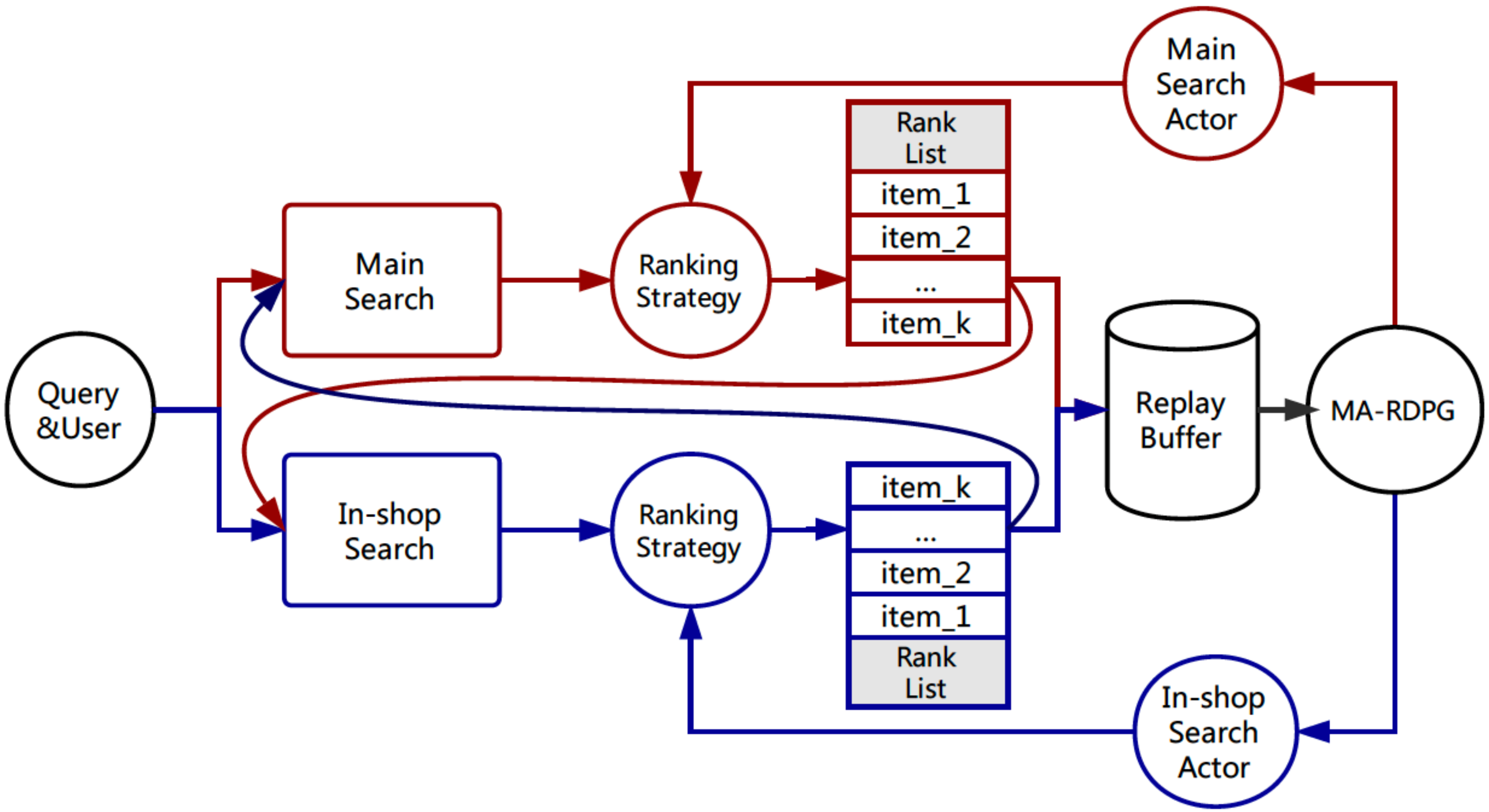
- We apply MA-RDPG to jointly optimize the ranking strategies in two search scenarios in Taobao

