DRL4NLP:

Deep Reinforcement Learning for Natural Language Processing

William Wang
UC Santa Barbara

Jiwei Li
Shannon.ai

Xiaodong He
JD AI Research

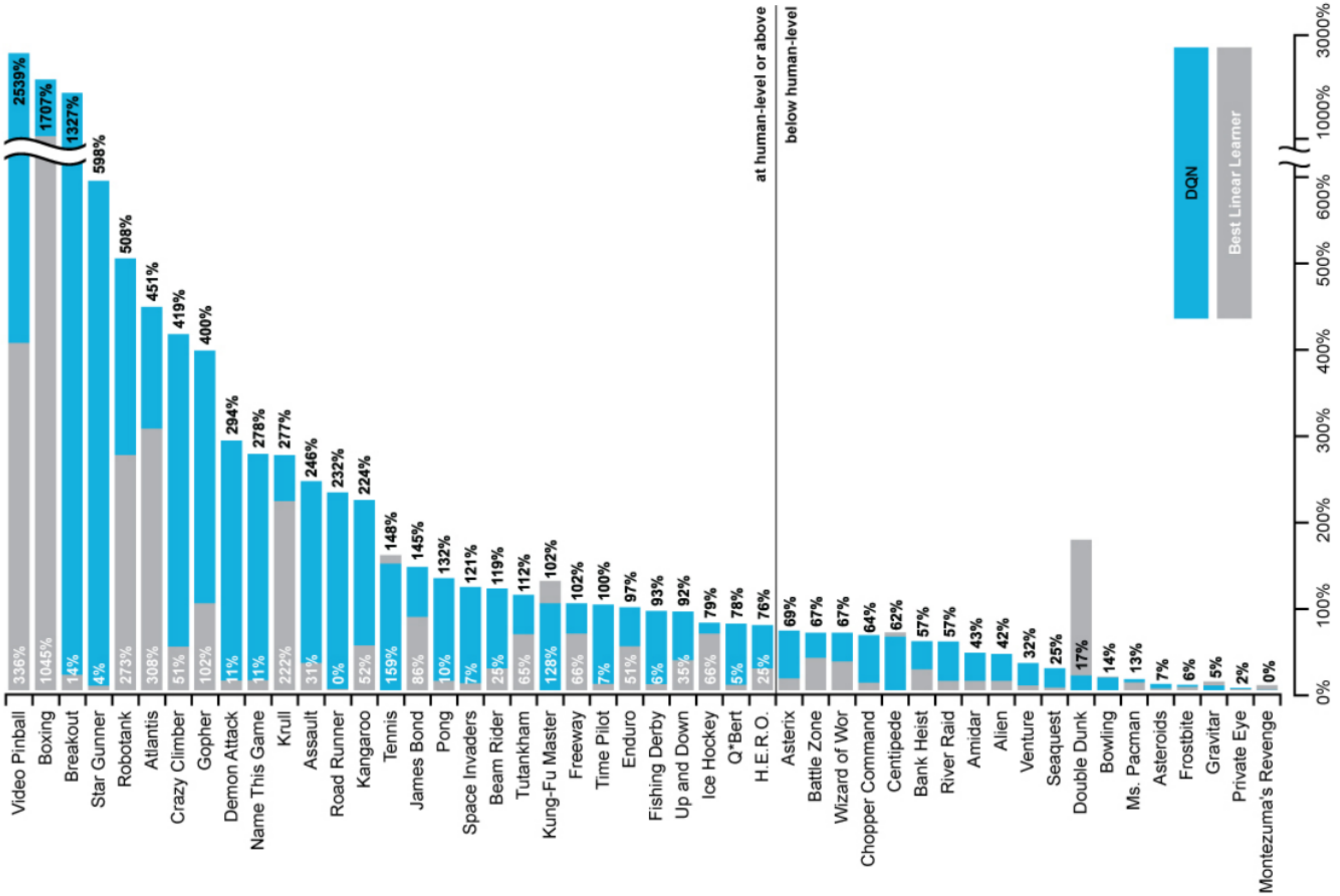
Tutorial Outline

- **Introduction**
- Fundamentals and Overview (William Wang)
- Deep Reinforcement Learning for Dialog (Jiwei Li)
- Challenges (Xiaodong He)
- Conclusion

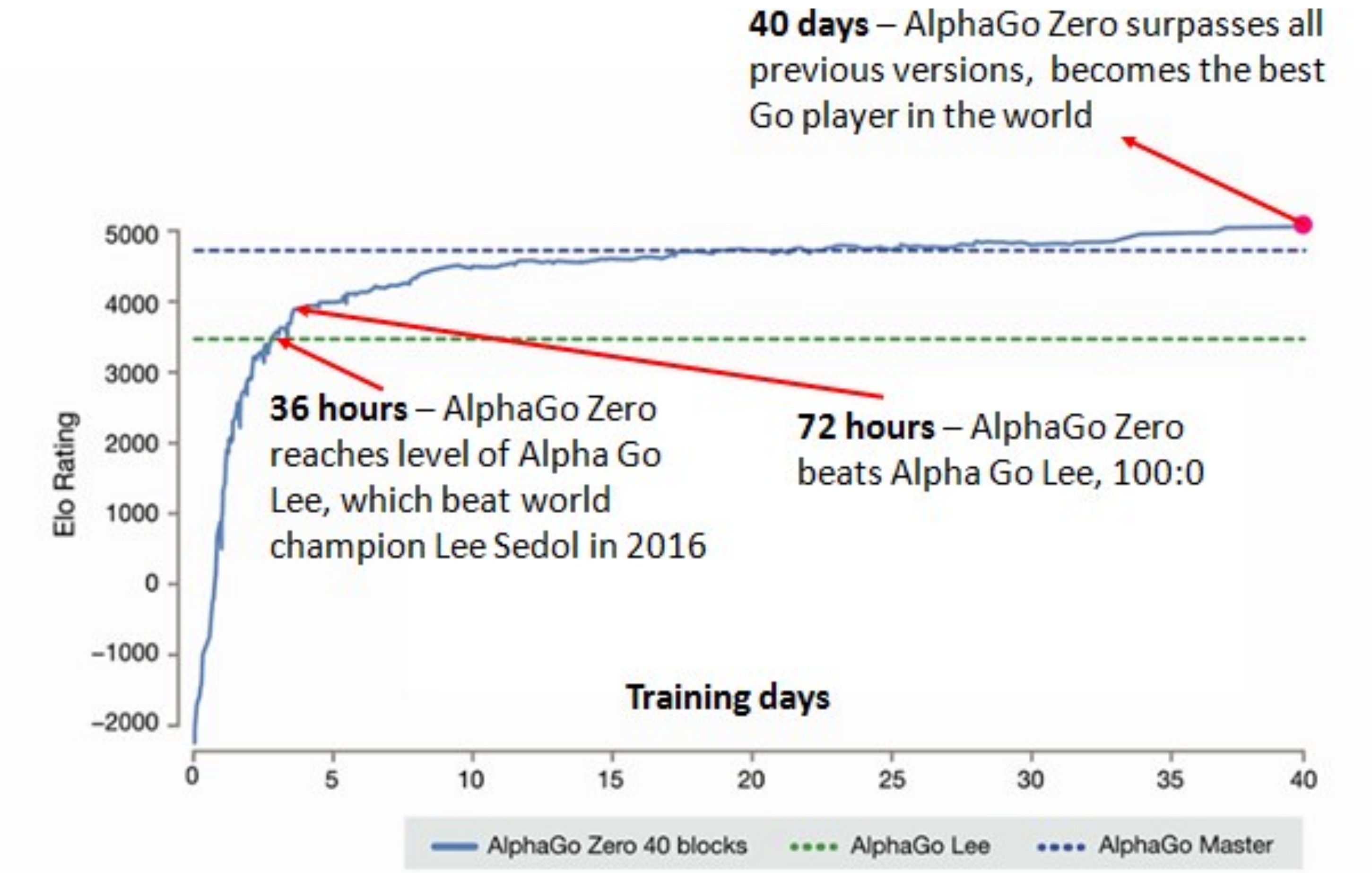
Introduction

DRL for Atari Games (Mnih et al., 2015)

nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
LEARNING CURVE
Self-taught AI software attains human-level performance in video games
SHARE DATA IN OUTBREAKS
A GIANT IN THE EARLY UNIVERSE
TELEPORTATION FOR TWO



AlphaGo Zero (Oct., 2017)



40 days – AlphaGo Zero surpasses all previous versions, becomes the best Go player in the world

36 hours – AlphaGo Zero reaches level of Alpha Go Lee, which beat world champion Lee Sedol in 2016

72 hours – AlphaGo Zero beats Alpha Go Lee, 100:0

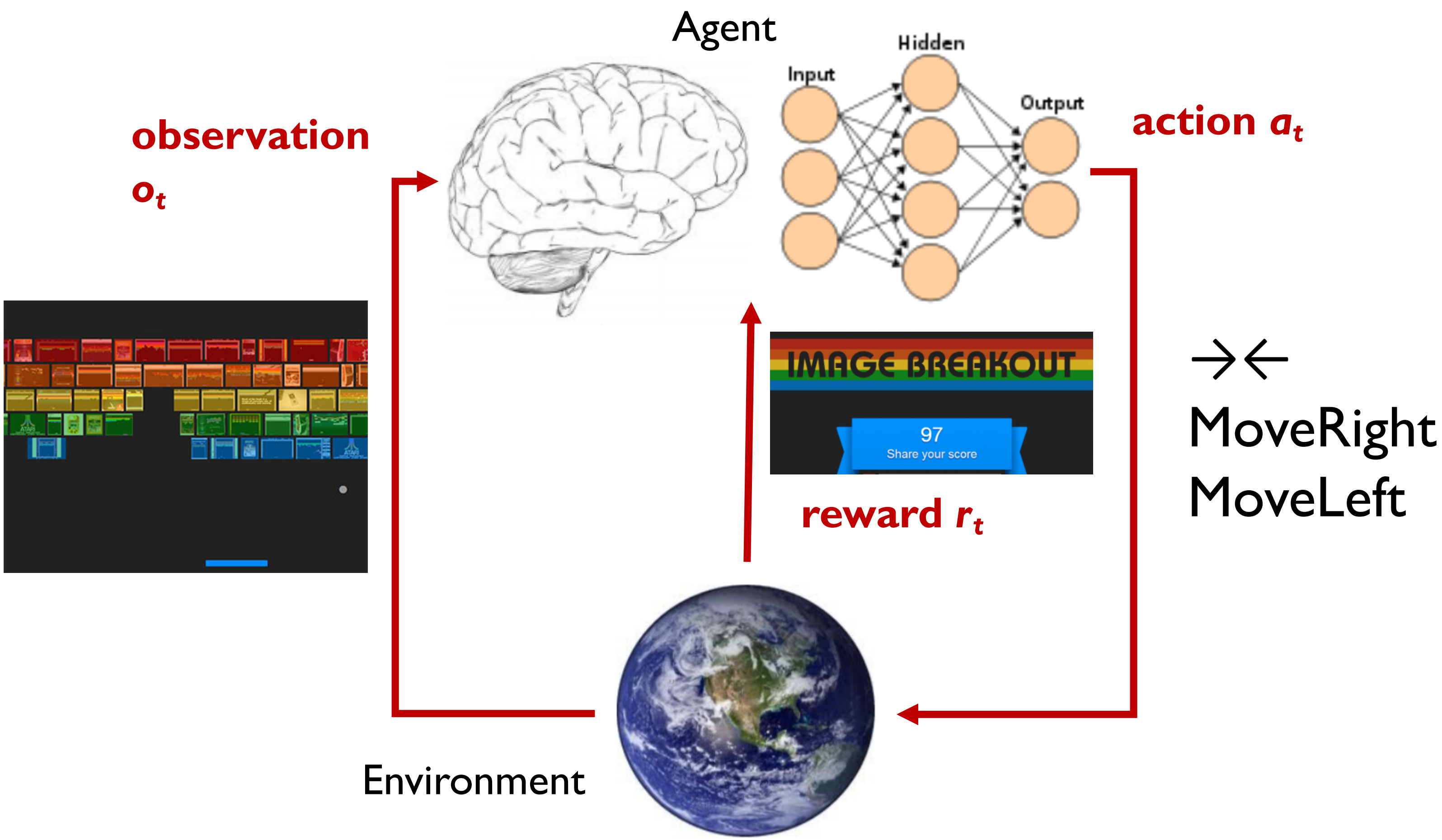
Reinforcement Learning

- RL is a general purpose framework for **decision making**
    - RL is for an *agent* with the capacity to *act*
    - Each *action* influences the agent's future *state*
    - Success is measured by a scalar *reward* signal

Big three: action, state, reward

Some slides adapted from David Silver.

Agent and Environment



Agent

observation o_t

action a_t

reward r_t

→←

MoveRight

MoveLeft

Environment

Some slides adapted from David Silver.

Major Components in an RL Agent

- An RL agent may include one or more of these components
  - **Policy**: agent's behavior function
  - **Value function**: how good is each state and/or action
  - **Model**: agent's representation of the environment

Some slides adapted from David Silver.

Reinforcement Learning Approach

- Policy-based RL
  - Search directly for optimal policy π^*

π^* is the policy achieving maximum future reward

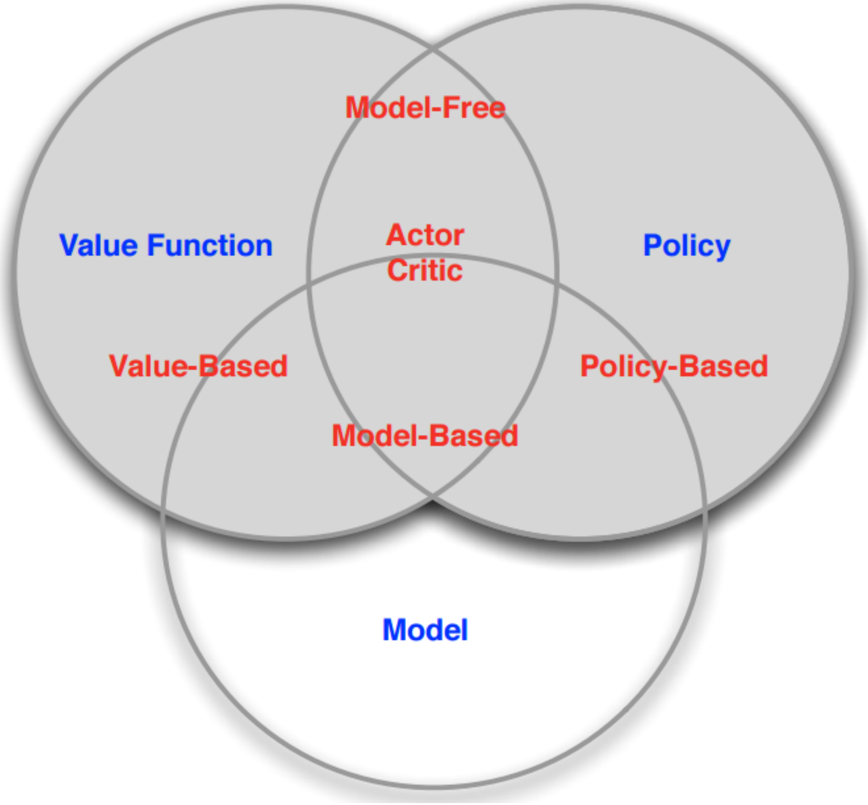
- Value-based RL
  - Estimate the optimal value function $Q^*(s, a)$

$Q^*(s, a)$ is maximum value achievable under any policy

- Model-based RL
  - Build a model of the environment
  - Plan (e.g. by lookahead) using model

Some slides adapted from David Silver.

RL Agent Taxonomy



Some slides adapted from David Silver.

Deep Reinforcement Learning

- Idea: deep learning for reinforcement learning
  - Use deep neural networks to represent
    - Value function
    - Policy
    - Model
  - Optimize loss function by SGD

Some slides adapted from David Silver.

Value-Based Deep RL

Estimate How Good Each State and/or Action is

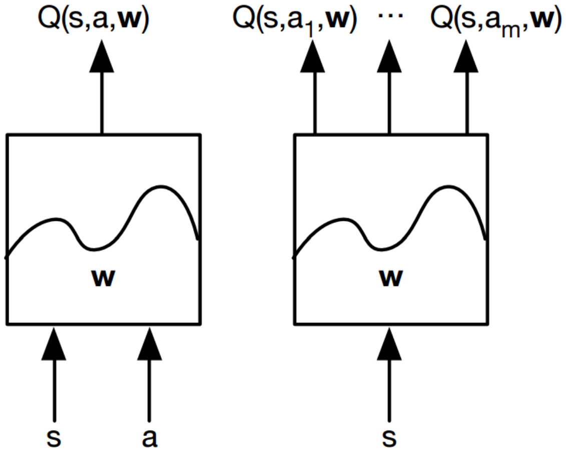
Some slides adapted from David Silver.

Value Function Approximation

- Value functions are represented by a *lookup table*

$$Q(s, a) \;\; \forall s, a$$

  - too many states and/or actions to store
  - not able to learn the value of each entry individually
- Values can be estimated with *function approximation*



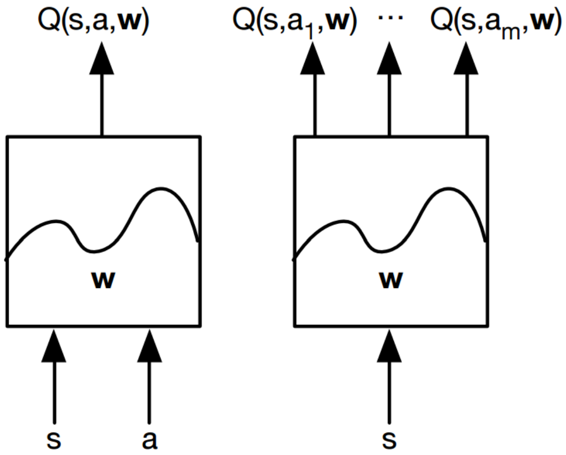
Some slides adapted from David Silver.

Q-Networks

- Q-networks represent value functions with weights w

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

  - generalize from seen states to unseen states
  - update parameter w for function approximation



Some slides adapted from David Silver.

Q-Learning

- Goal: estimate optimal Q-values
  - Optimal Q-values obey a Bellman equation

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

  - *Value iteration* algorithms solve the Bellman equation

$$Q_{i+1}(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q_i(s',a') \mid s,a\right]$$

Some slides adapted from David Silver.

Deep Q-Networks (DQN)

- Represent value function by deep Q-network with weights w

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

- Objective is to minimize MSE loss by SGD
  - Starts with initial state s, takes a, gets r, and sees s'
  - but we want w to give us better Q early in the game.

$$L(w) = \mathbb{E}\left[\left(r + \gamma \max_{a'} Q(s', a', w) - Q(s, a, w)\right)^2\right]$$

- Leading to the following Q-learning gradient

$$\frac{\partial L(w)}{\partial w} = \mathbb{E}\left[\left(r + \gamma \max_{a'} Q(s', a', w) - Q(s, a, w)\right) \frac{\partial Q(s, a, w)}{\partial w}\right]$$

Some slides adapted from David Silver.

Stability Issues with Deep RL

- Naive Q-learning oscillates or diverges with neural nets

1. Data is sequential
   - Successive samples are correlated, non-iid (independent and identically distributed)
2. Policy changes rapidly with slight changes to Q-values
   - Policy may oscillate
   - Distribution of data can swing from one extreme to another
3. Scale of rewards and Q-values is unknown
   - Naive Q-learning gradients can be unstable when backpropagated

Some slides adapted from David Silver.

Stable Solutions for DQN

- DQN provides a stable solutions to deep value-based RL

1. Use experience replay
    - Break correlations in data, bring us back to iid setting
    - Learn from all past policies
2. Freeze target Q-network
    - Avoid oscillation
    - Break correlations between Q-network and target
3. Clip rewards or normalize network adaptively to sensible range
    - Robust gradients

Some slides adapted from David Silver.

Policy-Based Deep RL

Estimate How Good An Agent’s Behavior is

Some slides adapted from David Silver.

Deep Policy Networks

- Represent policy by deep network with weights u

$$a = \pi(a \mid s, u) \qquad a = \pi(s, u)$$

stochastic policy   deterministic policy

- Objective is to maximize total discounted reward by SGD

$$L(u) = \mathbb{E}\left[r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots \mid \pi(\cdot, u)\right]$$

Some slides adapted from David Silver.

Policy Gradient

- The gradient of a stochastic policy $\pi(a \mid s, u)$ is given by

$$\frac{\partial L(u)}{\partial u} = \mathbb{E}_s \left[\frac{\partial \log \pi(a \mid s, u)}{\partial u} Q^{\pi}(s, a) \right]$$

- The gradient of a deterministic policy $\pi(s, u)$ is given by

$$\frac{\partial L(u)}{\partial u} = \mathbb{E}_s \left[\frac{\partial Q^{\pi}(s, a)}{\partial a} \frac{\partial a}{\partial u} \right] \qquad a = \pi(s, u)$$

Some slides adapted from David Silver.

Actor-Critic (Value-Based + Policy-Based)

- Estimate value function $Q(s, a, w) \approx Q^\pi(s, a)$
- Update policy parameters u by SGD
  - Stochastic policy

$$\frac{\partial L(u)}{\partial u} = \mathbb{E}_s \left[\frac{\partial \log \pi(a \mid s, u)}{\partial u} Q(s, a, w) \right]$$

  - Deterministic policy

$$\frac{\partial L(u)}{\partial u} = \mathbb{E}_s \left[\frac{\partial Q(s, a, w)}{\partial a} \frac{\partial a}{\partial u} \right]$$

Some slides adapted from David Silver.

Reinforcement Learning in Action

DRL4NLP: Overview of Applications

- Information Extraction
  - Narasimhan et al., EMNLP 2016
- Relational Reasoning
  - DeepPath (Xiong et al., EMNLP 2017)
- Sequence Learning
  - MIXER (Ranzato et al., ICLR 2016)
- Text Classification
  - Learning to Active Learn (Fang et al., EMNLP 2017)
  - Reinforced Co-Training (Wu et al., NAACL 2018)
  - Relation Classification (Qin et al., ACL 2018)

DRL4NLP: Overview of Applications

- Coreference Resolution
  - Clark and Manning (EMNLP 2016)
  - Yin et al., (ACL 2018)
- Summarization
  - Paulus et al., (ICLR 2018)
  - Celikyilmaz et al., (ACL 2018)
- Language and Vision
  - Video Captioning (Wang et al., CVPR 2018)
  - Visual-Language Navigation (Xiong et al., IJCAI 2018)
  - Model-Free + Model-Based RL (Wang et al., ECCV 2018)

Tutorial Outline

- Introduction
- **Fundamentals and Overview** (William Wang)
- Deep Reinforcement Learning for Dialog (Jiwei Li)
- Challenges (Xiaodong He)
- Conclusion

Fundamentals and Overview

- Why DRL4NLP?
- Important Directions of DRL4NLP.

Why does one use (D)RL in NLP?