Main Search

In-shop Search

Application in Search

- We apply MA-RDPG to jointly optimize the ranking strategies in two search scenarios in Taobao



How Training Happens

- **Step 1**: Start from a base ranking algorithm
- **Step 2**: Collect user feedback data with the current ranking system
- **Step 3**: Train our MA-RDPG algorithm to obtain new ranking weights (i.e., the action of the agents by deterministic policy)
- **Step 4**: Apply the new weights to the online ranking systems
- **Goto Step 2** until convergence

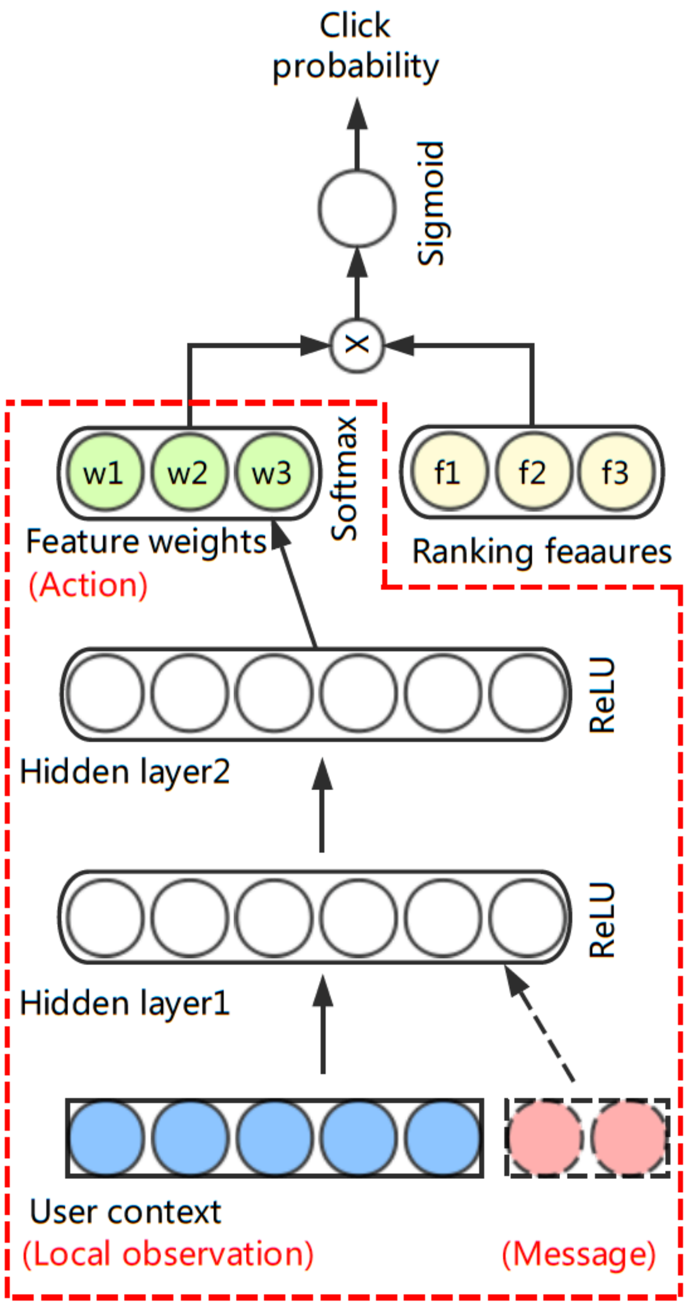
Application in Search

- The observations, actions, rewards for the agents:
  - **Observations**: the features of each ranking scenarios
    - **the attributes of the customer** (age, gender, purchasing power, etc.)
    - **the properties of the customer's clicked items** (price, conversion rate, sales volume, etc.)
    - **the query type and the scenario index** (main or in-shop search)

Application in Search

- The observations, actions, rewards for the agents:
  - **Actions**: the **weight vector** for the ranking features
  - **Continuous actions, deterministic policies**

$$a_t^{i_t} = \mu^{i_t}(s_t; \theta^{i_t}) \approx \mu^{i_t}(h_{t-1}, o_t^{i_t}; \theta^{i_t})$$



Application in Search

- The observations, actions, rewards for the agents:
  - Rewards: user feedback on the presented product list
    - if a purchase behavior happens, **reward = the price of the bought product**
    - if a click happens, **reward = 1**
    - if there is no purchase nor click, **reward = -1**
    - if a user leaves the page without buying any product, **reward = −5**.