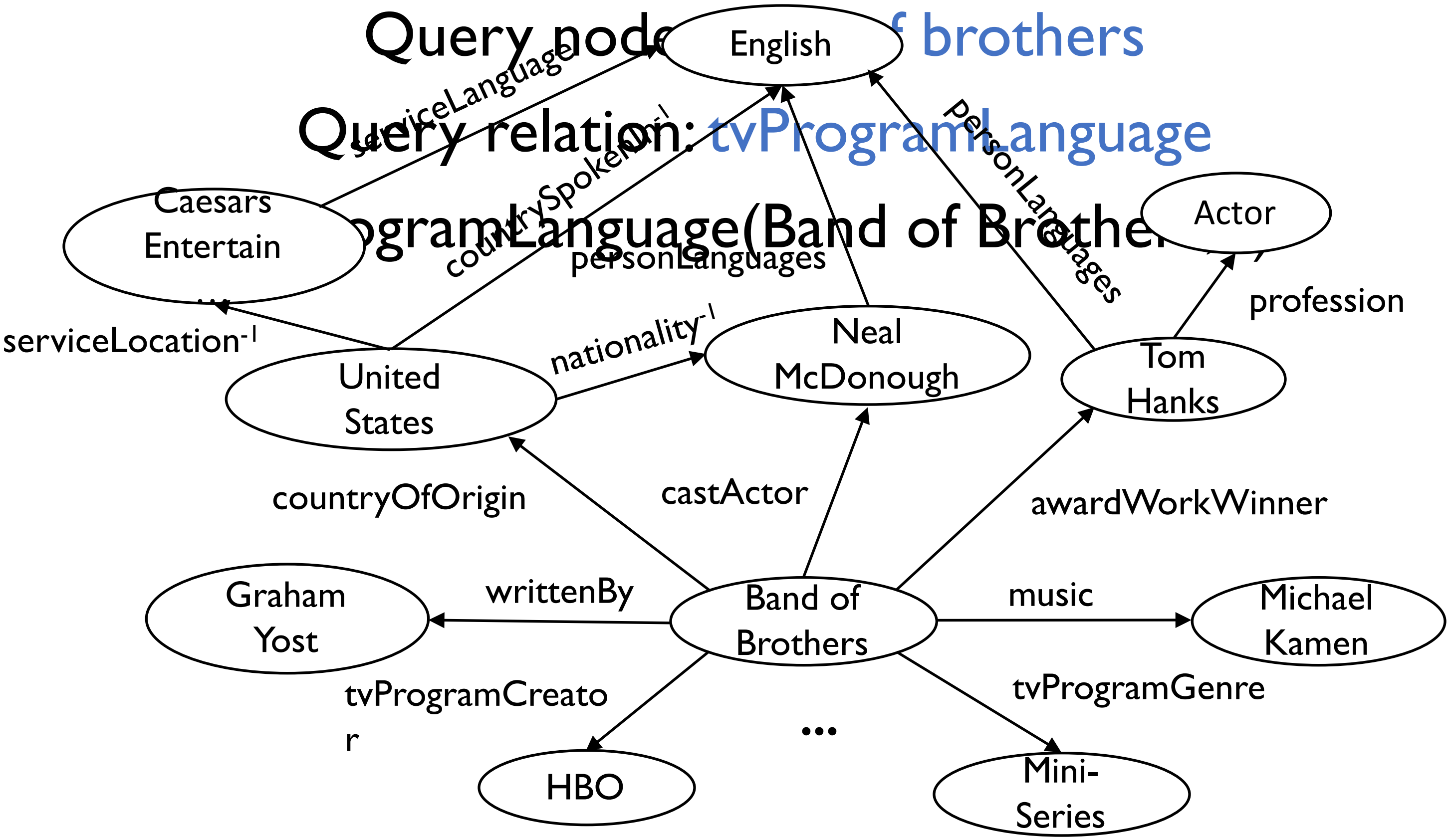
1. Learning to search and reason.
2. Instead of minimizing the surrogate loss (e.g., XE, hinge loss), optimize the end metric (e.g., BLEU, ROUGE) directly.
3. Select the right (unlabeled) data.
4. Back-propagate the reward to update the model.

Learning to Search and Reason

SEARN (Daume III et al., 2009): Learning to Search

1. use a good initial policy at training time to produce a sequence of actions (e.g., the choice of the next word)
2. a search algorithm is run to determine the optimal action at each time step
3. a new classifier (a.k.a. policy) is trained to predict that action

Reasoning on Knowledge Graph

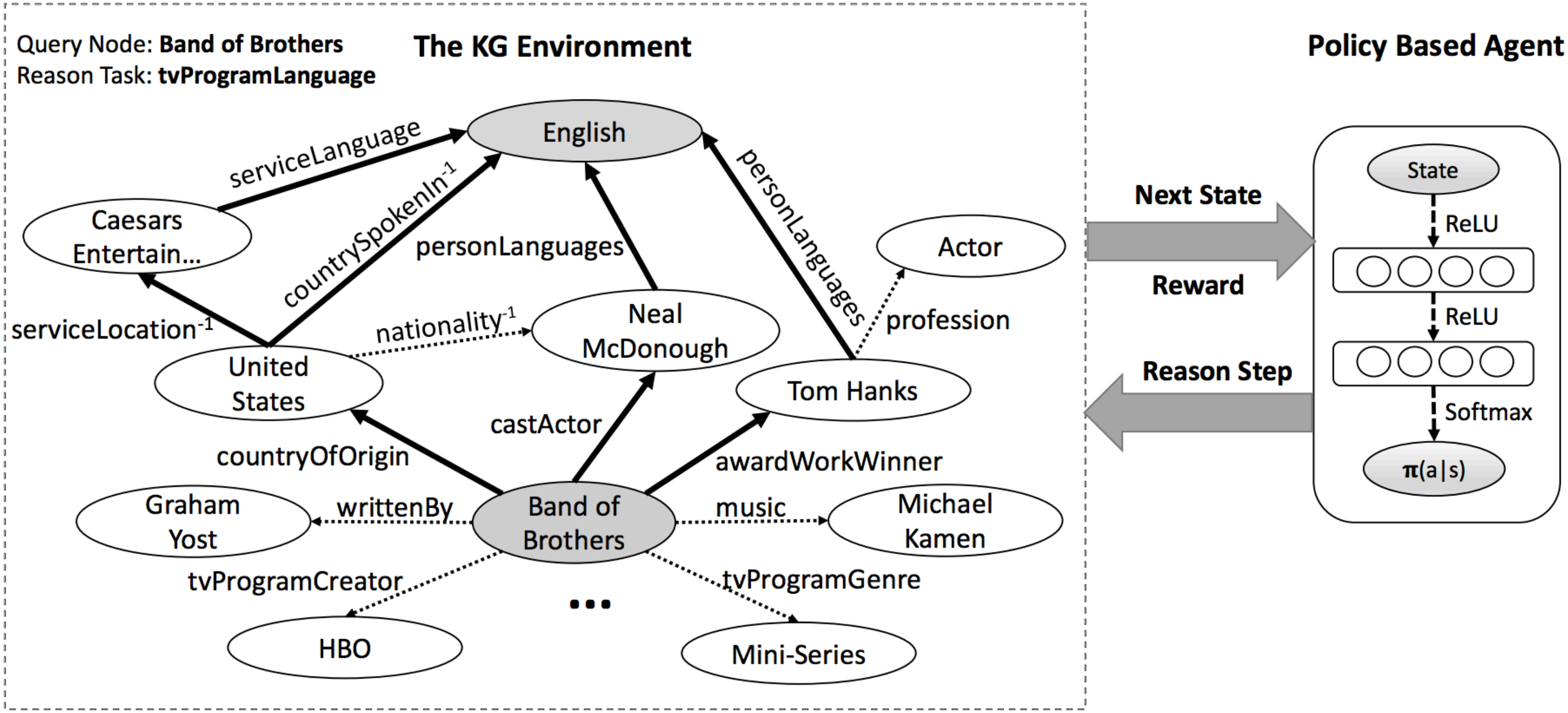


Query node f brothers

Query relation: tvProgramLanguage

ogramLanguage(Band of Brothe

DeepPath: DRL for KG Reasoning (Xiong et al., EMNLP 2017)



Components of MDP

- Markov decision process $< S, A, P, R >$
  - S: continuous states represented with embeddings
  - A: action space (relations)
  - $P(S_{t+1} = s' | S_t = s, A_t = a)$: transition probability
  - $R(s, a)$: reward received for each taken step

- With pretrained KG embeddings
  - $s_t = e_t \oplus (e_{target} - e_t)$
  - $A = \{r_1, r_2, \dots, r_n\}$, all relations in the KG

Reward Functions

- Global Accuracy

$$r_{\text{GLOBAL}} = \begin{cases} +1, & \text{if the path reaches } e_{target} \\ -1, & \text{otherwise} \end{cases}$$

- Path Efficiency

$$r_{\text{EFFICIENCY}} = \frac{1}{length(p)}$$

- Path Diversity

$$r_{\text{DIVERSITY}} = -\frac{1}{|F|} \sum_{i=1}^{|F|} cos(\mathbf{p}, \mathbf{p}_i)$$

Training with Policy Gradient

- Monte-Carlo Policy Gradient (REINFORCE, William, 1992)

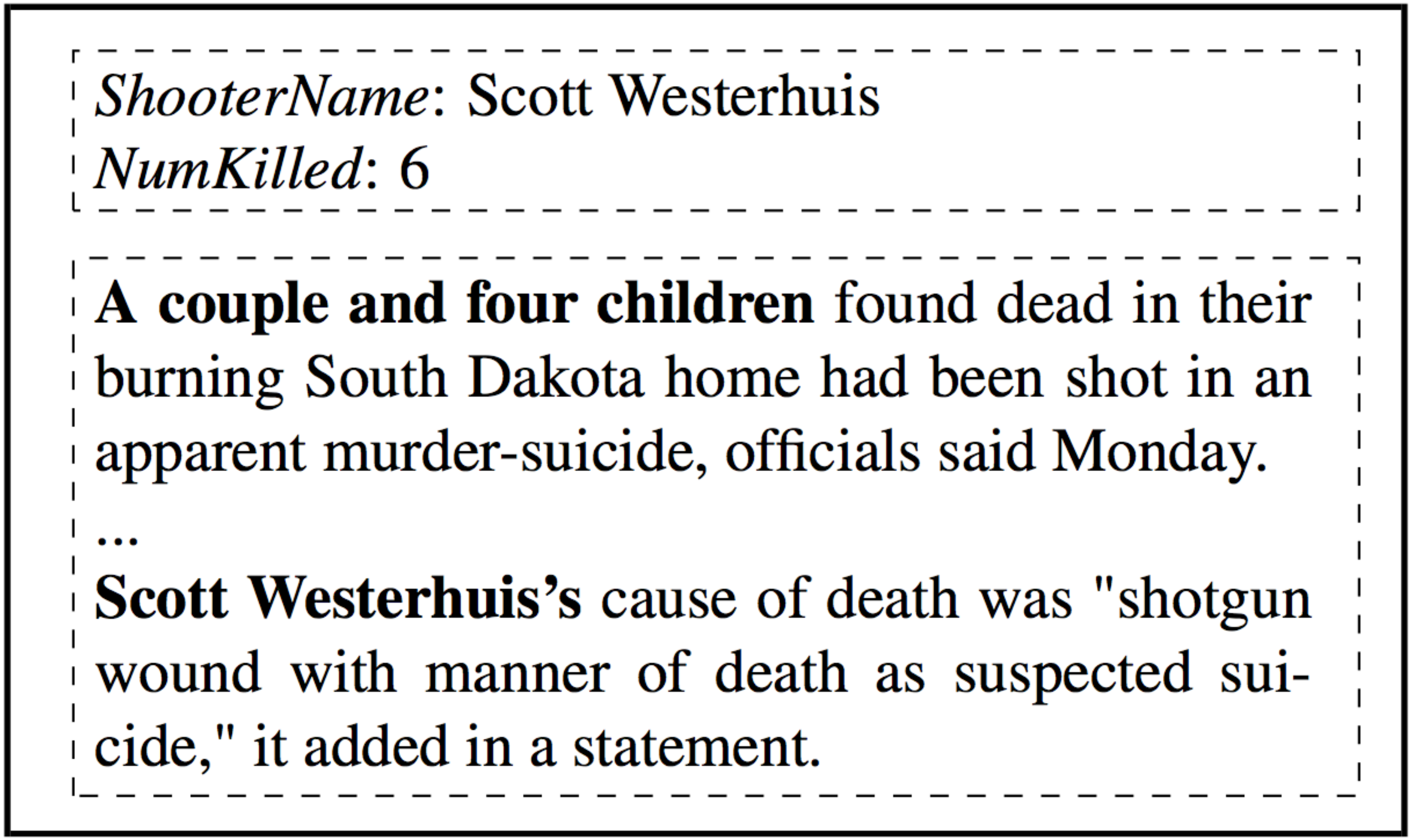
$$\nabla_\theta J(\theta) = \sum_t \sum_{a \in \mathcal{A}} \pi(a|s_t;\theta)\nabla_\theta \log \pi(a|s_t;\theta) R(s_t, a_t)$$

$$\approx \nabla_\theta \sum_t \log \pi(a = r_t|s_t;\theta) R(s_t, a_t)$$

$$R(s_t, a_t) = \lambda_1 r_{global} + \lambda_2 r_{efficiency} + \lambda_3 r_{diversity}$$

Learning Data Selection Policy with DRL

DRL for Information Extraction (Narasimhan et al., EMNLP 2016)



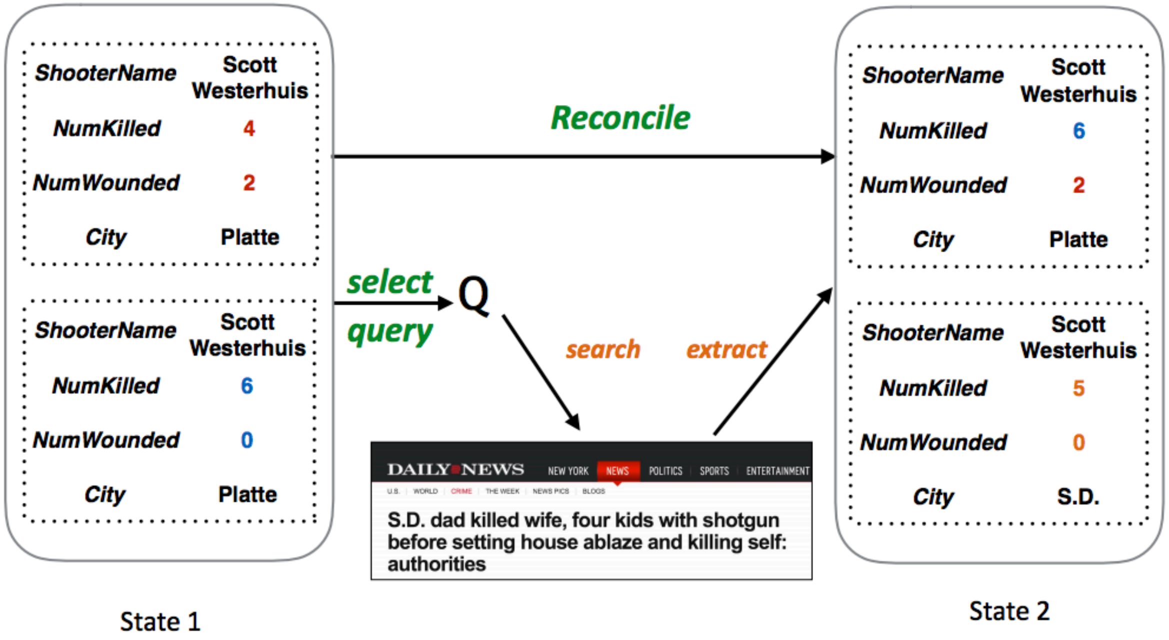
ShooterName: Scott Westerhuis
NumKilled: 6

A couple and four children found dead in their burning South Dakota home had been shot in an apparent murder-suicide, officials said Monday.

...

Scott Westerhuis's cause of death was "shotgun wound with manner of death as suspected suicide," it added in a statement.

DRL for Information Extraction (Narasimhan et al., EMNLP 2016)



S.D. dad killed wife, four kids with shotgun before setting house ablaze and killing self: authorities

Current Values:
ShooterName → Scott Westerhuis
NumKilled → 4
NumWounded → 2
City → Platte

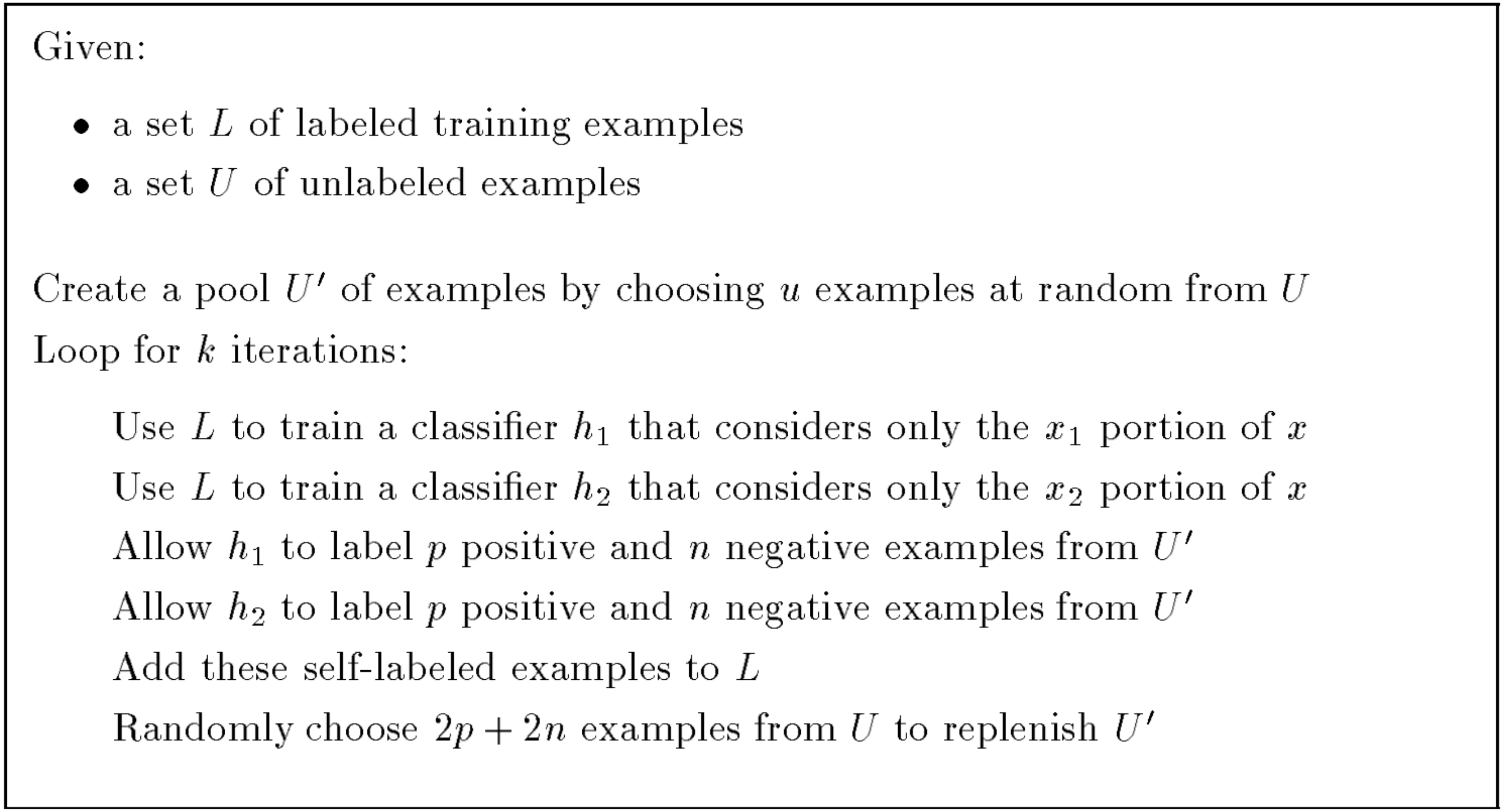
New Values:
ShooterName → Scott Westerhuis
NumKilled → 6
NumWounded → 0
City → Platte

State:
$\langle 0.3, 0.2, 0.5, 0.1,$ ← currentConf
$0.4, 0.6, 0.2, 0.4,$ ← newConf
$1, 0, 0, 1, 0, 1, 1, 0,$ ← matches
$0.2, 0.3, ..., 0.1, 0.5,$ ← contextWords
$0.65\rangle$ ← document tf-idf similarity

Can DRL help select unlabeled data for semi-supervised text classification?

A Classic Example of Semi-Supervised Learning

- Co-Training (Blum and Mitchell, 1998)

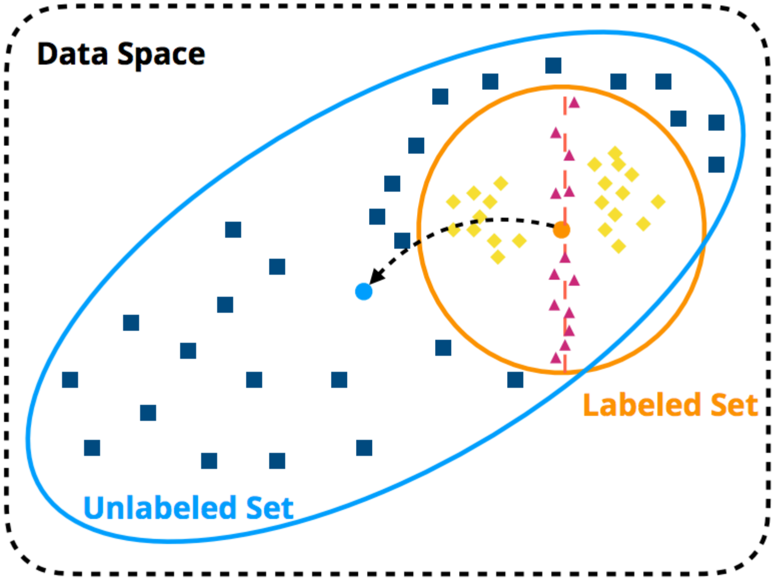


```
Given:
    • a set $L$ of labeled training examples
    • a set $U$ of unlabeled examples

Create a pool $U'$ of examples by choosing $u$ examples at random from $U$
Loop for $k$ iterations:
    Use $L$ to train a classifier $h_1$ that considers only the $x_1$ portion of $x$
    Use $L$ to train a classifier $h_2$ that considers only the $x_2$ portion of $x$
    Allow $h_1$ to label $p$ positive and $n$ negative examples from $U'$
    Allow $h_2$ to label $p$ positive and $n$ negative examples from $U'$
    Add these self-labeled examples to $L$
    Randomly choose $2p + 2n$ examples from $U$ to replenish $U'$
```

Challenges

- The two classifiers in co-training have to be independent.
- Choosing highly-confident self-labeled examples could be suboptimal.
- Sampling bias shift is common.

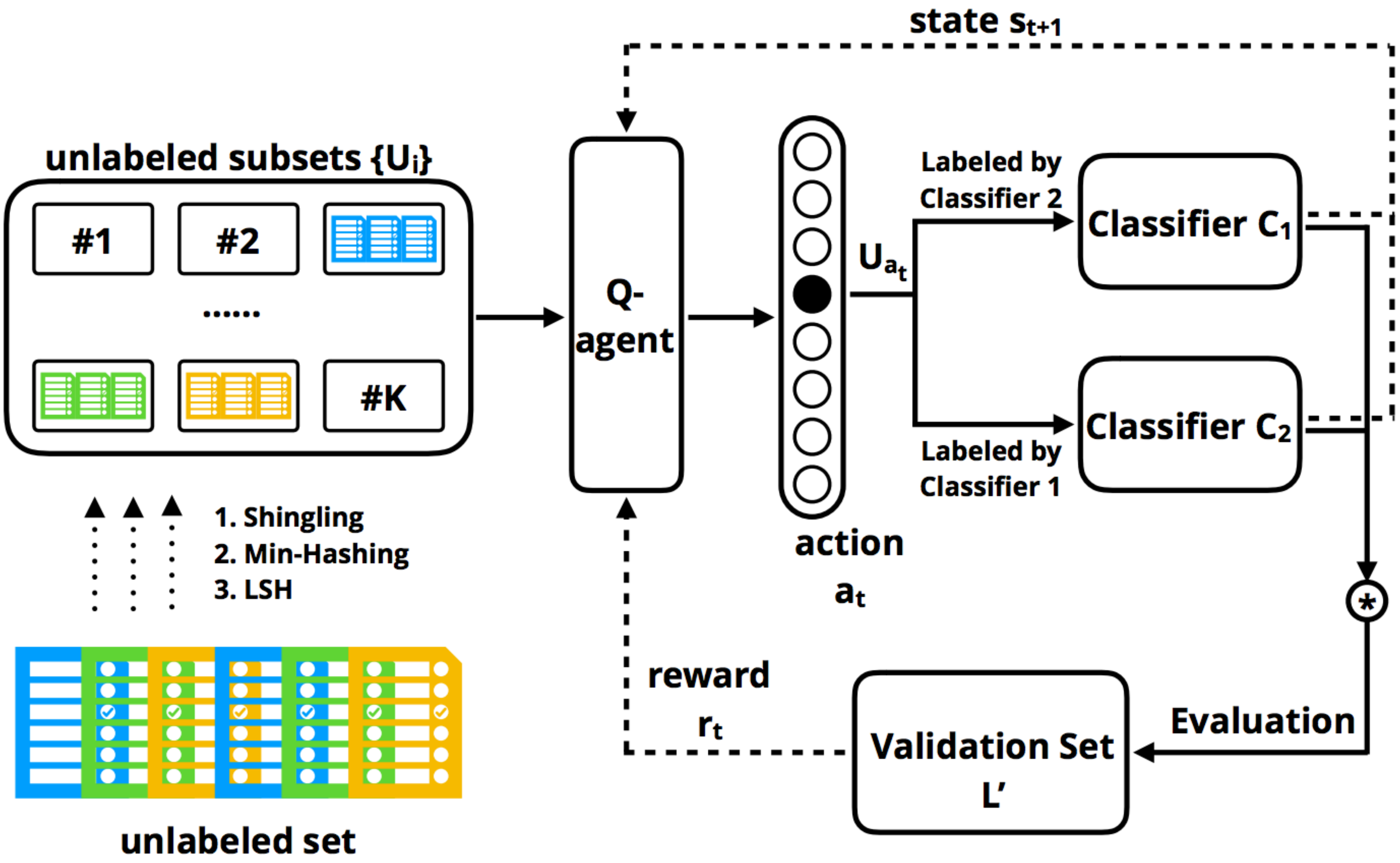


Reinforced SSL

- Assumption: not all the unlabeled data are useful.
- Idea: performance-driven semi-supervised learning that learns an unlabeled data selection policy with RL, instead of using random sampling.

- 1. Partition the unlabeled data space
- 2. Train a RL agent to select useful unlabeled data
- 3. Reward: change in accuracy on the validation set

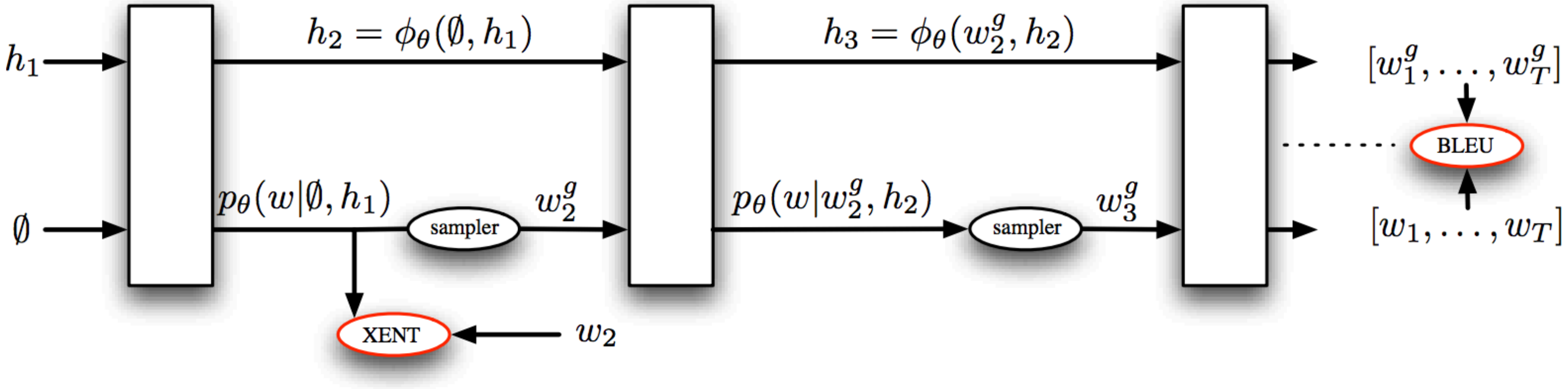
Reinforced Co-Training (Wu et al., NAACL 2018)



Directly Optimization of the Metric

MIXER (Ranzato et al., ICLR 2016)

- Optimize the cross-entropy loss and the BLEU score directly using REINFORCE (Williams, 1992).

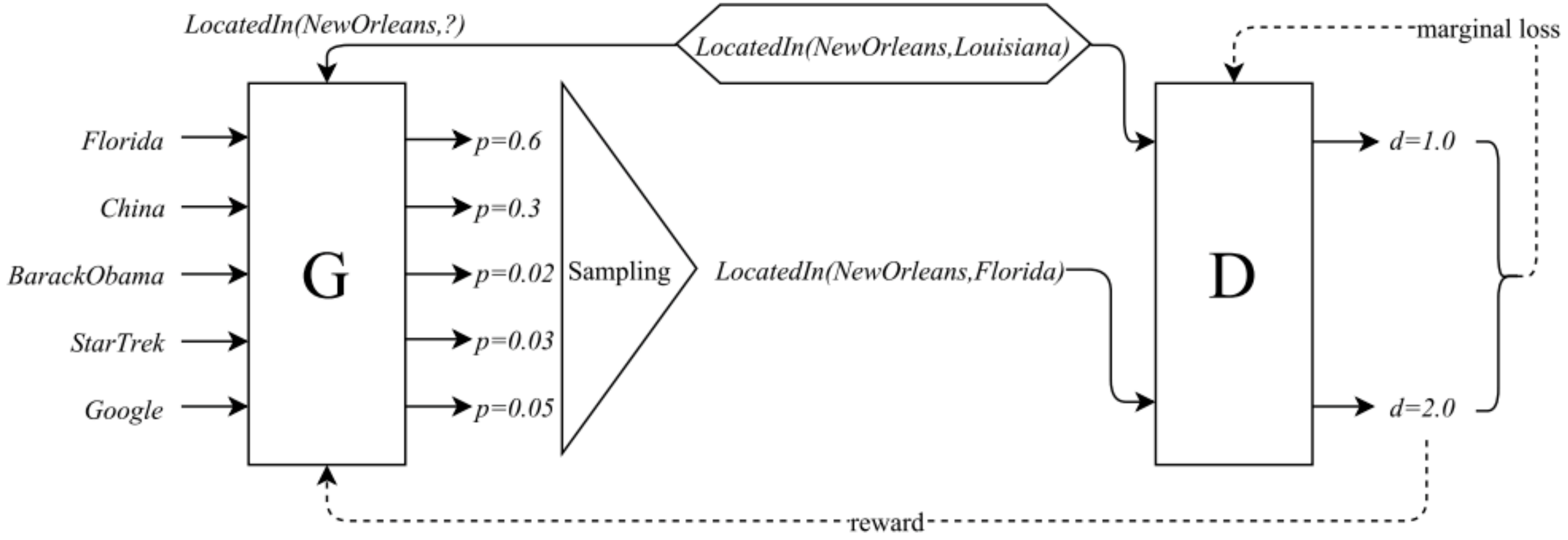


Backprop Reward via One-Step RL

One-Step Reinforcement Learning in Action

KBGAN: Learning to Generate High-Quality Negative Examples (Cai and Wang, NAACL 2018)

Idea: use adversarial learning to iteratively learn better negative examples.



KBGAN: Overview

- Both G and D are KG embedding models.

- Input:
  - Pre-trained generator G with score function $f_G(h, r, t)$.
  - Pre-trained discriminator D with score function $f_D(h, r, t)$.
- Adversarial Learning:
  - Use softmax to score and rank negative triples.
  - Update D with original positive examples and highly-ranked negative examples.
  - Pass the reward for policy gradient update for G.

- Output:
  - Adversarially trained KG embedding discriminator D.

KBGAN: Adversarial Negative Training

For each positive triple from the minibatch:

1. Generator: Rank negative examples.

$$p_G(h', r, t' | h, r, t) = \frac{\exp f_G(h', r, t')}{\sum \exp f_G(h^*, r, t^*)}$$
$$(h^*, r, t^*) \in Neg(h, r, t)$$

2. Discriminator: Standard margin-based update.

$$L_D = \sum_{(h,r,t) \in \mathcal{T}} [f_D(h, r, t) - f_D(h', r, t') + \gamma]_+$$
$$(h', r, t') \sim p_G(h', r, t' | h, r, t) \quad (3)$$

KBGAN: One-Step RL for Updating the Generator

3. Compute the Reward for the Generator.

$$\mathrm{r} = -f_D(h', r, t').$$

4. Policy gradient update for the Generator.

$$G_G \longleftarrow G_G + (r - b)\nabla_{\theta_G} \log p_s;$$

The baseline b is total reward sum / mini-batch size.

Important Directions in DRL

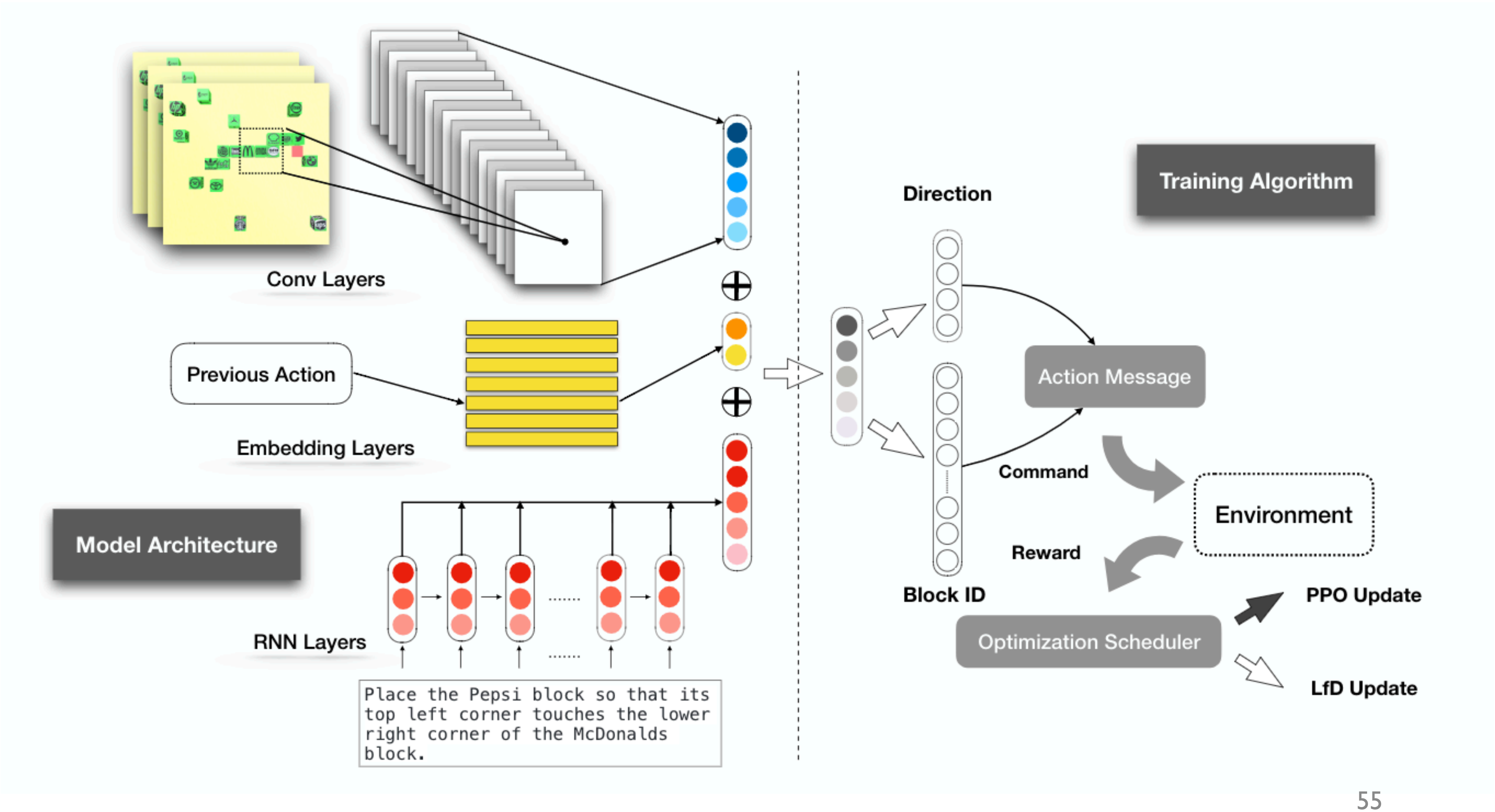
- Learning from Demonstration.
- Hierarchical DRL.
- Inverse DRL.
- Sample-efficiency.

Learning from Demonstration

Motivations

- Exploitation vs exploration.
- Cold-start DRL is very challenging.
- Pre-training (a.k.a. demonstration is common).
- Some entropy in an agent's policy is healthy.

Scheduled Policy Optimization (Xiong et al., IJCAI-ECAI 2018)



```
Place the Pepsi block so that its
top left corner touches the lower
right corner of the McDonalds
block.
```

Hierarchical Deep Reinforcement Learning

Hierarchical Deep Reinforcement Learning for Video Captioning (Wang et al., CVPR 2018)

Caption #1: A woman offers her dog some food.

Caption #2: A woman is eating and sharing food with her dog.

Caption #3: A woman is sharing a snack with a dog.

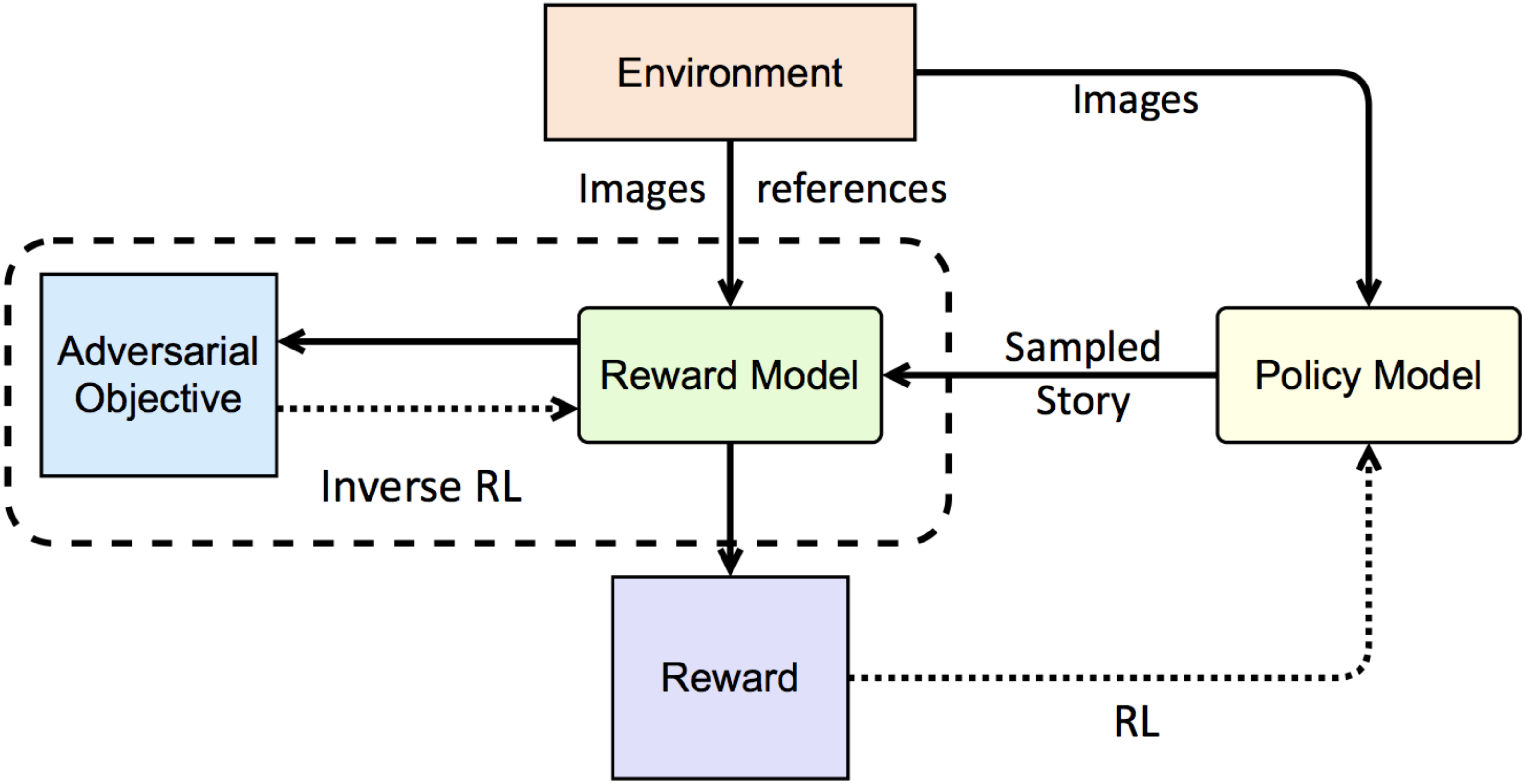
Caption: A person sits on a bed and puts a laptop into a bag. The person stands up, puts the bag on one shoulder, and walks out of the room.

Inverse Deep Reinforcement Learning

No Metrics Are Perfect: Adversarial Reward Learning (Wang et al., ACL 2018)

- Task: visual storytelling (generate a story from a sequence of images in a photo album).
- Difficulty: how to quantify a good story?
- Idea: given a policy, learn the reward function.

No Metrics Are Perfect: Adversarial Reward Learning (Wang et al., ACL 2018)

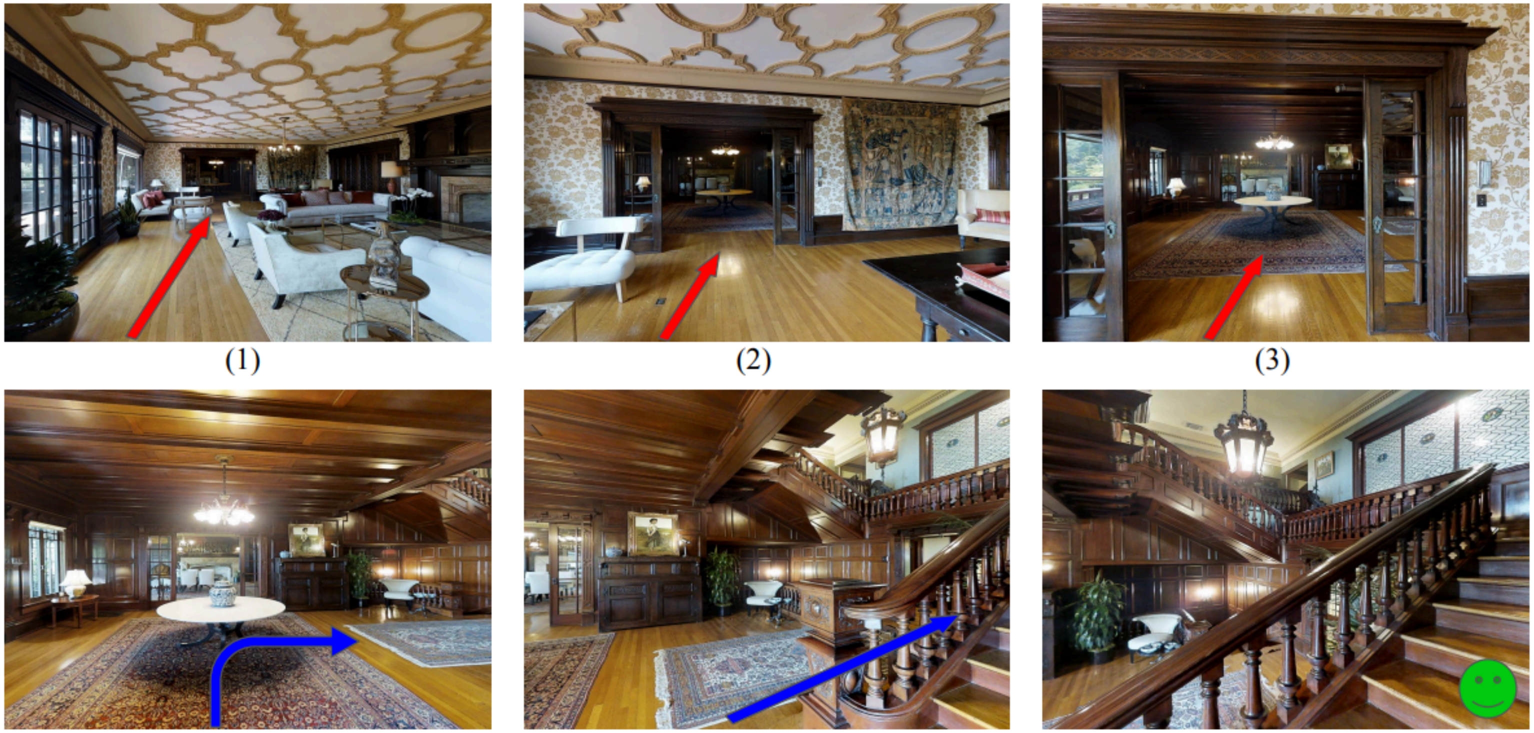


When will IRL work?

- When the optimization target is complex.
- There are no easy formulations of the reward.
- If you can clearly define the reward, don’t use IRL and it will not work.

Improving Sample Efficiency: Combine Model-Free and Model-Based RL

Vision and Language Navigation (Anderson et al., CVPR 2018)

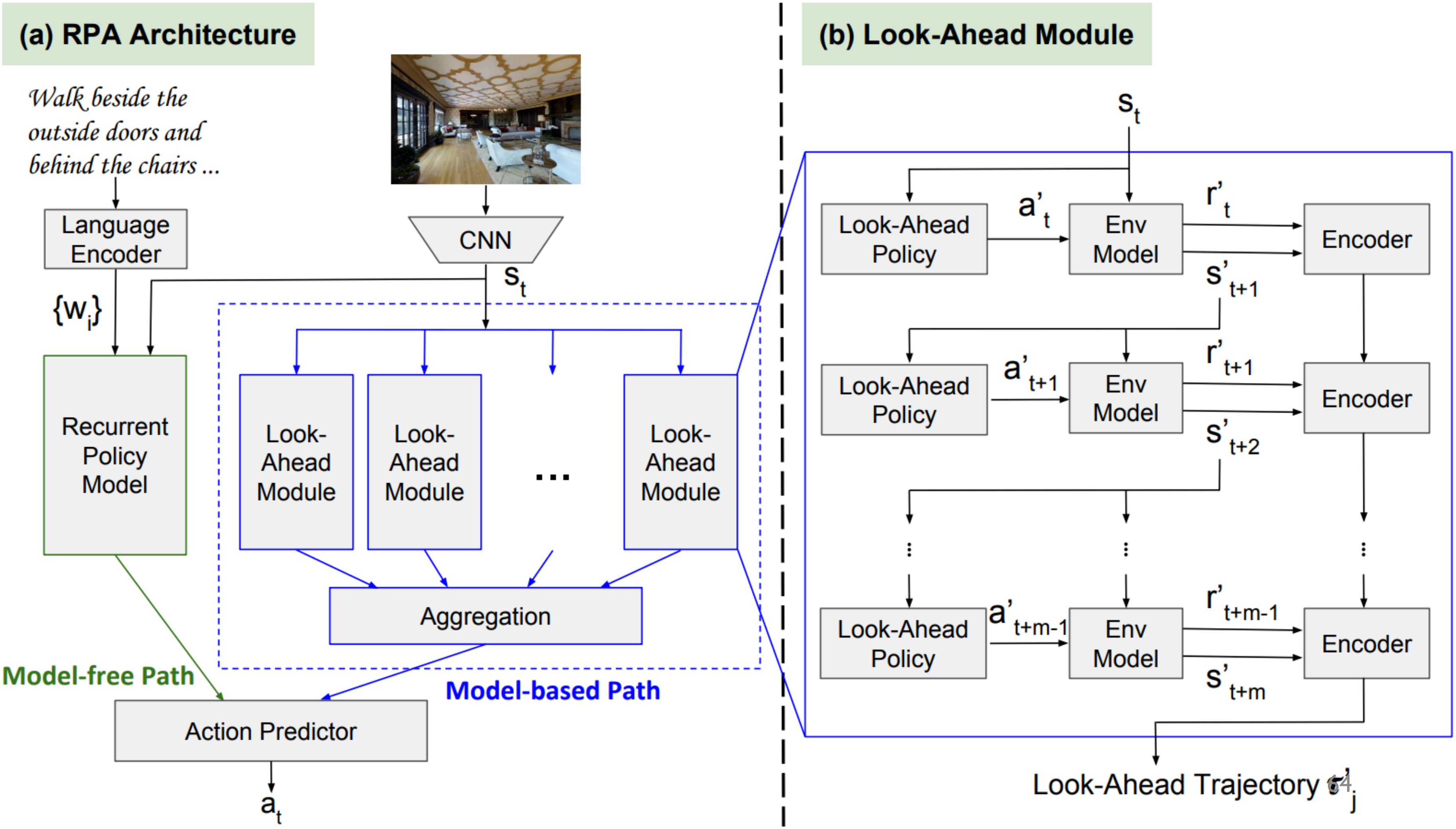


(1) (2) (3)

(4) (5) (6)

Walk beside the outside doors and behind the chairs across the room. Turn right and walk up the stairs. Stop on the seventh step.