Experiment Results

- We deploy our MA-RDPG online in Taobao
- We choose three baselines
  - EW (Empirical Weight) + L2R (**Learning to rank, a strong model previously used by Taobao**)
  - L2R+EW
  - L2R+L2R

Experiment Results

- GMV gap evaluated on an online Taobao platform

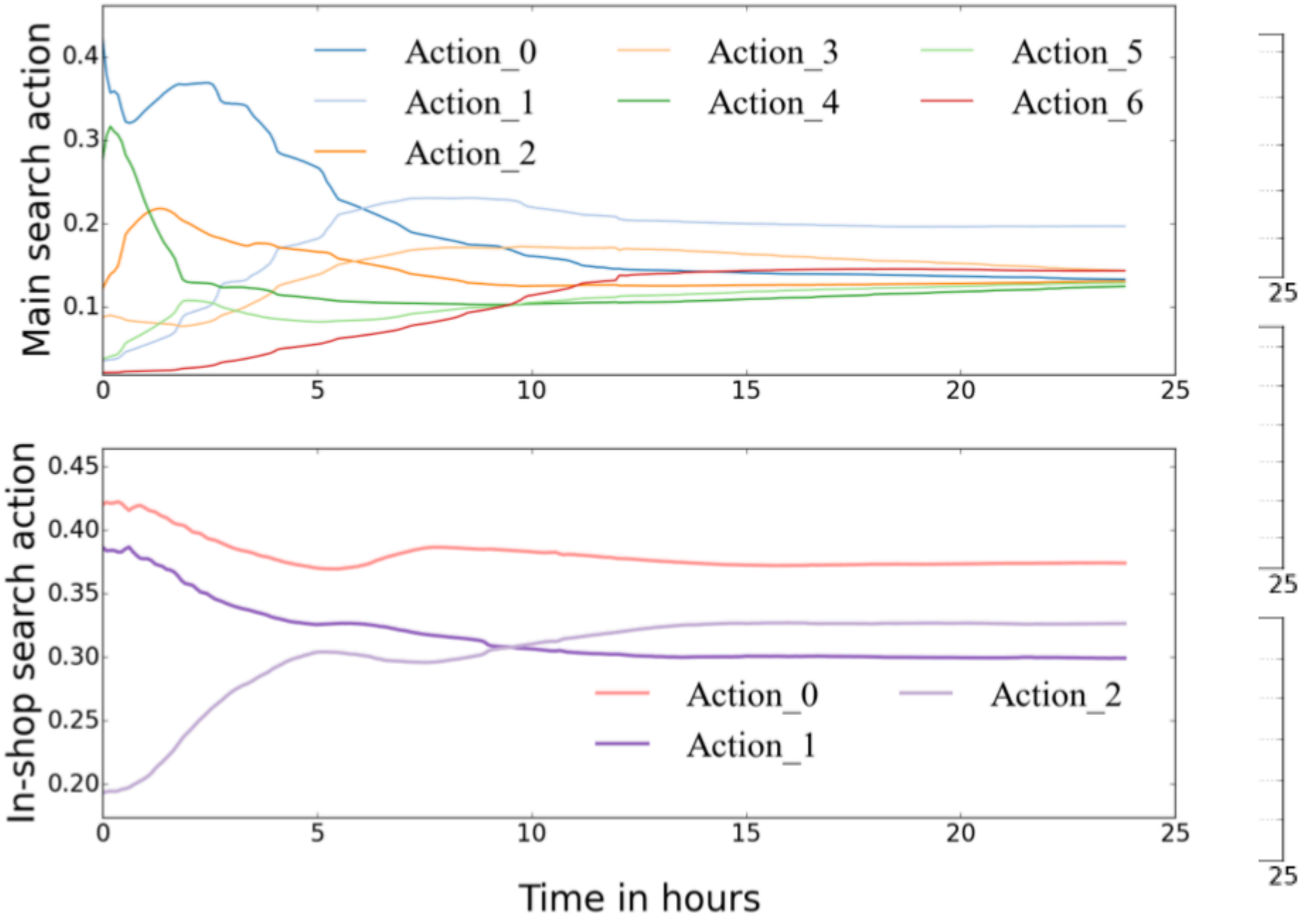
Relative improvement against EW+EW

day	EW + L2R			L2R + EW			L2R + L2R			MA-RDPG		
	main	in-shop	total	main	in-shop	total	main	in-shop	total	main	in-shop	total
1	0.04%	1.78%	0.58%	5.07%	-1.49%	3.04%	5.22%	0.78%	3.84%	5.37%	2.39%	4.45%
2	0.01%	1.98%	0.62%	4.96%	-0.86%	3.16%	4.82%	1.02%	3.64%	5.54%	2.53%	4.61%