Look Before You Leap: Combine Model-Free and Model-Based RL for Look-Ahead Search (Wang et al., ECCV 2018)



Conclusion of Part I

- We provided a gentle introduction to DRL.
- We showed the current landscape of DRL4NLP research.
- What do (NLP) people use DRL for?
- Intriguing directions in DRL4NLP.

Open-sourced software:

- DeepPath: https://github.com/xwhan/DeepPath
- KBGAN: https://github.com/cai-lw/KBGAN
- Scheduled Policy Optimization: https://github.com/xwhan/walk_the_blocks
- AREL: https://github.com/littlekobe/AREL

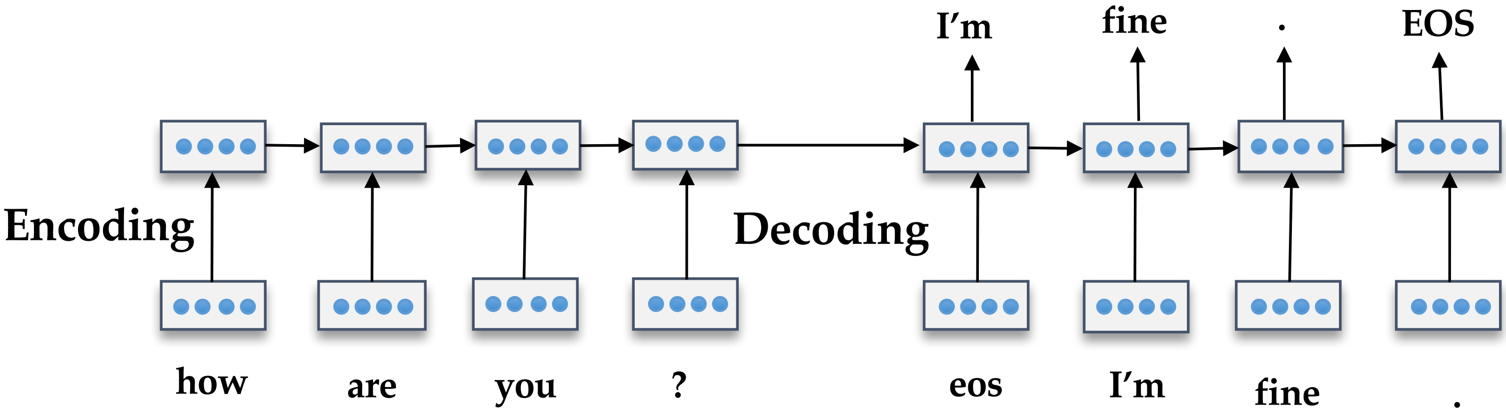
Outline

- Introduction
- Fundamentals and Overview (William Wang)
- **Deep Reinforcement Learning for Dialog (Jiwei Li)**
- Challenges (Xiaodong He)
- Conclusion

Seq2Seq Models for Response Generation

(Sutskever et al., 2014; Jean et al., 2014; Luong et al., 2015)

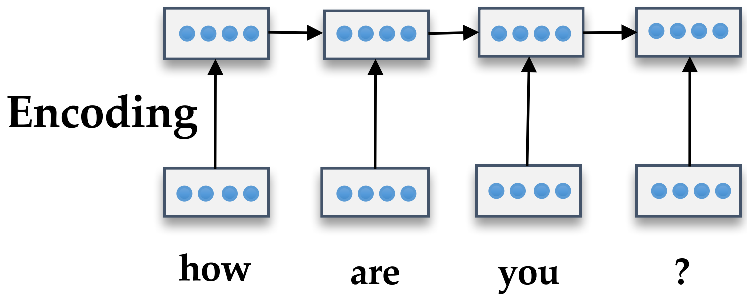
$$\text{Loss} = -\log p(\text{response}|\text{message})$$



Seq2Seq Models for Response Generation

(Sutskever et al., 2014; Jean et al., 2014; Luong et al., 2015)

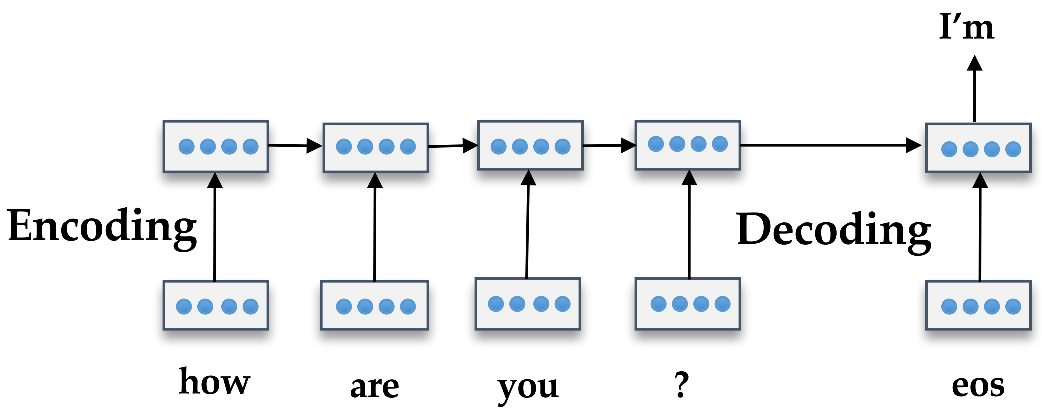
$$\text{Loss} = -\log p(\text{response}|\text{message})$$



Seq2Seq Models for Response Generation

(Sutskever et al., 2014; Jean et al., 2014; Luong et al., 2015)

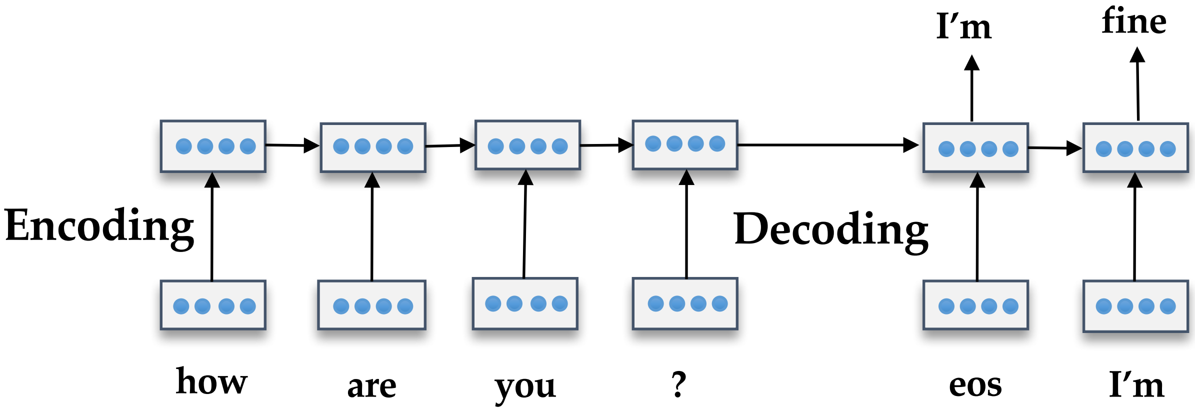
$$\text{Loss} = -\log p(\text{response}|\text{message})$$



Seq2Seq Models for Response Generation

(Sutskever et al., 2014; Jean et al., 2014; Luong et al., 2015)

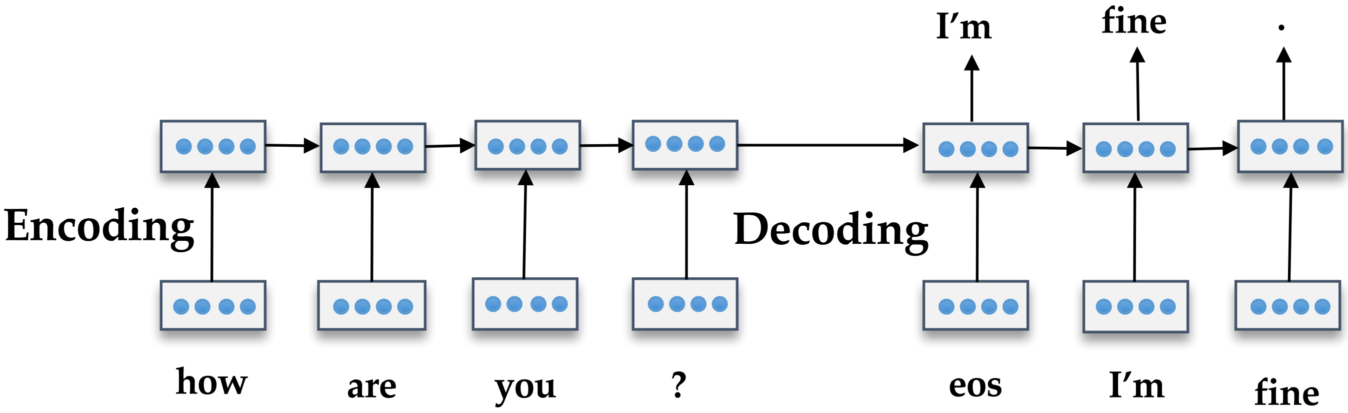
$$\text{Loss} = -\log p(\text{response}|\text{message})$$



Seq2Seq Models for Response Generation

(Sutskever et al., 2014; Jean et al., 2014; Luong et al., 2015)

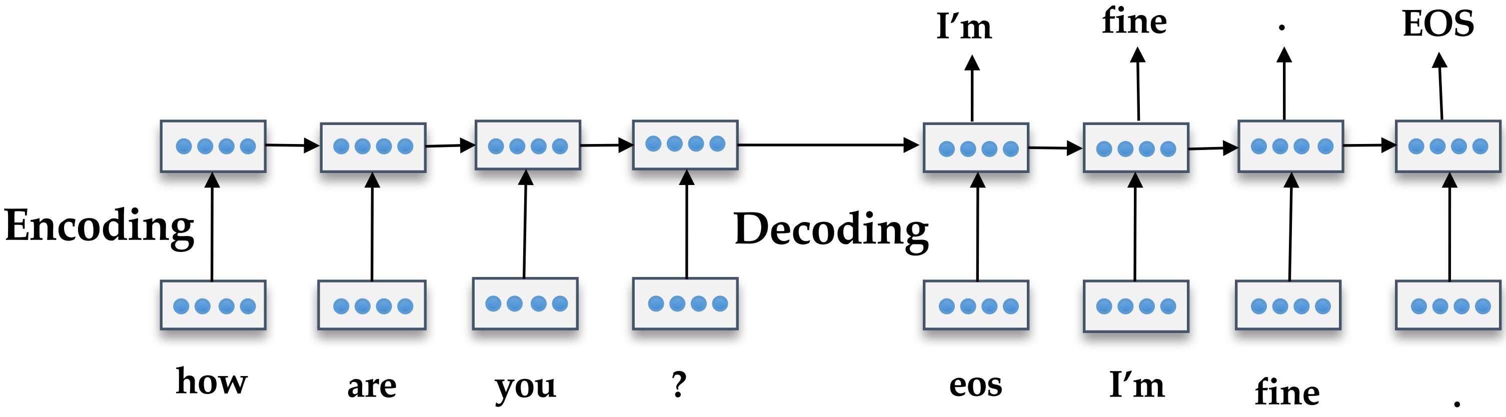
$$\text{Loss} = -\log p(\text{response}|\text{message})$$



Seq2Seq Models for Response Generation

(Sutskever et al., 2014; Jean et al., 2014; Luong et al., 2015)

$$\text{Loss} = -\log p(\text{response}|\text{message})$$

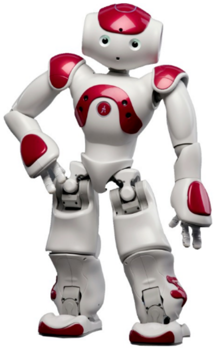


Issues

How do we handle long-term dialogue success?

- Problem 1: Repetitive responses.

Two bots talk with each other



Repetitive responses.

Shut up !

Repetitive responses.

Shut up !

Repetitive responses.

Shut up !

No, you shut up !

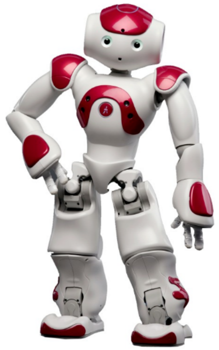
Repetitive responses.

Shut up !

No, you shut up !

No, you shut up !

Repetitive responses.



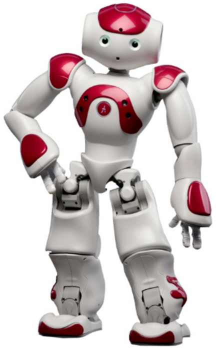
Shut up !

No, you shut up !

No, you shut up !

No, you shut up !

Repetitive responses.



Shut up !

No, you shut up !

No, you shut up !

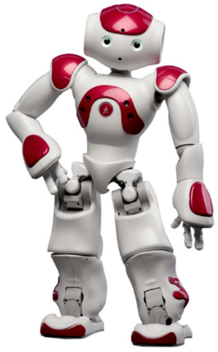
No, you shut up !

No, you shut up !

● ● ●

● ● ●

Repetitive responses.



See you later !

See you later !

See you later !

See you later !

See you later !

……

Issues

How do we handle long-term dialogue success?

- Problem 1: Repetitive responses.
- Problem 2: Short-sighted conversation decisions.

Short-sighted conversation decisions

How old are you ?

Short-sighted conversation decisions

How old are you ?

i 'm 16 .

Short-sighted conversation decisions

How old are you ?

i 'm 16 .

16 ?

Short-sighted conversation decisions

How old are you ?

i 'm 16 .

16 ?

i don 't know what you 're talking about

Short-sighted conversation decisions

How old are you ?

i 'm 16 .

16 ?

i don 't know what you 're talking about

you don 't know what you 're saying

Short-sighted conversation decisions

How old are you ?

i 'm 16 .

16 ?

i don 't know what you 're talking about

you don 't know what you 're saying

i don 't know what you 're talking about

Short-sighted conversation decisions

How old are you ?

i 'm 16 .

16 ?

i don 't know what you 're talking about

you don 't know what you 're saying

i don 't know what you 're talking about

you don 't know what you 're saying

Short-sighted conversation decisions

A bad action

How old are you ?

i 'm 16 .

16 ?

i don 't know what you 're talking about

you don 't know what you 're saying

i don 't know what you 're talking about

you don 't know what you 're saying

Short-sighted conversation decisions

How old are you ?

i 'm 16 .

16 ?

i don 't know what you 're talking about

you don 't know what you 're saying

i don 't know what you 're talking about

you don 't know what you 're saying

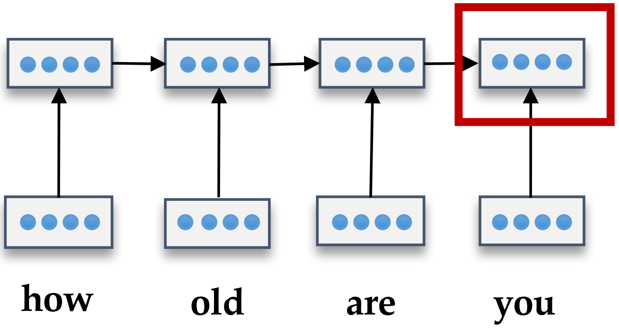
Outcome does not emerge until a few turns later

Reinforcement Learning

Notations: State

r_{i-1}

How old are you ?



Encoding

Notations: Action

r_{i-1}

How old are you ?

r_i

i 'm 16 .

Reward

Reward

1. Ease of answering

$$r(\text{response}) = -\log(\text{dull utterances}|\text{response})$$

Reward

1. Ease of answering

$$r(\text{response}) = -\log(\text{dull utterances}|\text{response})$$

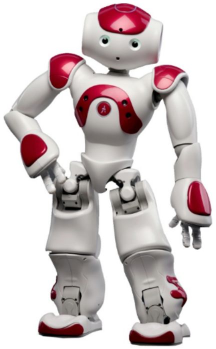
”I don’t know what you are talking about”

Reward

2. Information Flow

Reward

2. Information Flow



See you later !

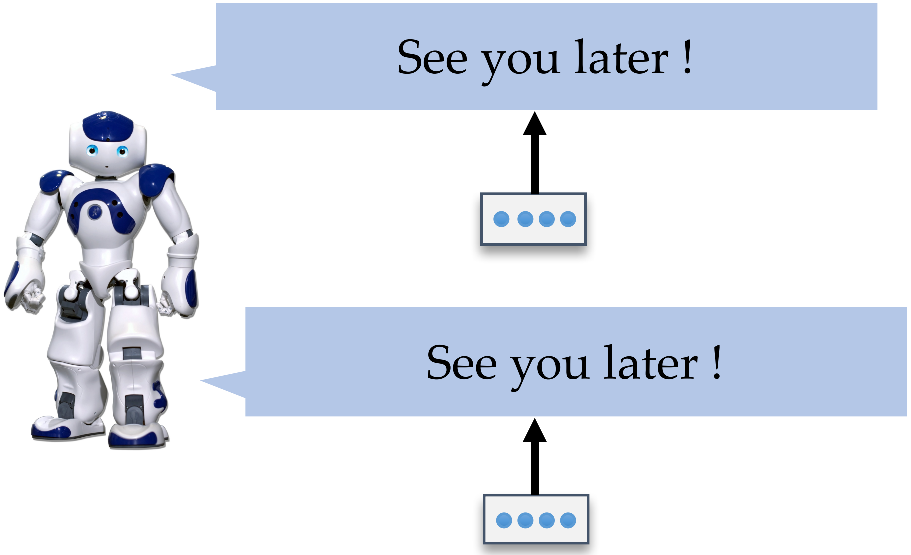
See you later !

See you later !

See you later !

Reward

2. Information Flow



See you later !

See you later !

$$r_2 = -\log \mathrm{Sigmoid}(\cos(s_1, s_2))$$

Reward

3. Meaningfulness

message

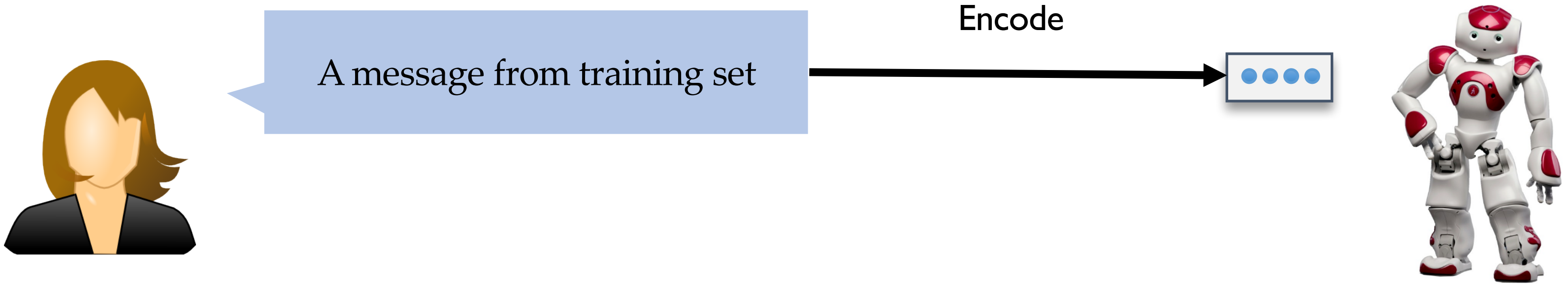
How old are you ?

response

i 'm 16 .

$$r = \log p(\text{response}|\text{message}) + \log p(\text{message}|\text{response})$$

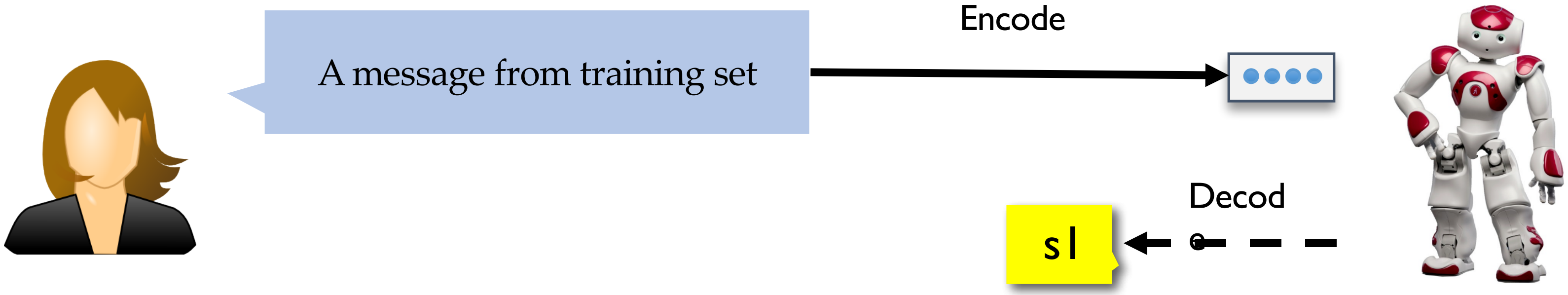
Simulation



A message from training set

Encode

Simulation



A message from training set

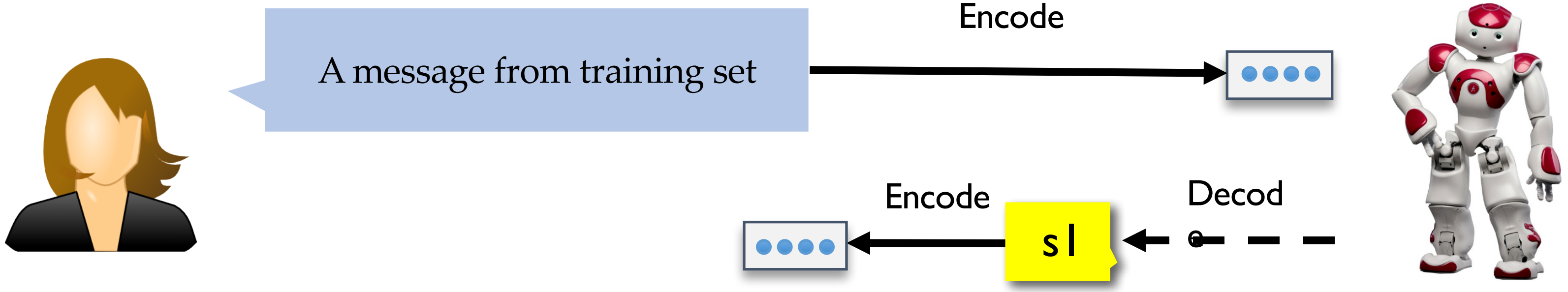
Encode

Decod

s l

. . .

Simulation



A message from training set

Encode

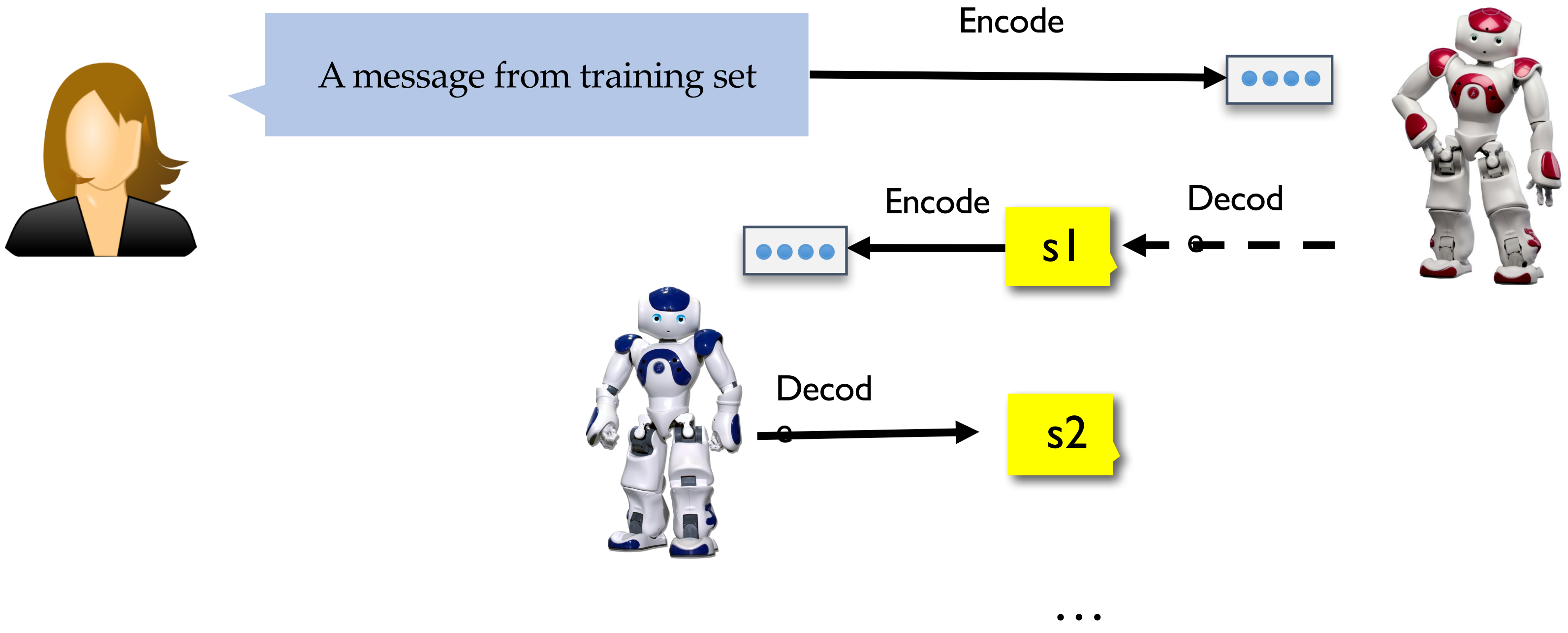
Encode

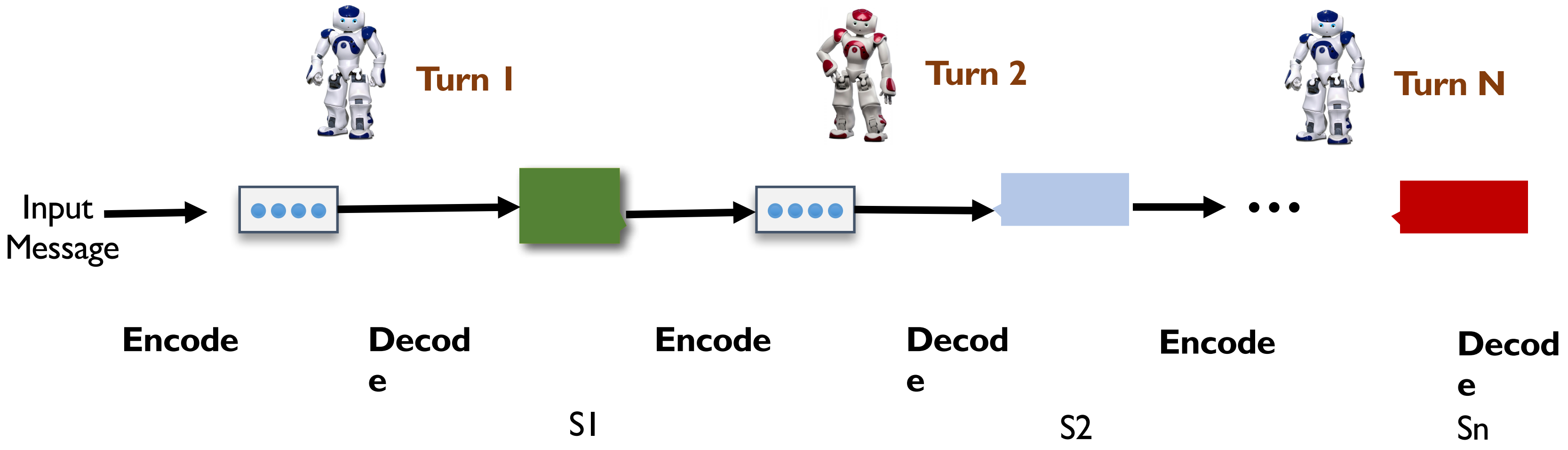
s l

Decod

…

Simulation





Turn 1

Turn 2

Turn N

Input Message

Encode

Decode

S1

Encode

Decode

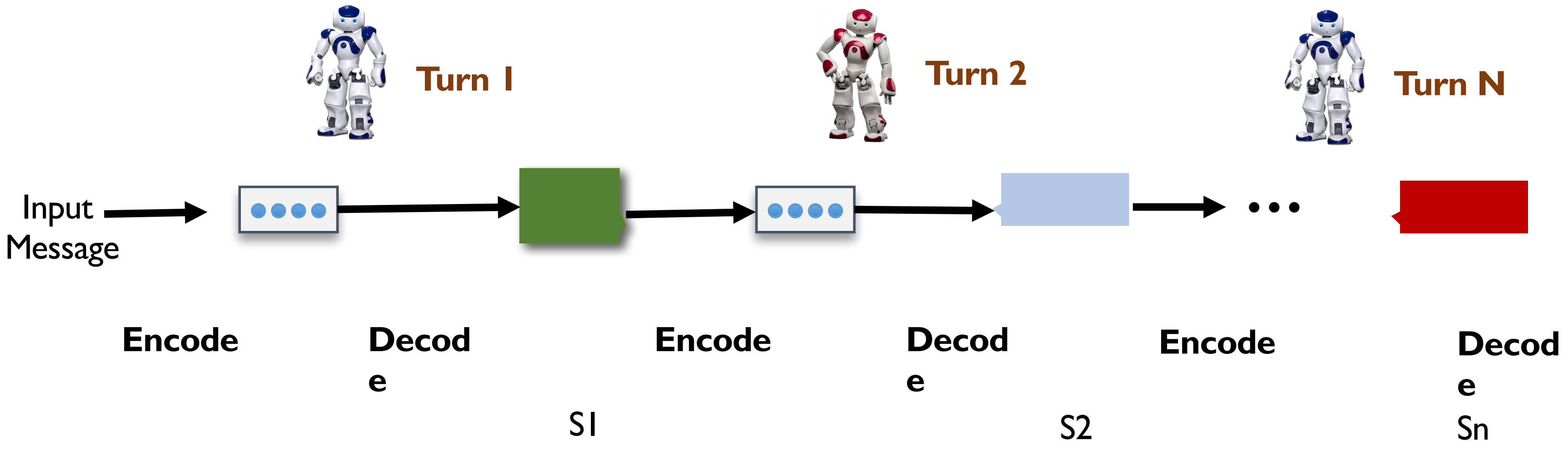
S2

Encode

Decode

Sn

Compute Accumulated Reward R(S1,S2,…,Sn)



Turn 1

Turn 2

Turn N

Input Message

•••

Encode

Decod e

S1

Encode

Decod e

S2

Encode

Decod e

Sn

REINFORCE Algorithm (William,1992)

$$J(\theta) = \mathbb{E}[R(s_1, s_2, ..., s_N)]$$

Evaluation

	RL-win	RL-lose	Tie
Single-Turn Quality	0.40	0.36	0.24
Multi-Turn Quality	0.72	0.12	0.16

Evaluation

	RL-win	RL-lose	Tie
Single-Turn Quality	0.40	0.36	0.24
Multi-Turn Quality	0.72	0.12	0.16

Results

Input	Mutual Information	The RL model
How old are you ?	I’m 16.	I’m 16. why are you asking ?

Results

Input	Mutual Information	The RL model
How old are you ?	I’m 16.	I’m 16. why are you asking ?
what is your full name ?	i have no idea	what 's yours ?

Results

Input	Mutual Information	The RL model
How old are you ?	I’m 16.	I’m 16. why are you asking ?
what is your full name ?	i have no idea	what 's yours ?
I don 't want to go home tonight .	Really ?	Why ?

Results

Input	Mutual Information	The RL model
How old are you ?	I'm 16.	I'm 16. why are you asking ?
what is 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.	Would I see you if I didn 't ?

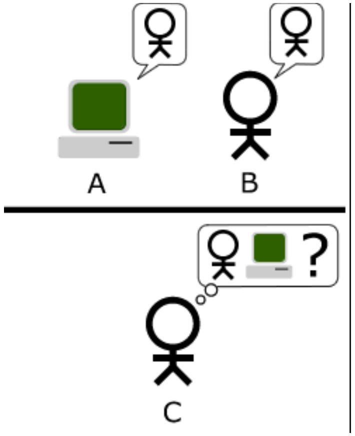
Reward for Good Dialogue

1. Easy to answer
2. Information Flow
3. Meaningfulness

What Should Rewards for Good Dialogue Be Like ?

Reward for Good Dialogue

Turing Test



Reward for Good Dialogue

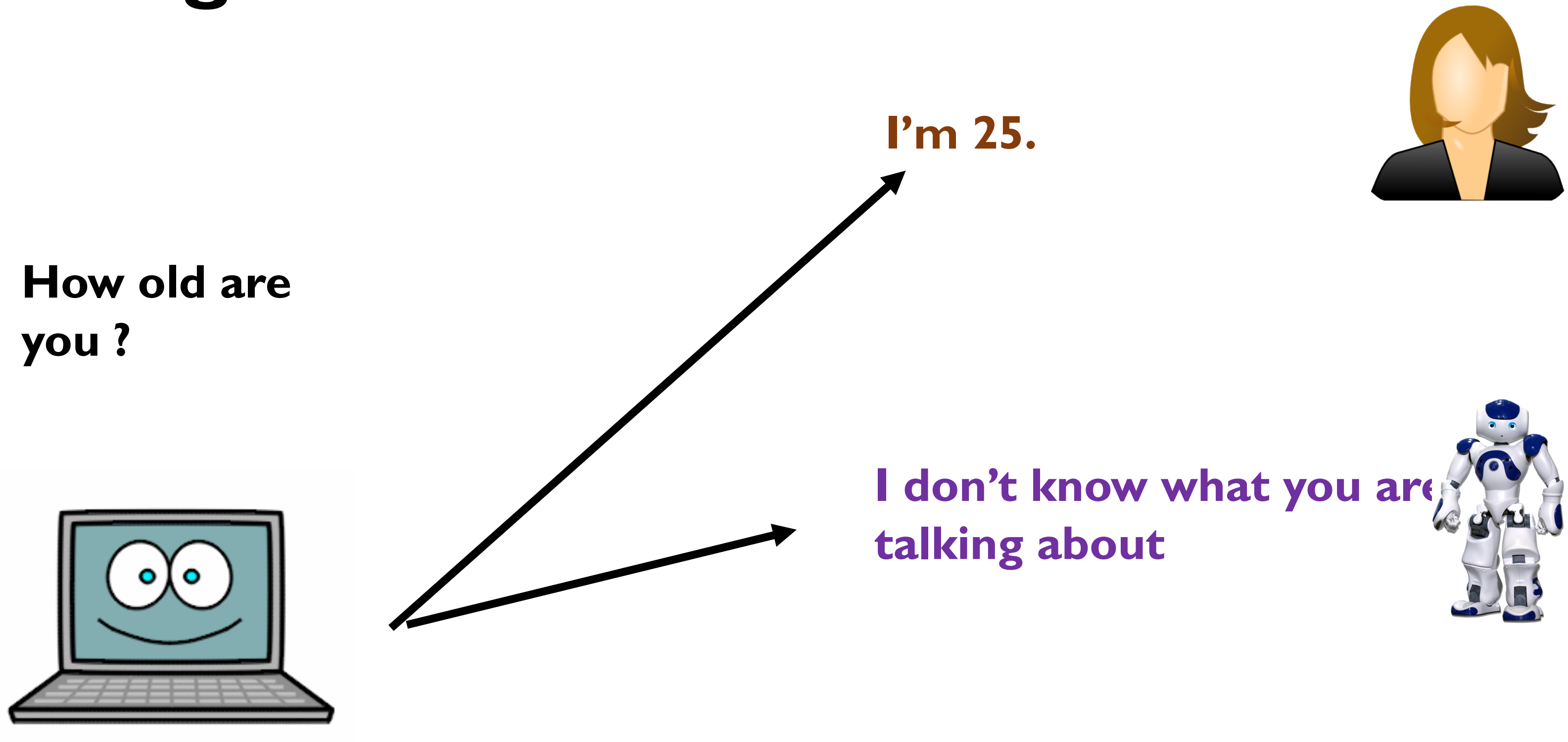
I'm 25.

How old are you ?

I don't know what you are talking about

A human evaluator/ judge

Reward for Good Dialogue

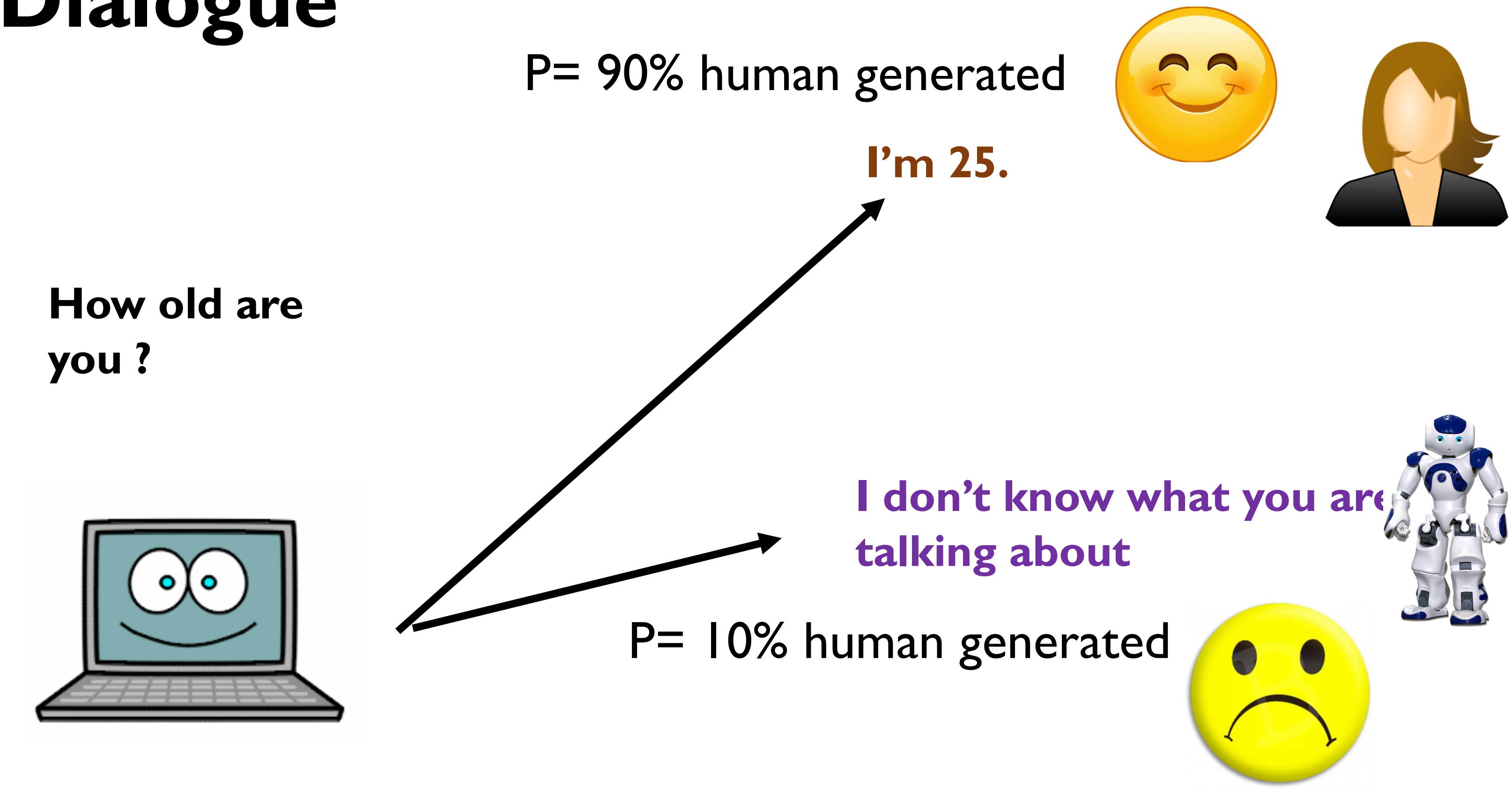


How old are you ?

I'm 25.

I don't know what you are talking about

Reward for Good Dialogue



How old are you ?

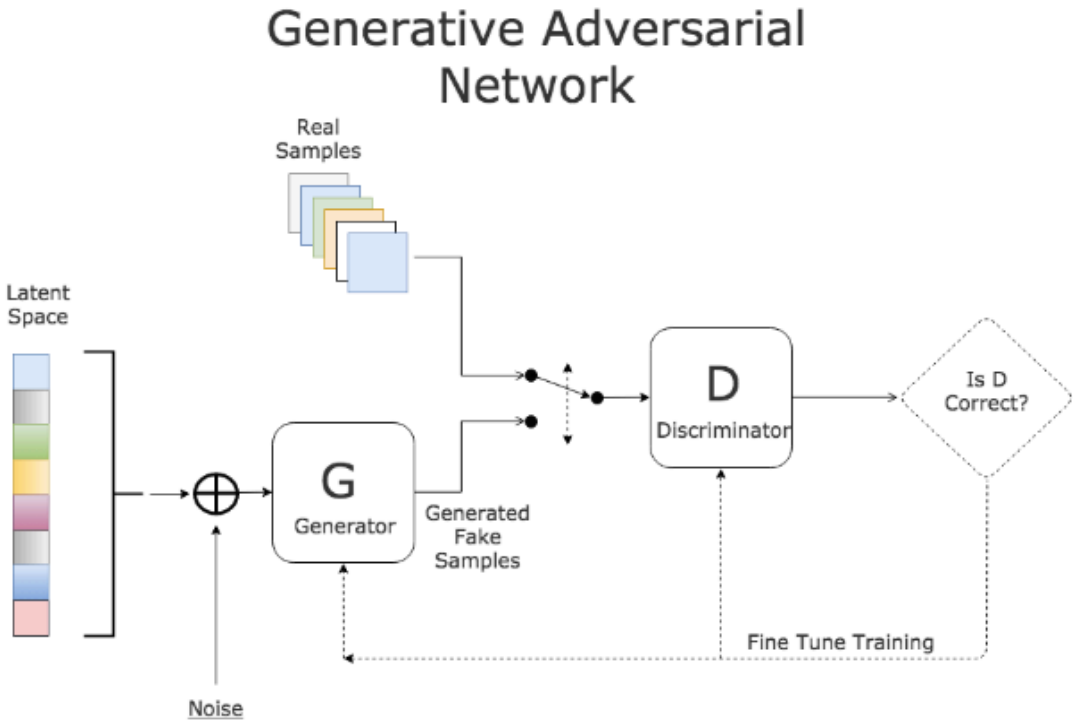
P= 90% human generated

I'm 25.

I don't know what you are talking about

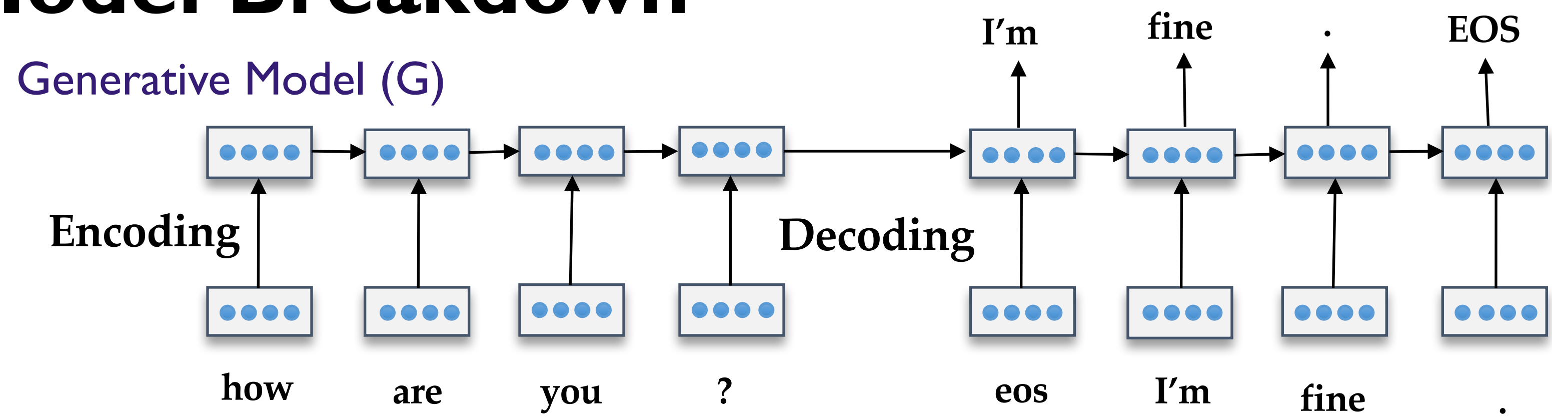
P= 10% human generated

Adversarial Learning in Image Generation (Goodfellow et al., 2014)



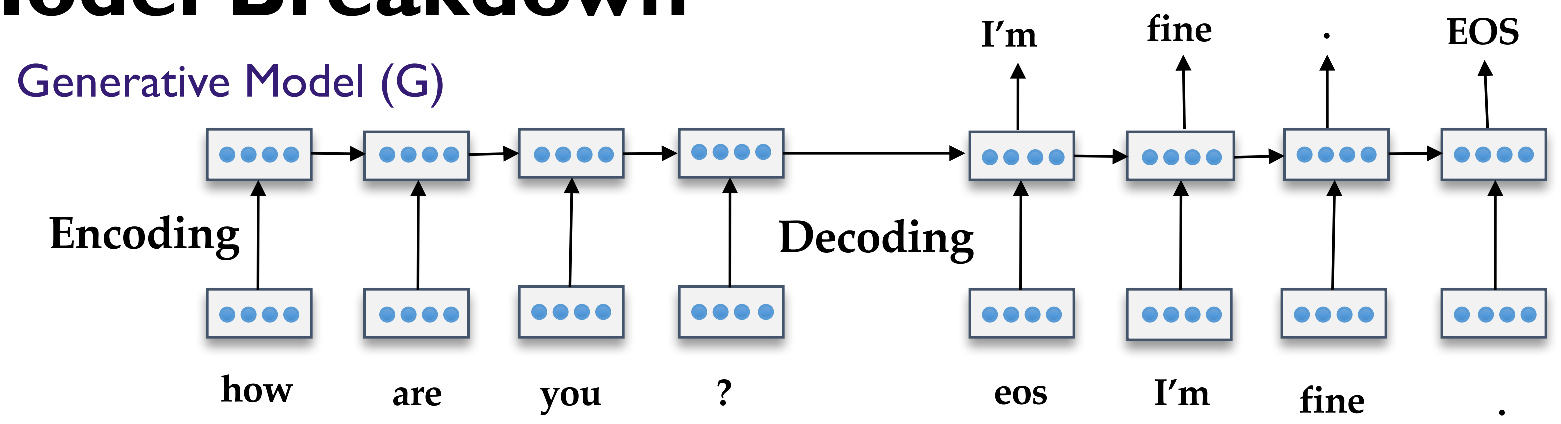
Model Breakdown

Generative Model (G)