3	0.08%	2.11%	0.71%	4.82%	-1.39%	2.89%	5.02%	0.89%	3.74%	5.29%	2.83%	4.53%
4	0.09%	1.89%	0.64%	5.12%	-1.07%	3.20%	5.19%	0.52%	3.74%	5.60%	2.67%	4.69%
5	-0.08%	2.24%	0.64%	4.88%	-1.15%	3.01%	4.77%	0.93%	3.58%	5.29%	2.50%	4.43%
6	0.14%	2.23%	0.79%	5.07%	-0.94%	3.21%	4.86%	0.82%	3.61%	5.59%	2.37%	4.59%
7	-0.06%	2.12%	0.62%	5.21%	-1.32%	3.19%	5.14%	1.16%	3.91%	5.30%	2.69%	4.49%
avg.	0.03%	2.05%	0.66%	5.02%	-1.17%	3.09%	5.00%	0.87%	3.72%	5.43%	2.57%	4.54%

Recent results online: MA-RDPG gains 3% improvement against L2R+L2R

Experiment Results

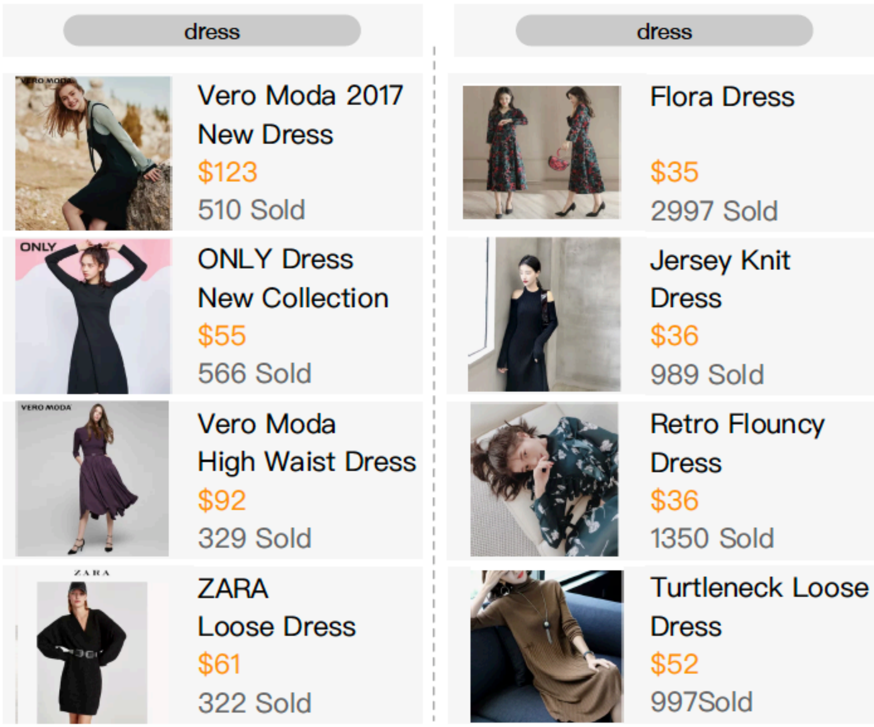
- Learning process of the loss function, critic value and GMV gap