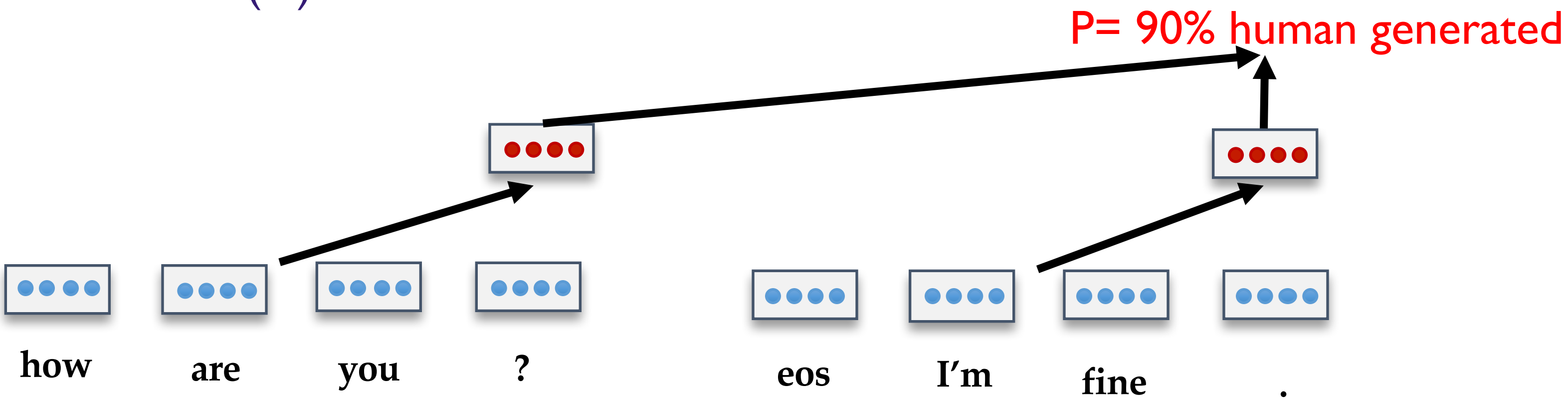


Model Breakdown

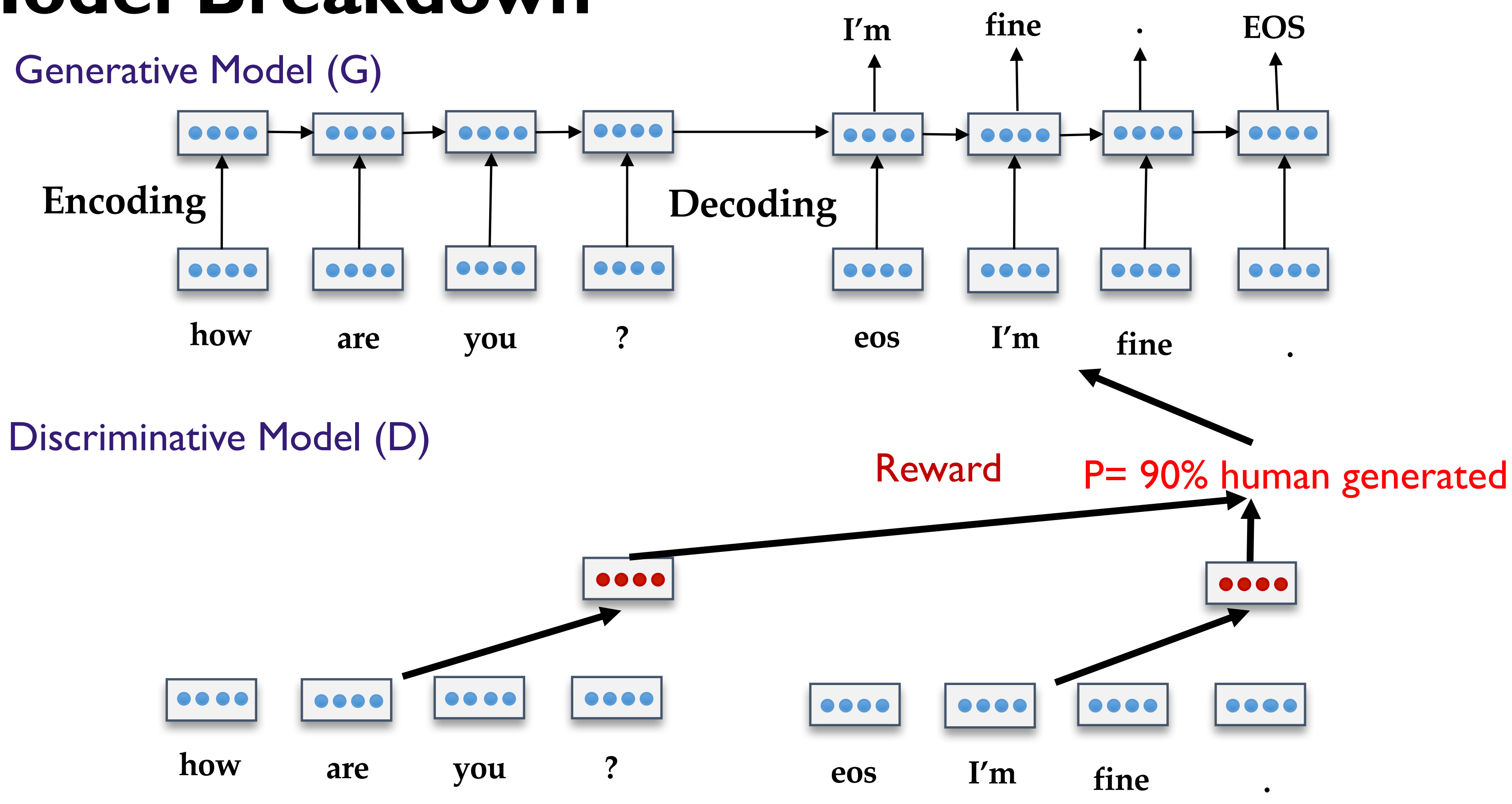
Generative Model (G)



Discriminative Model (D)

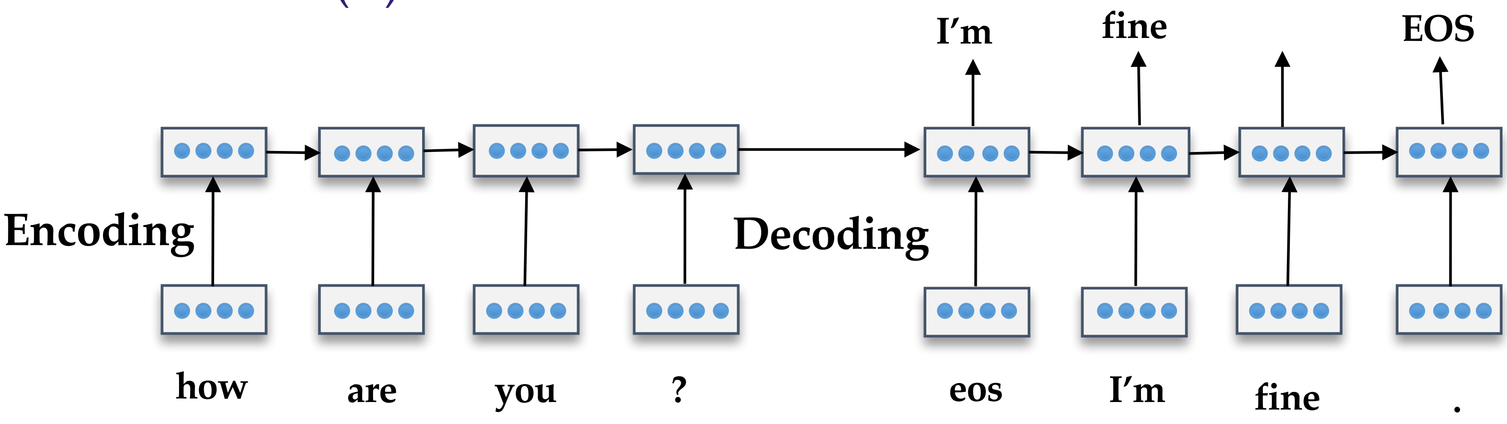


Model Breakdown



Policy Gradient

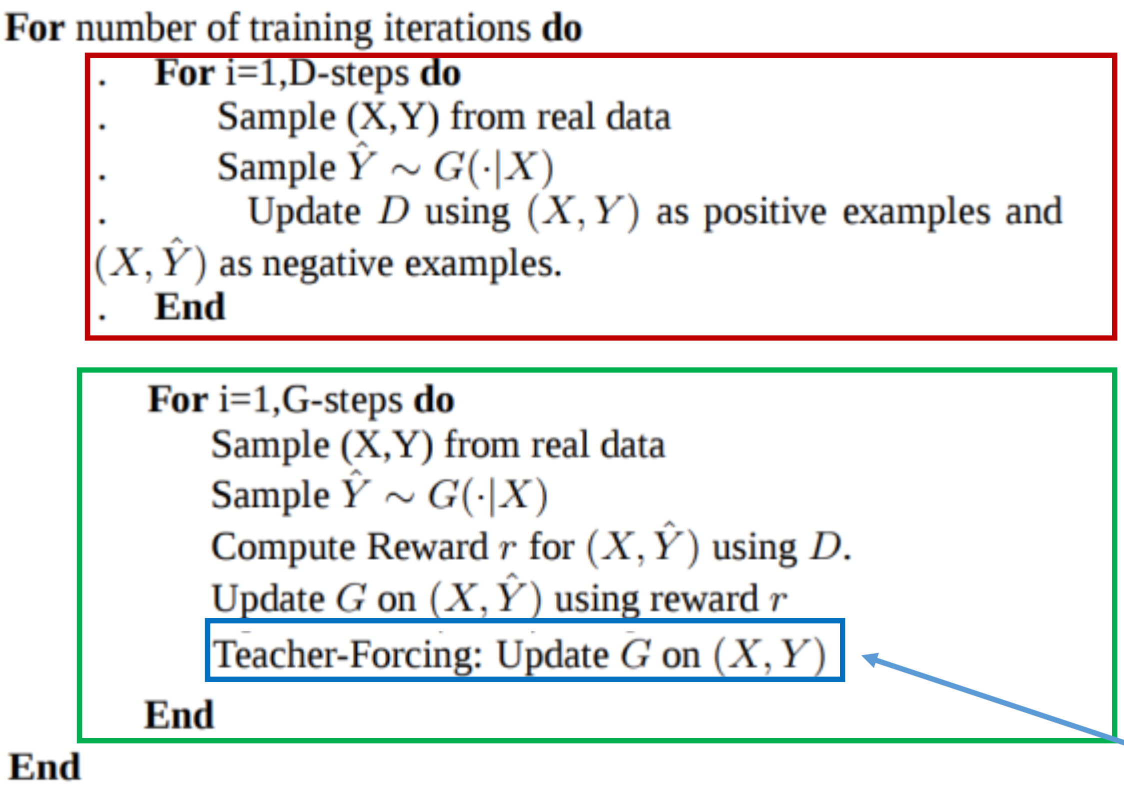
Generative Model (G)



REINFORCE Algorithm (William,1992)

$$J = E[R(y)]$$

Adversarial Learning for Neural Dialogue Generation



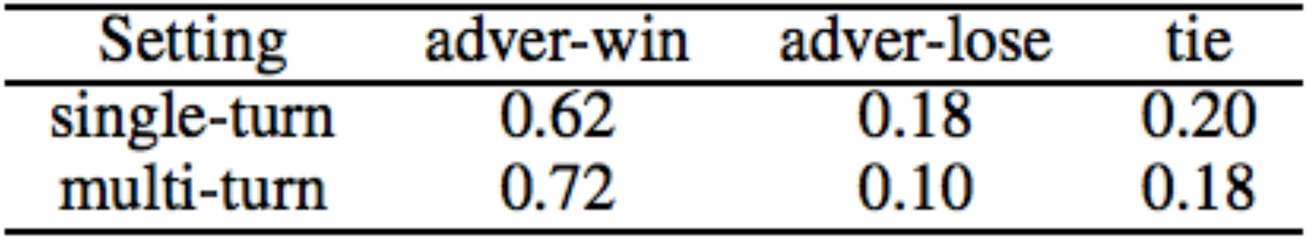
```
For number of training iterations do
    .   For i=1,D-steps do
    .       Sample (X,Y) from real data
    .       Sample $\hat{Y} \sim G(\cdot|X)$
    .         Update $D$ using $(X,Y)$ as positive examples and
    $(X,\hat{Y})$ as negative examples.
    .   End
        For i=1,G-steps do
            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
```

Update the Discriminator

Update the Generator

The discriminator forces the generator to produce correct responses

Human Evaluation



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

The previous RL model only perform better on multi-turn conversations

Results: Adversarial Learning Improves Response Generation

Human Evaluator

vs a vanilla generation model

Adversarial Win	Adversarial Lose	Tie
62%	18%	20%

Sample response

Tell me ... how long have you had this falling sickness ?

System	Response

Sample response

Tell me ... how long have you had this falling sickness ?

System	Response
Vanilla-Seq2Seq	I don’t know what you are talking about.

Sample response

Tell me ... how long have you had this falling sickness ?

System	Response
Vanilla-Seq2Seq	I don’t know what you are talking about.
Mutual Information	I’m not a doctor.

Sample response

Tell me ... how long have you had this falling sickness ?

System	Response
Vanilla-Seq2Seq	I don’t know what you are talking about.
Mutual Information	I’m not a doctor.
Adversarial Learning	A few months, I guess.

Outline

- Introduction
- Fundamentals and Overview (William Wang)
- Deep Reinforcement Learning for Dialog (Jiwei Li)
- **Frontiers and Challenges (Xiaodong He)**
- Conclusion

Frontiers and Challenges

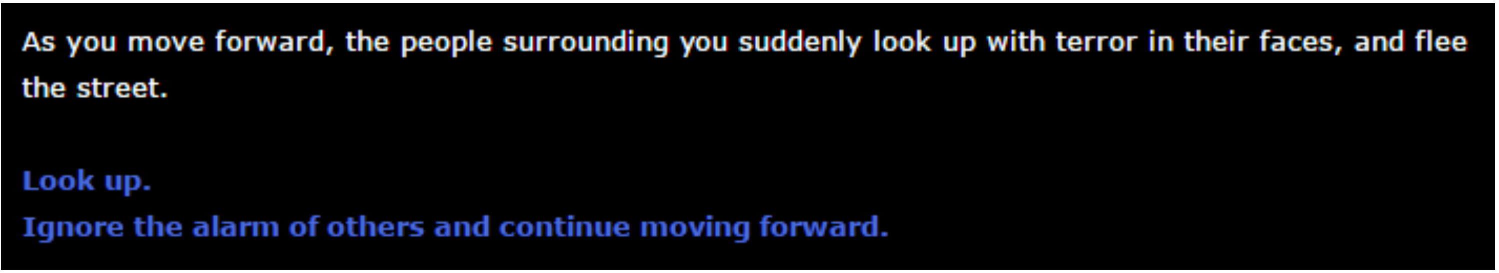
- NLP problems that presents new challenges to RL
  - An unbounded action space defined by natural language
  - Dealing with combinatorial actions and external knowledges
  - Learning reward functions for NLG
- RL problems that are particularly relevant to NLP
  - Sample complexity
  - Model-based vs. model free RL
  - Acquiring rewards

Consider a Sequential Decision Making Problem in NLP

- E.g., Playing text-based games, Webpage navigation, task completion, …
- At time t:
  - Agent observes **the state as a string of text** , e.g., state-text s_t
  - Agent also knows **a set of possible actions, each is described as a string text**, e.g., action-texts
  - Agent tries to understand the “state text” and all possible “action texts”, and takes the **right** action – to maximize the long term reward
  - Then, the environment state transits to a new state, agent receives an immediate reward, and move to t+1

RL for Natural Language Understanding Tasks

- Reinforcement learning (RL) with a natural language state and action space
  - Applications such as text games, webpage navigation, dialog systems
  - Challenging because the potential state and action space are large and sparse
- An example: text-based game



State text

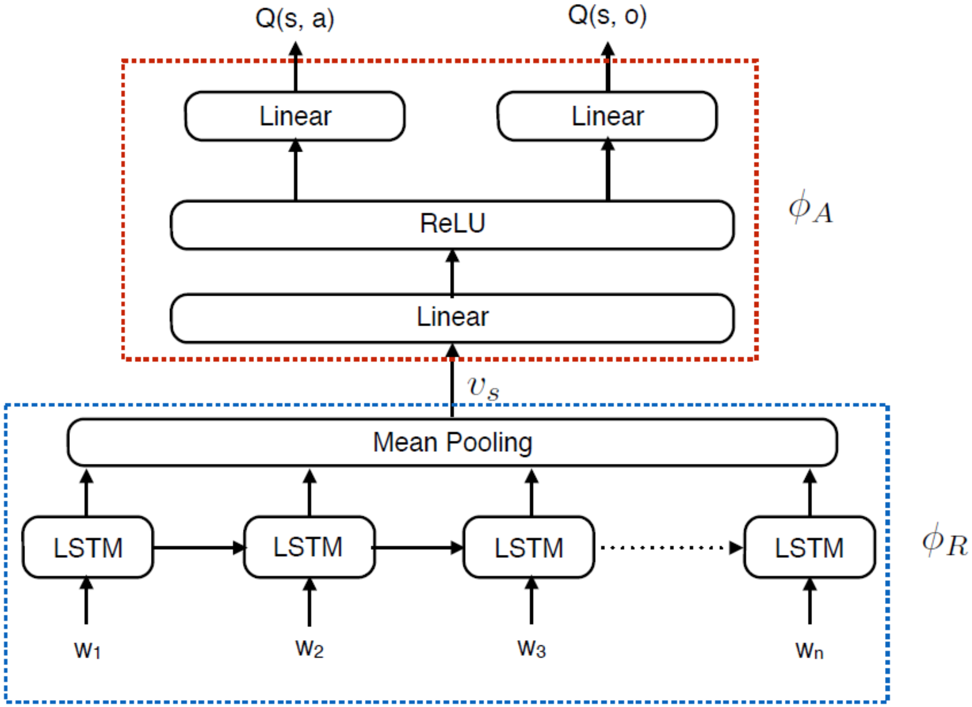
As you move forward, the people surrounding you suddenly look up with terror in their faces, and flee the street.

Action texts

Look up.
Ignore the alarm of others and continue moving forward.

DQN for RL in NLP

- LSTM-DQN
  - State is represented by a continuous vector (by a LSTM)
  - Actions and objects are considered as independent symbols
  - Tested on a MUD style text-based game playing benchmark

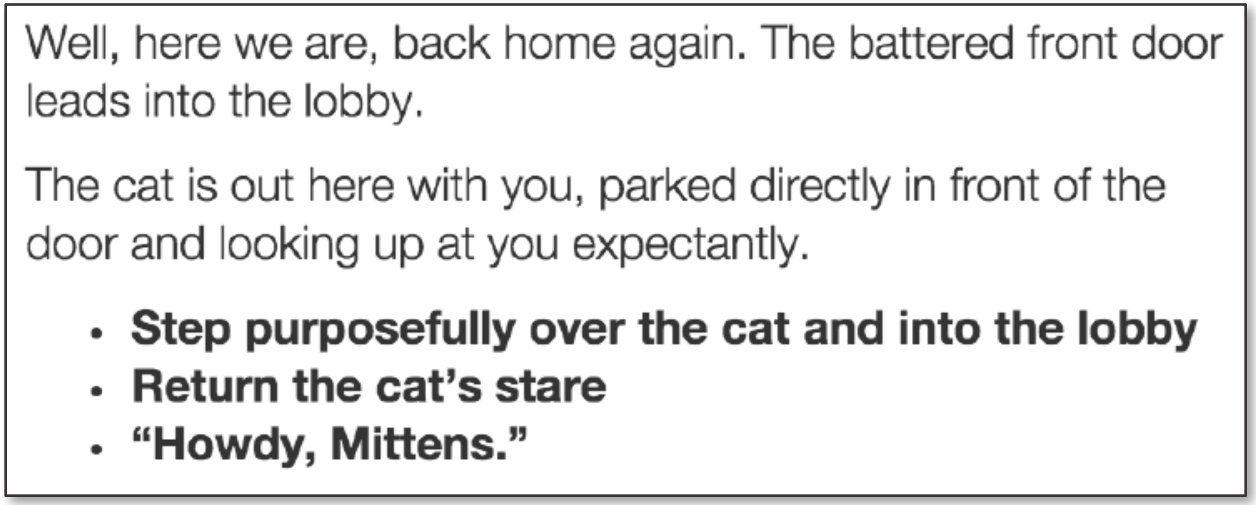


Narasimhan, K., Kulkarni, T. and Barzilay, R., 2015. Language understanding for text-based games using deep reinforcement learning. *EMNLP*.

Unbounded action space in RL for NLP

But, not only the state space is huge, the action space is huge, too.

– Action is characterized by unbounded natural language description.



Well, here we are, back home again. The battered front door leads into the lobby.

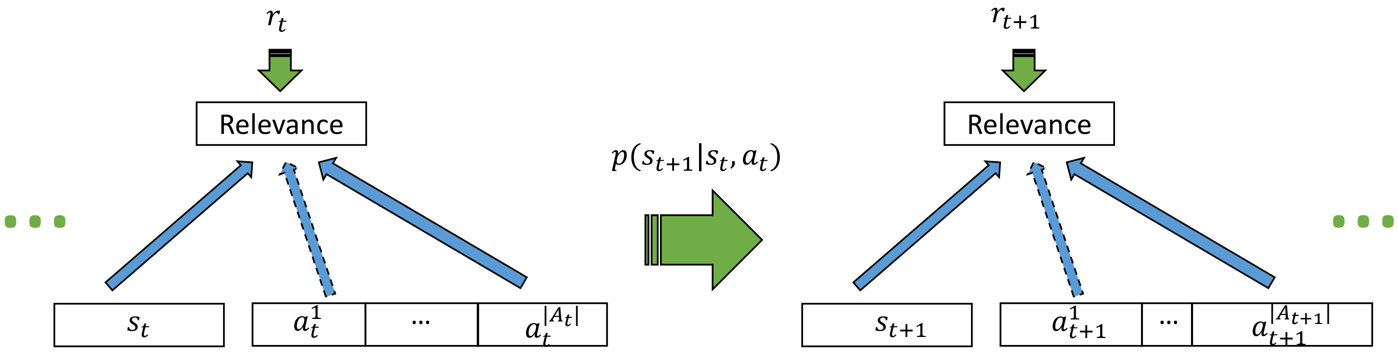
The cat is out here with you, parked directly in front of the door and looking up at you expectantly.

- **Step purposefully over the cat and into the lobby**
- **Return the cat's stare**
- **"Howdy, Mittens."**

Example: a snapshot of a text-based game

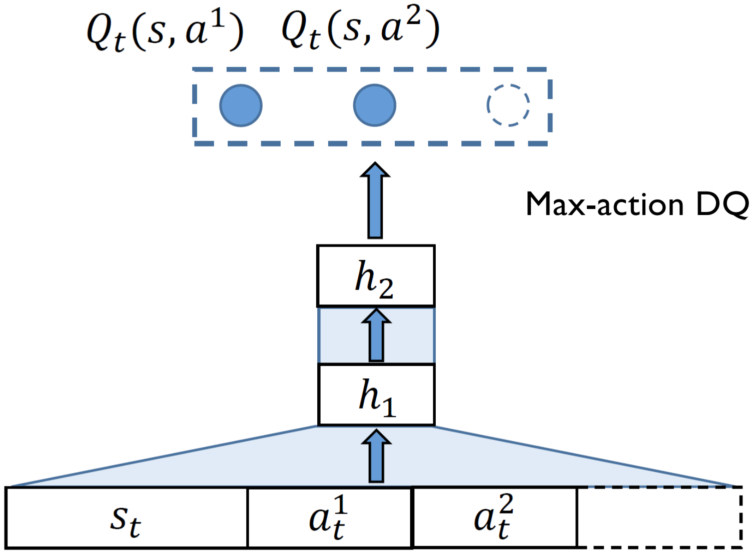
The Reinforcement Learning for NL problem

- RL for text understanding
  - Unbounded state and action spaces (both in texts)
  - Time-varying feasible action set
    - At each time, the actions are different texts.
    - At each time, the number of actions are different.

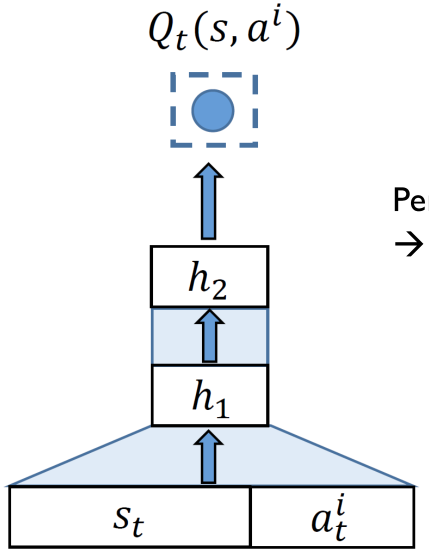


Baselines: Variants of Deep Q-Network

- Q-function: using a single deep neural network as function approximation
- Input: concatenated state-actions (BoW)
- Output: Q-values for different actions



Max-action DQN

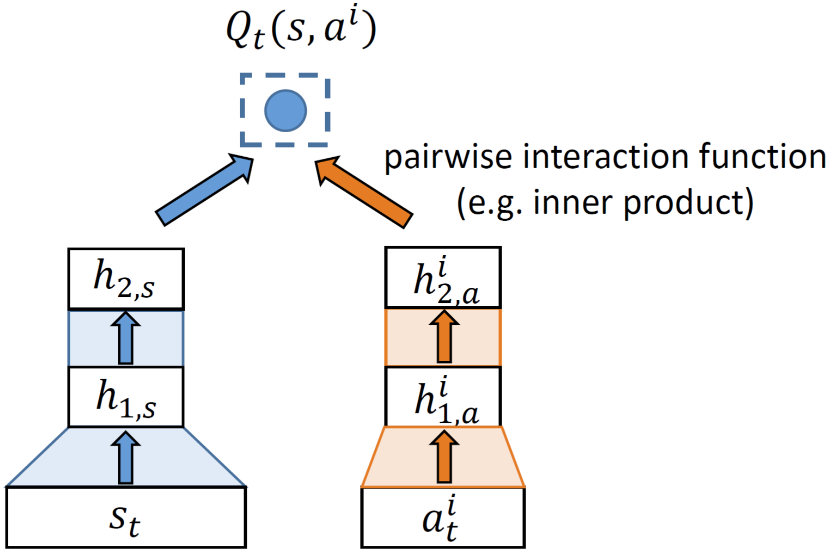


Per-action DQN
→ max over a_t^i

Deep Reinforcement Relevance Network (DRRN)

- **Similar to the DSSM (deep structured semantic model), project both *s* and *a* into a continuous space**
  - Separate state and action embeddings
  - Interaction at the embedding space
  - $Q(s, a^i; \Theta) = g\left(h_{L,s},\ h^i_{L,a}\right)$

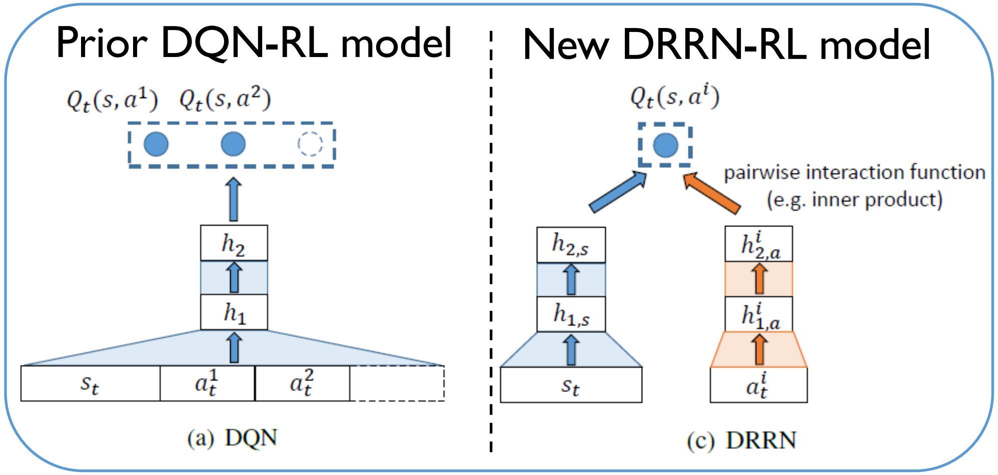
Motivation:
- Language is different in these two contexts.
- Text similarity does NOT always lead to the best action.



[Huang, He, Gao, Deng, Acero, Heck, 2013. "Learning Deep Structured Semantic Models for for Web Search using Clickthrough Data," CIKM]; [He, Chen, He, Gao, Li, Deng, Ostendorf, 2016. "Deep Reinforcement Learning with a Natural Language Action Space," ACL]

Reflection: DRRN

- **Prior DQN** work (e.g., Atari game, AlphaGo): state space unbounded, **action space bounded**.
- **In NLP** tasks, usually **the action space is unbounded** since it is characterized by natural language, which is discrete and nearly unconstrained.
- **New DRRN: (**Deep Reinforcement Relevance Network**)**
  - Project both the state and the action into a continuous space
  - Q-function is an relevance function of the state vector and the action vector

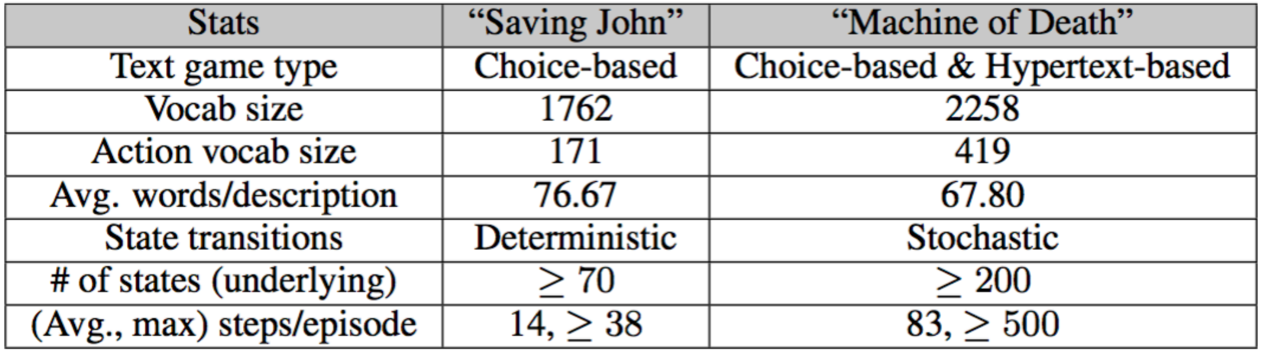


(a) DQN

(c) DRRN

Experiments: Tasks

- Two text games



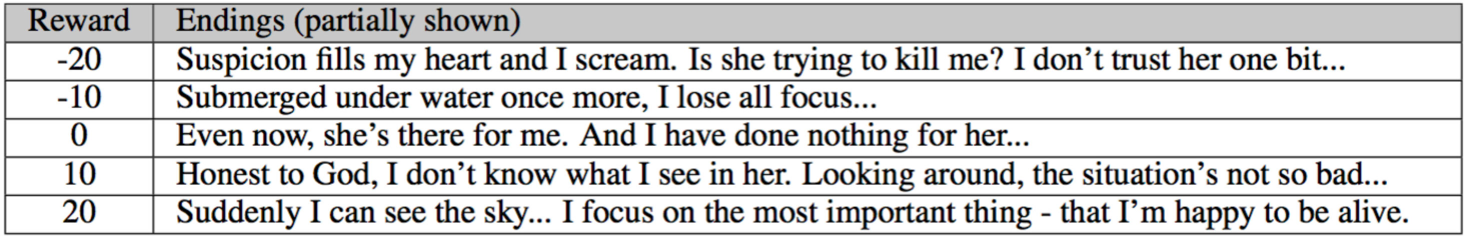
Stats	"Saving John"	"Machine of Death"
Text game type	Choice-based	Choice-based & Hypertext-based
Vocab size	1762	2258
Action vocab size	171	419
Avg. words/description	76.67	67.80
State transitions	Deterministic	Stochastic
# of states (underlying)	≥ 70	≥ 200
(Avg., max) steps/episode	$14, \geq 38$	$83, \geq 500$

- Hand annotate rewards for distinct endings
  - Simulators available at: https://github.com/jvking/text-games

Experiments

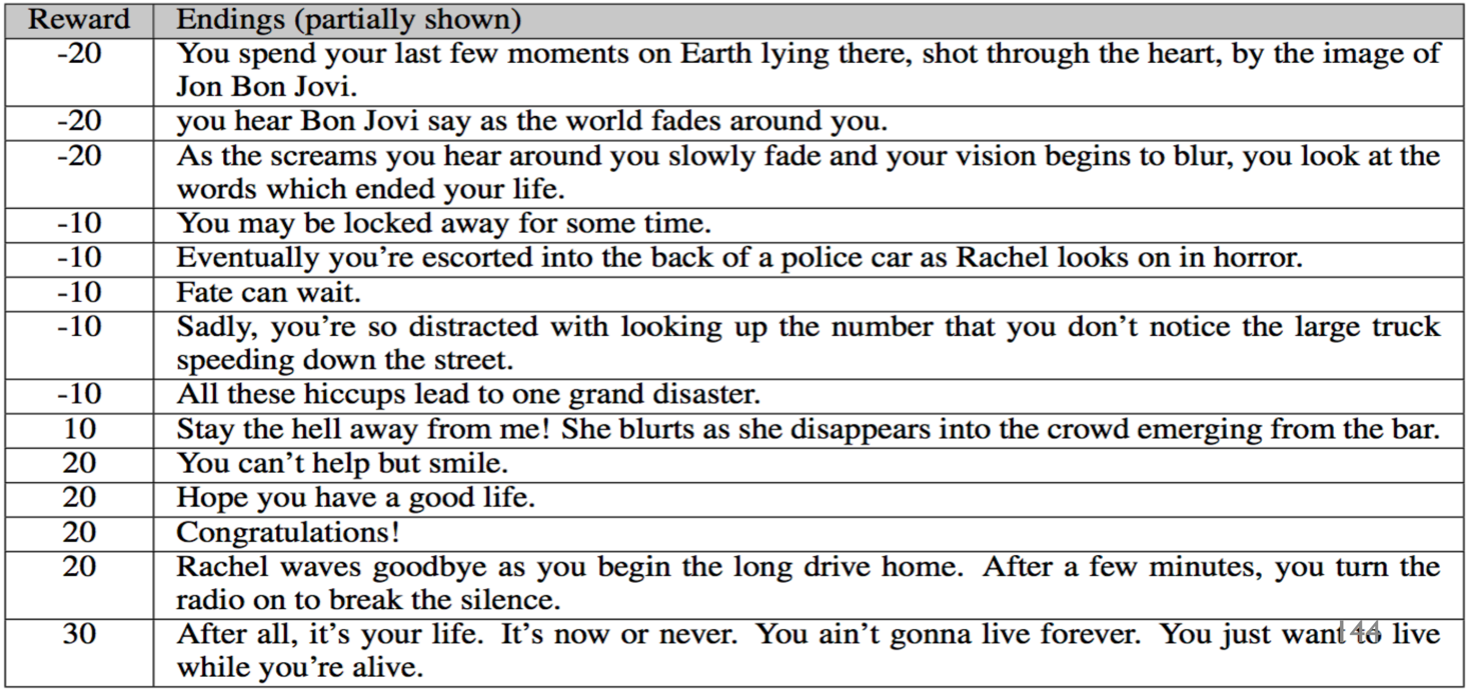
- Tasks: Text Games/Interactive Fictions

Task 1:
"Save John"



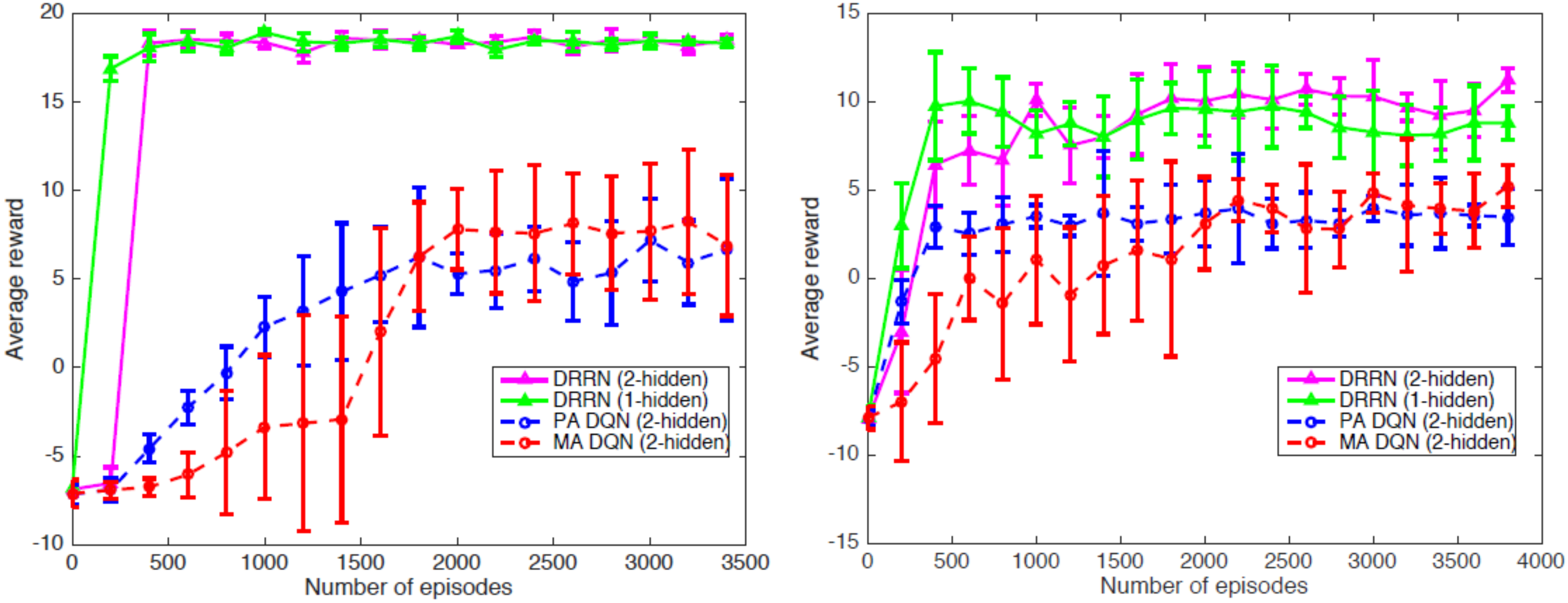
Reward	Endings (partially shown)
-20	Suspicion fills my heart and I scream. Is she trying to kill me? I don't trust her one bit...
-10	Submerged under water once more, I lose all focus...
0	Even now, she's there for me. And I have done nothing for her...
10	Honest to God, I don't know what I see in her. Looking around, the situation's not so bad...
20	Suddenly I can see the sky... I focus on the most important thing - that I'm happy to be alive.

Task 2:
"Machine of Death"



Reward	Endings (partially shown)
-20	You spend your last few moments on Earth lying there, shot through the heart, by the image of Jon Bon Jovi.
-20	you hear Bon Jovi say as the world fades around you.
-20	As the screams you hear around you slowly fade and your vision begins to blur, you look at the words which ended your life.
-10	You may be locked away for some time.
-10	Eventually you're escorted into the back of a police car as Rachel looks on in horror.
-10	Fate can wait.
-10	Sadly, you're so distracted with looking up the number that you don't notice the large truck speeding down the street.
-10	All these hiccups lead to one grand disaster.
10	Stay the hell away from me! She blurts as she disappears into the crowd emerging from the bar.
20	You can't help but smile.
20	Hope you have a good life.
20	Congratulations!
20	Rachel waves goodbye as you begin the long drive home. After a few minutes, you turn the radio on to break the silence.
30	After all, it's your life. It's now or never. You ain't gonna live forever. You just want to live while you're alive.

Learning curve: DRRN vs. DQN



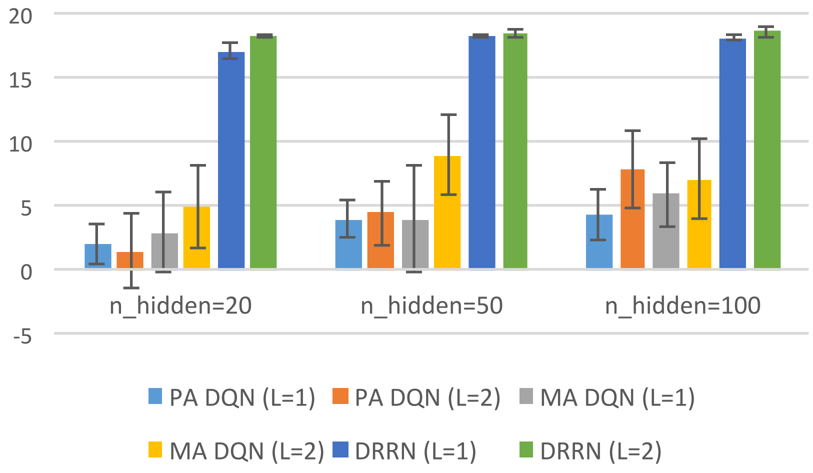
(a) Game 1: "Saving John"

(b) Game 2: "Machine of Death"

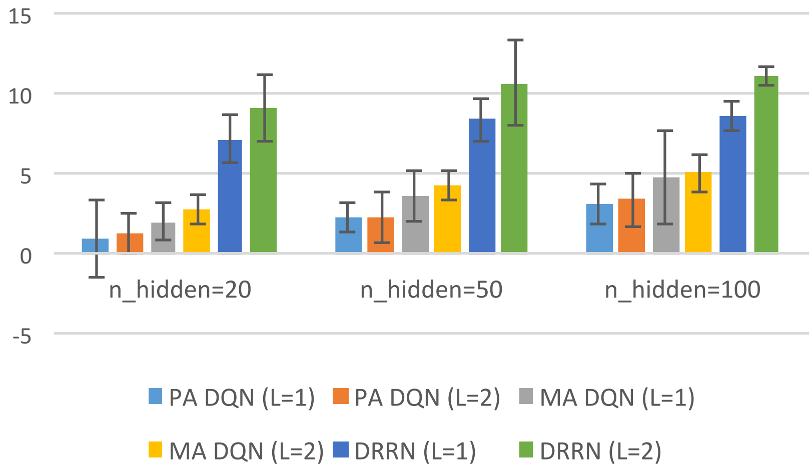
Tested on two text games

Experiments: Final Performance

Game 1:“Saving John”



Game 2:“Machine of Death”



The DRRN performs consistently better than all baselines, and often with a lower variance.
Big gain from having separate state & action embedding spaces (DQN vs. DRRN).

Visualization of the learned continuous space

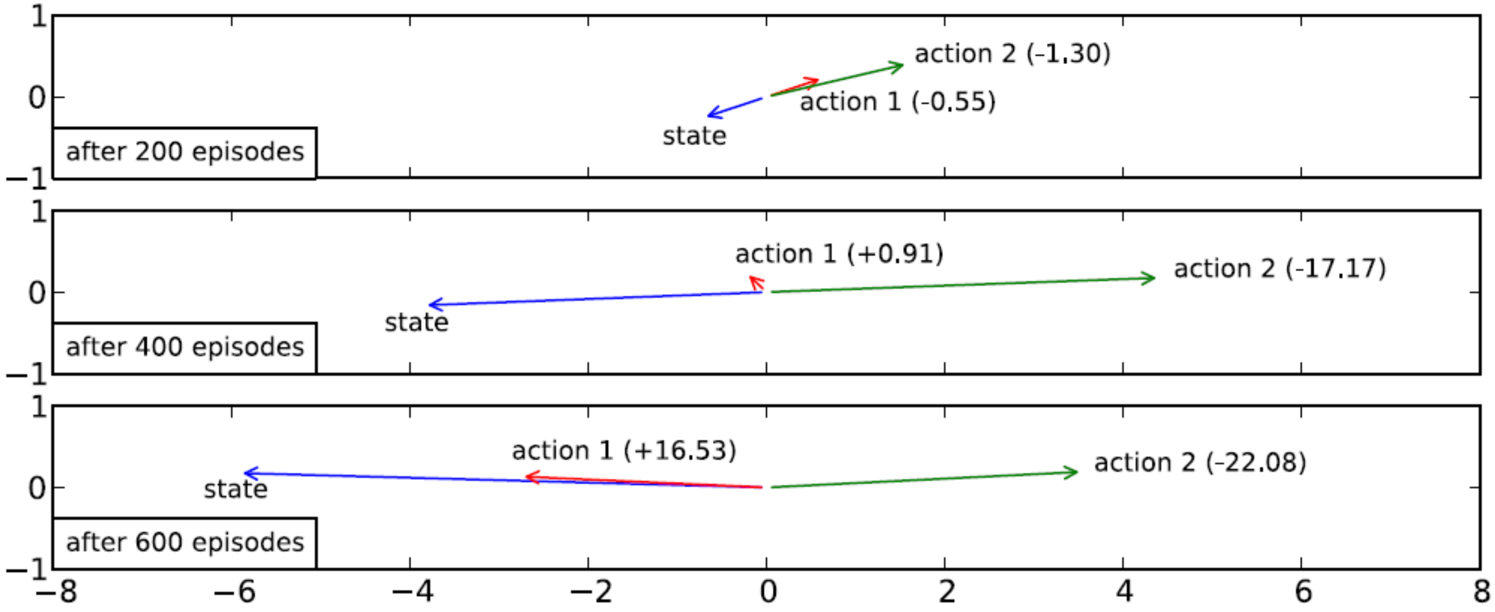
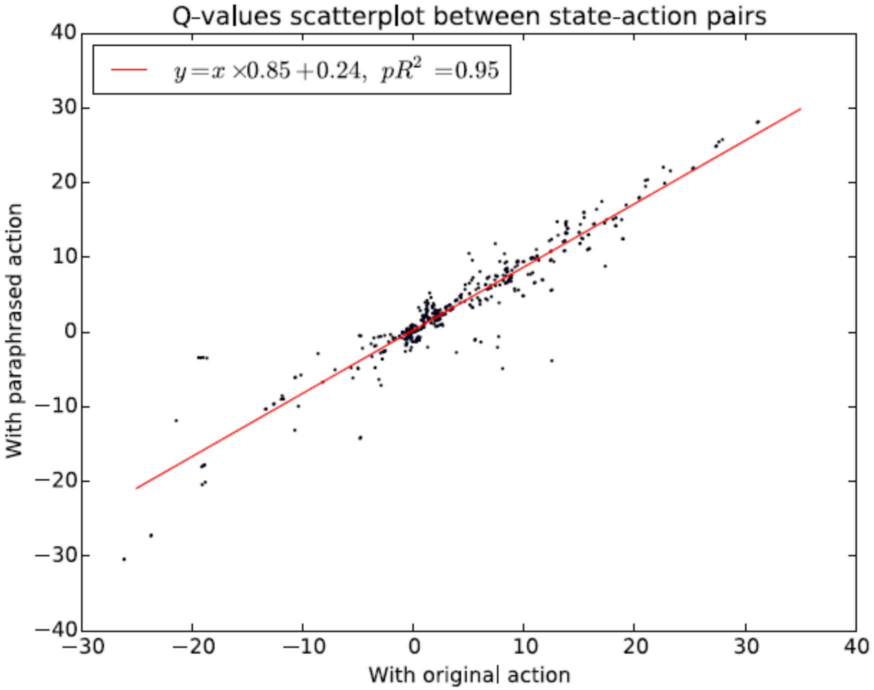


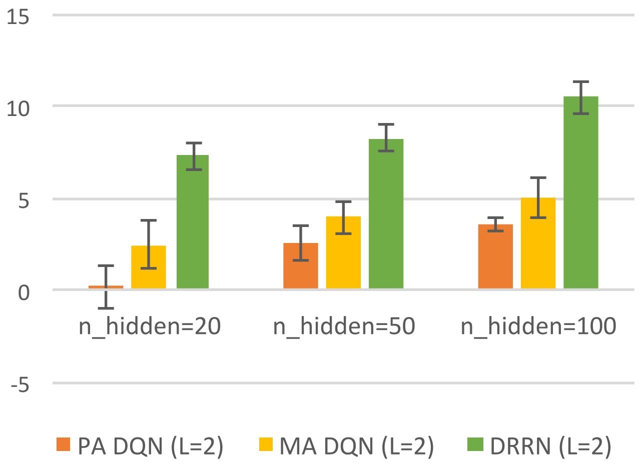
Figure 2: PCA projections of text embedding vectors for state and associated action vectors after 200, 400 and 600 training episodes. The state is “As you move forward, the people surrounding you suddenly look up with terror in their faces, and flee the street.” Action 1 (good choice) is “Look up”, and action 2 (poor choice) is “Ignore the alarm of others and continue moving forward.”