Experiment Results

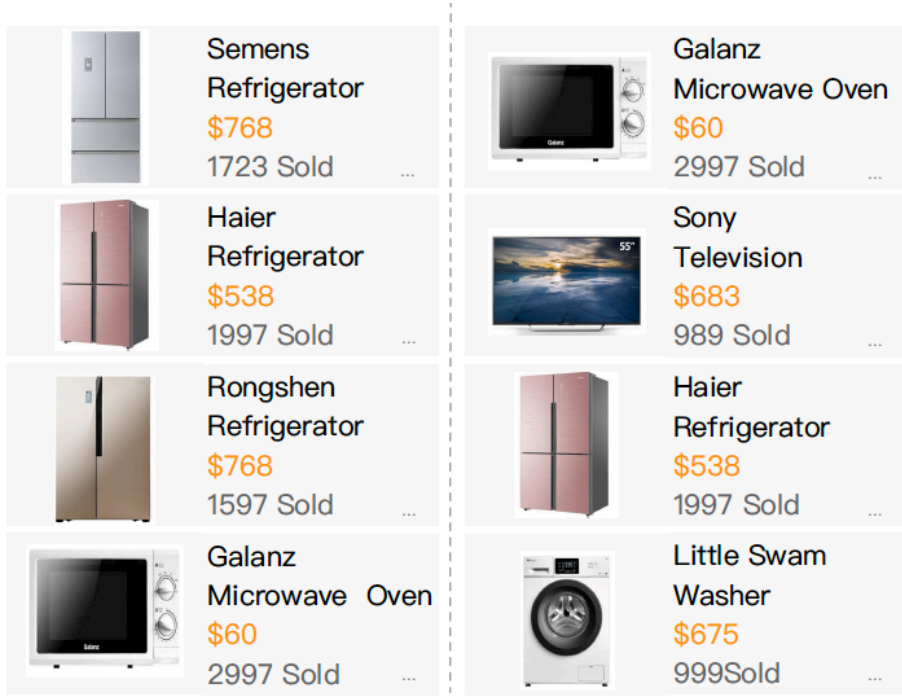
- Case Study

L2R+L2R | MA-DPRG



Main Search Results

L2R+L2R | MA-DPRG



In-shop Search Results

Summary

- Multi-scenario ranking (or optimization) as a **fully cooperative, partially observable, multi-agent sequential decision** problem
- **Multi-agent, deterministic policy** RL to enable multiple agents to work collaboratively to optimize the overall performance.
- **Significant gain** in improving ranking systems in real online service (Taobao)
- **Learning from user feedback, through interactions!**

Messages and Lessons

- **Keys** to the success of RL in NLP
  - Formulate a task as a **natural sequential decision** problem where current decisions affect future ones!
  - Remember the **nature** of **trial-and-error** when you have no access to full, strong supervision.
  - Encode the **expertise** or **prior knowledge** of the task in rewards.
  - Applicable in many **weak supervision** settings.

Messages and Lessons

- **Lessons** we learned
  - A **warm-start** is important, using pre-training (due to too many spurious solutions and too sparse rewards)
  - Very **marginal** improvements to full supervision settings
  - Very **marginal** improvements for large action space problems (e.g., language generation)
  - Patient enough to the **training tricks and tunings**

Thanks for attention!

Language Generation: Dialogue as an Example

Minlie Huang

Tsinghua University

Thanks for Your Attention

- Minlie Huang, Tsinghua University
- aihuang@tsinghua.edu.cn
- http://aihuang.org/p
- Recruiting post-doctors!