Experiments: Generalization

- In the testing stage, use **unseen** paraphrased actions



Game 2: "Machine of Death"



Q-function example values after converged

	Text (with predicted Q-values)
State	As you move forward, the people surrounding you suddenly look up with terror in their faces, and flee the street.
Actions in the original game	Ignore the alarm of others and continue moving forward. (-21.5) Look up. (16.6)
Paraphrased actions (not original)	Disregard the caution of others and keep pushing ahead. (-11.9) Turn up and look. (17.5)
Fake actions (not original)	Stay there. (2.8) Stay calmly. (2.0) Screw it. I'm going carefully. (-17.4) Yell at everyone. (-13.5) Insert a coin. (-1.4) Throw a coin to the ground. (-3.6)

Note that, the DRRN generalizes to unseen actions well, e.g., for these "not original" actions, the model still gives a proper estimate of the Q-value.

From games to large scale real-world scenarios

- Task:

  Build an agent runs on real world ***Reddit*** dataset
  https://www.reddit.com/

  reads Reddit posts

  recommends threads in *real time* with most future popularity

- Approach:
  - RL with specially designed Q-function for combinatorial action spaces

Motivation

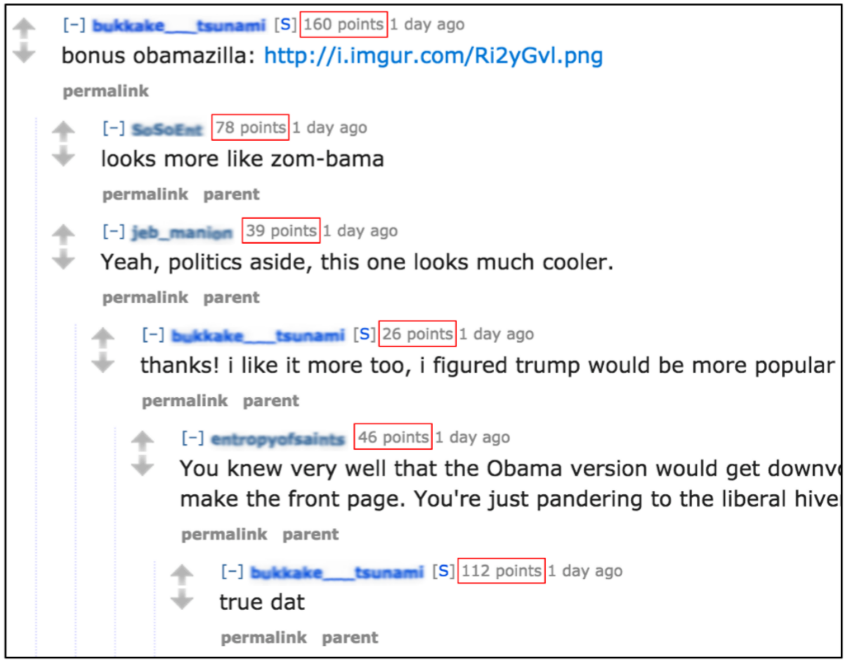
- we consider Reddit popularity prediction, which is different to newsfeed recommendation in two respects:
  - Making recommendations based on the anticipated long-term interest level of a broad group of readers from a target community, rather than for individuals.
  - Community interest level is not often immediately clear -- there is a time lag before the level of interest starts to take off. Here, *the goal is recommendation in **real time** – attempting to identify hot updates before they become hot to keep the reader at the leading edge.*

Solution

- Problem fits reinforcement learning paradigm
- ***Combinatorial*** action space
  - Sub-action is a post
  - Action is a set of interdependent documents
    - Two problems: i) potentially high computational complexity, ii) estimating the long-term reward (the Q-value in reinforcement learning) from a combination of sub-actions characterized by natural language.
    - The paper focuses on (ii).

Problem Setting

- Registered Reddit users initiate a post and people respond with comments, Together, the comments and the original post form a **discussion tree**.
- Comments (and posts) are associated with positive and negative votes (i.e., likes and dislikes) that are combined to get a **karma score**, which can be used as a measure for popularity.
- As in Fig 1., it is **quite common that a lower karma comment will lead to more children and popular comments in the future** (e.g. "true dat").
- In a real-time comment recommendation system, the eventual karma of a comment is not immediately available, so **prediction of popularity is based on the text** in the comment in the context of prior comments in the subtree and other comments in the current time window.



[–] [redacted] [S] 160 points 1 day ago

bonus obamazilla: http://i.imgur.com/Ri2yGvl.png

permalink

[–] [redacted] 78 points 1 day ago

looks more like zom-bama

permalink parent

[–] [redacted] 39 points 1 day ago

Yeah, politics aside, this one looks much cooler.

permalink parent

[–] [redacted] [S] 26 points 1 day ago

thanks! i like it more too, i figured trump would be more popular

permalink parent

[–] [redacted] 46 points 1 day ago

You knew very well that the Obama version would get downv
make the front page. You're just pandering to the liberal hive

permalink parent

[–] [redacted] [S] 112 points 1 day ago

true dat

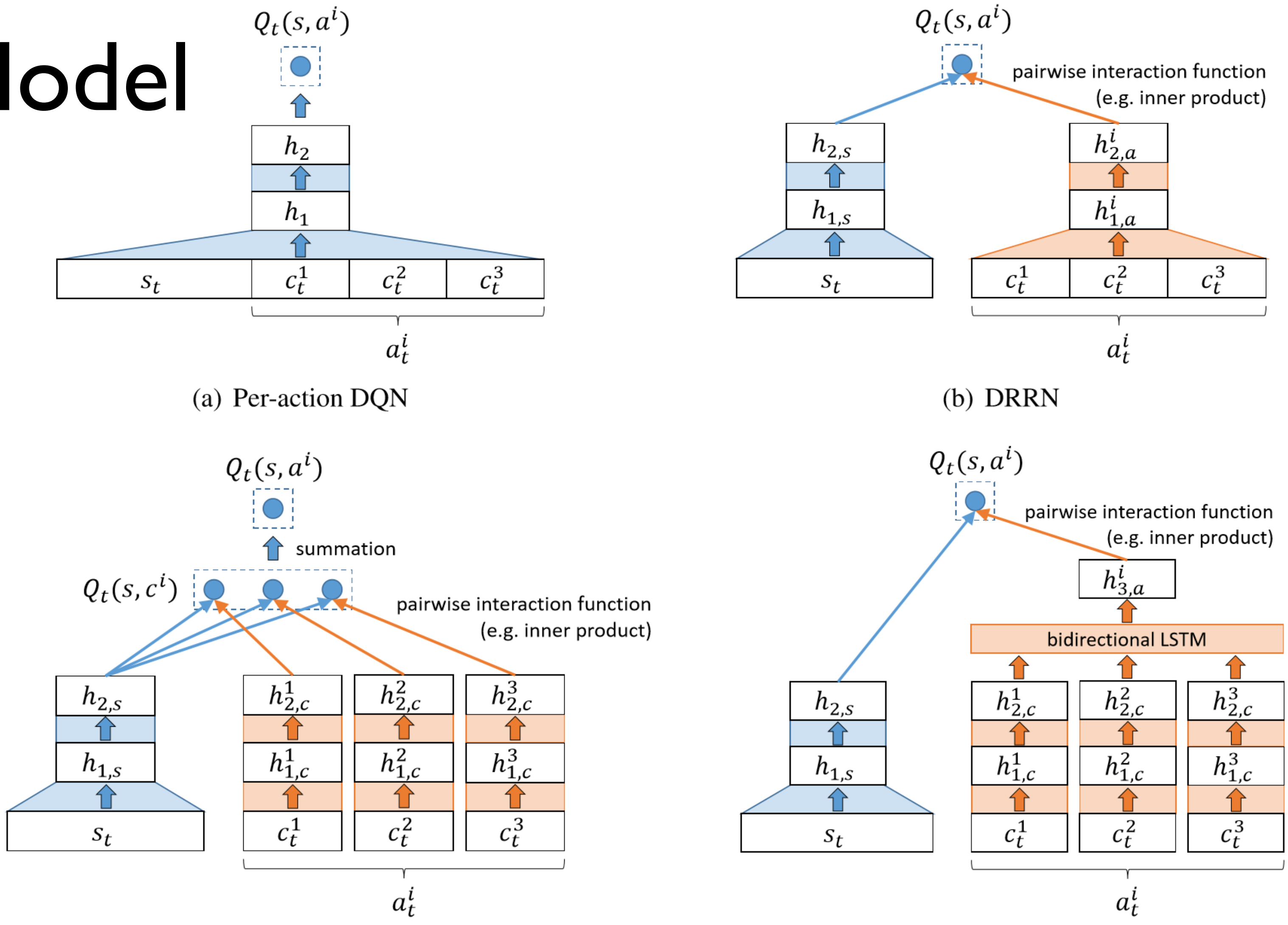
permalink parent

Figure 1: A snapshot of the top of a Reddit discussion tree, where karma scores are shown in red boxes.

Solution

- State
  - the collection of comments previously recommended.
- Action
  - Picking a new set of comments. Note that we only consider new comments associated with the threads of the discussion that we are currently following with the assumption that prior context is needed to interpret the comments.
- Reward
  - Long term Reddit voting scores, e.g., Karma scores after the thread settles down.
- Environment
  - The partially observed discussion tree

Model



(a) Per-action DQN

(b) DRRN

(c) DRRN-Sum

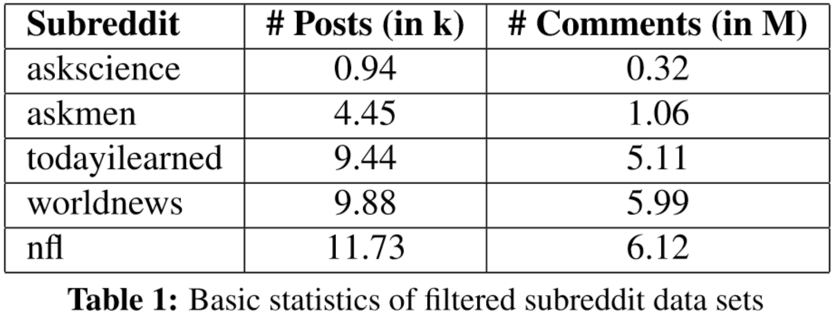
(d) DRRN-BiLSTM

Figure 2: Different deep Q-learning architectures

[He, Ostendorf, He, Chen, Gao, Li, Deng, 2016. "Deep Reinforcement Learning with a Combinatorial Action Space for Predicting Popular Reddit Threads," EMNLP]

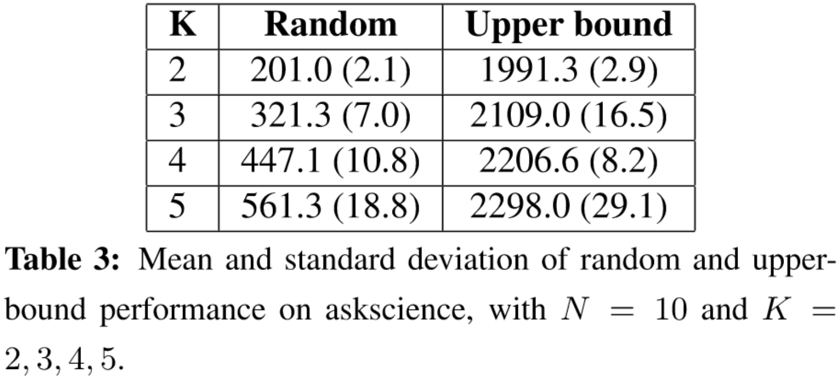
Experiments

- Data and stats



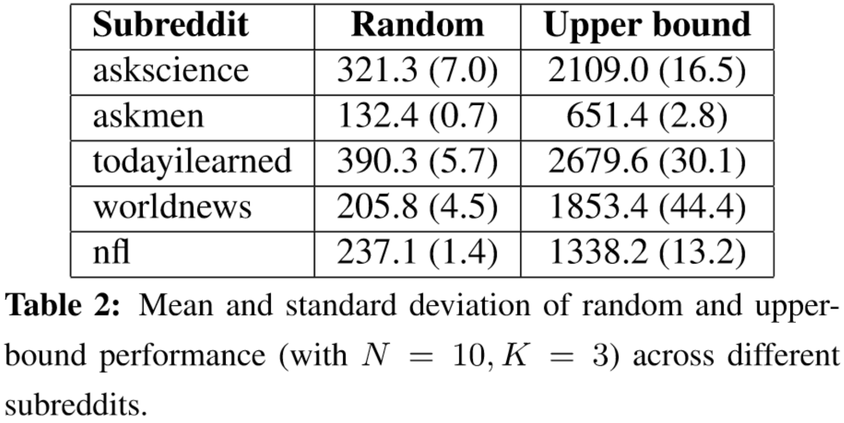
Subreddit	# Posts (in k)	# Comments (in M)
askscience	0.94	0.32
askmen	4.45	1.06
todayilearned	9.44	5.11
worldnews	9.88	5.99
nfl	11.73	6.12

Table 1: Basic statistics of filtered subreddit data sets



K	Random	Upper bound
2	201.0 (2.1)	1991.3 (2.9)
3	321.3 (7.0)	2109.0 (16.5)
4	447.1 (10.8)	2206.6 (8.2)
5	561.3 (18.8)	2298.0 (29.1)

Table 3: Mean and standard deviation of random and upper-bound performance on askscience, with $N = 10$ and $K = 2, 3, 4, 5$.

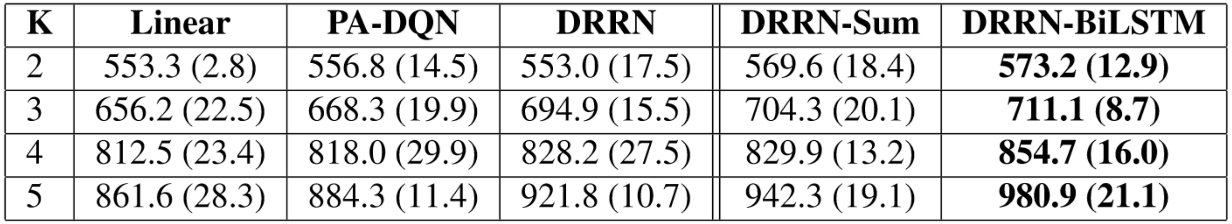


Subreddit	Random	Upper bound
askscience	321.3 (7.0)	2109.0 (16.5)
askmen	132.4 (0.7)	651.4 (2.8)
todayilearned	390.3 (5.7)	2679.6 (30.1)
worldnews	205.8 (4.5)	1853.4 (44.4)
nfl	237.1 (1.4)	1338.2 (13.2)

Table 2: Mean and standard deviation of random and upper-bound performance (with $N = 10, K = 3$) across different subreddits.

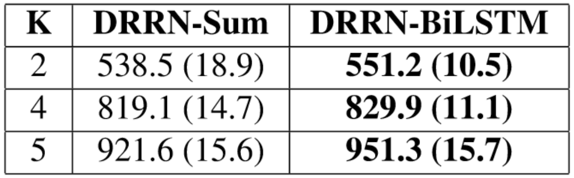
Results

- On the askscience sub-reddit



K	Linear	PA-DQN	DRRN	DRRN-Sum	DRRN-BiLSTM
2	553.3 (2.8)	556.8 (14.5)	553.0 (17.5)	569.6 (18.4)	**573.2 (12.9)**
3	656.2 (22.5)	668.3 (19.9)	694.9 (15.5)	704.3 (20.1)	**711.1 (8.7)**
4	812.5 (23.4)	818.0 (29.9)	828.2 (27.5)	829.9 (13.2)	**854.7 (16.0)**
5	861.6 (28.3)	884.3 (11.4)	921.8 (10.7)	942.3 (19.1)	**980.9 (21.1)**

Table 4: On askscience, average karma scores and standard deviation of baselines and proposed methods (with $N = 10$)

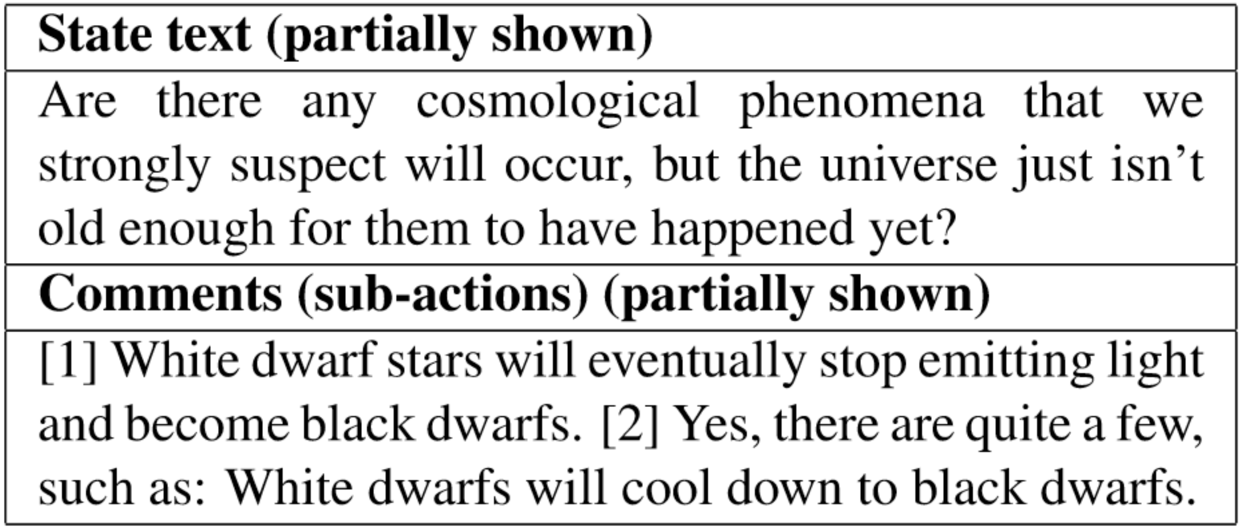


K	DRRN-Sum	DRRN-BiLSTM
2	538.5 (18.9)	**551.2 (10.5)**
4	819.1 (14.7)	**829.9 (11.1)**
5	921.6 (15.6)	**951.3 (15.7)**

Table 5: On askscience, average karma scores and standard deviation of proposed methods trained with $K = 3$ and test with different K's

Example

- In table 6, by combining the second sub-action compared to choosing just the first sub-action alone, DRRN-Sum and DRRN-BiLSTM predict 86% and 26% relative increase in action-value, respectively. Since these two sub-actions are highly redundant, we hypothesize DRRN-BiLSTM is better than DRRN-Sum at capturing interdependency between sub-actions.

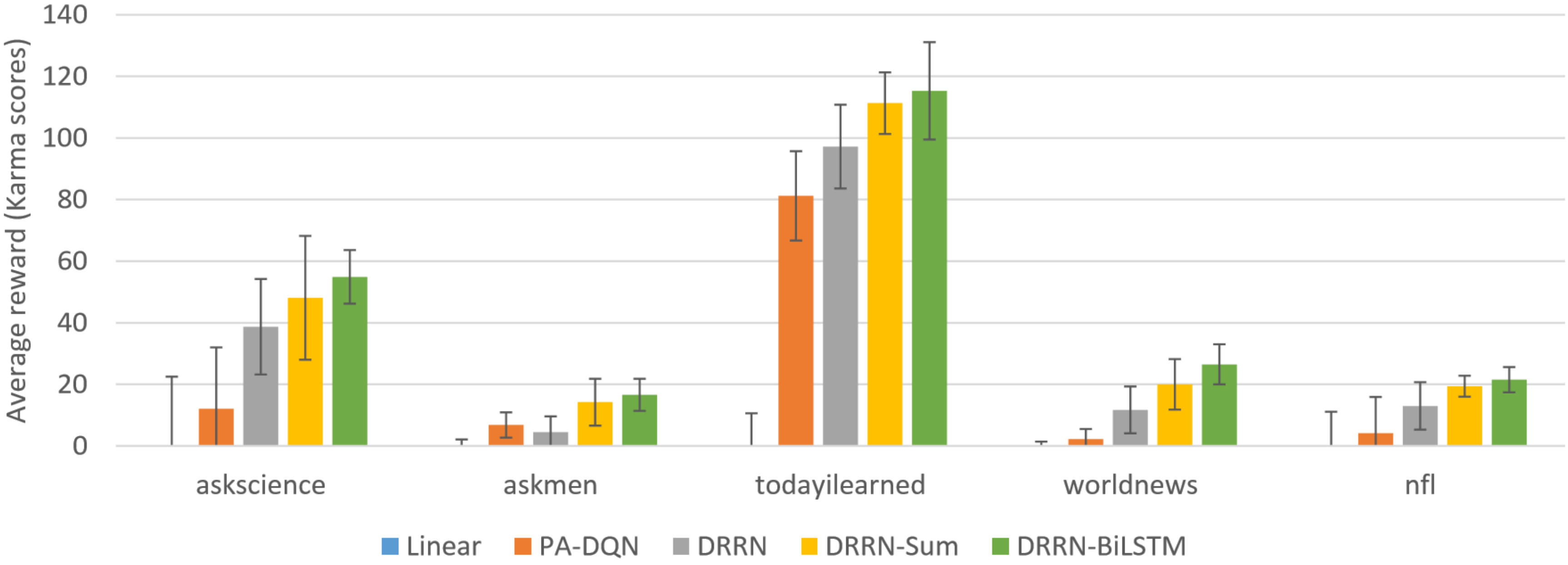


State text (partially shown)
Are there any cosmological phenomena that we strongly suspect will occur, but the universe just isn't old enough for them to have happened yet?
Comments (sub-actions) (partially shown)
[1] White dwarf stars will eventually stop emitting light and become black dwarfs. [2] Yes, there are quite a few, such as: White dwarfs will cool down to black dwarfs.

Table 6: An example state and its sub-actions

Results on more sub-reddit domains

Average karma score gains over the baseline and standard deviation across different subreddits (N = 10,K = 3)



Incorporating External Knowledge

- In many NLP tasks such as Reddit post understanding, external knowledge (such as world knowledge) is helpful
- How to incorporate the knowledge into a RL framework is interesting
  - How to retrieve **complementary** knowledge to enrich the state?

Reinforcement Learning with External Knowledge

Retrieve external knowledge to augment a state-side representation

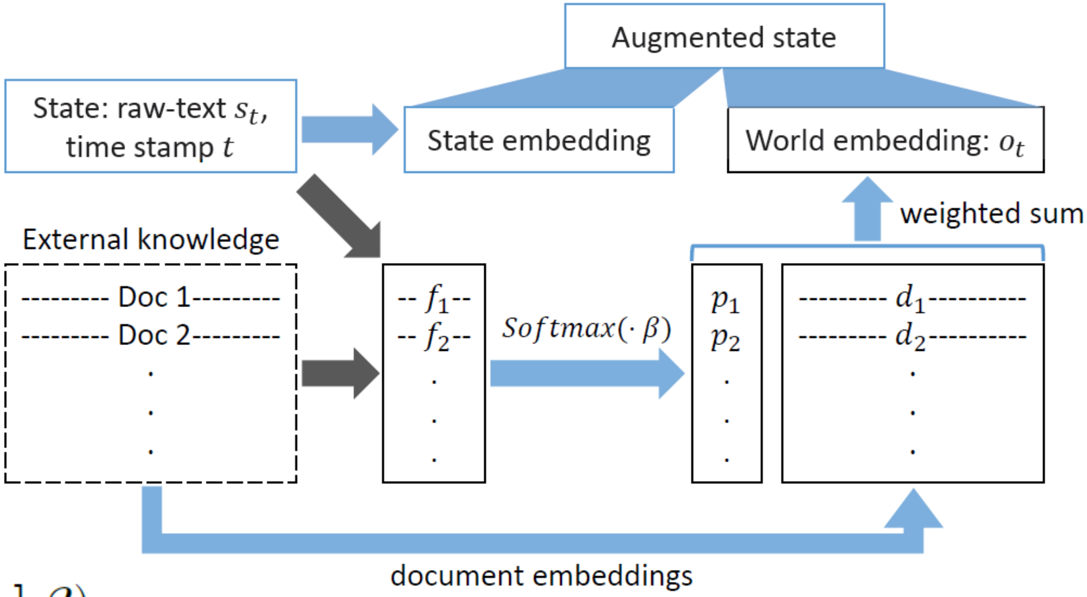
An attention-like approach is used

Not content-based retrieval

But **event-based knowledge retrieval**

Event features:

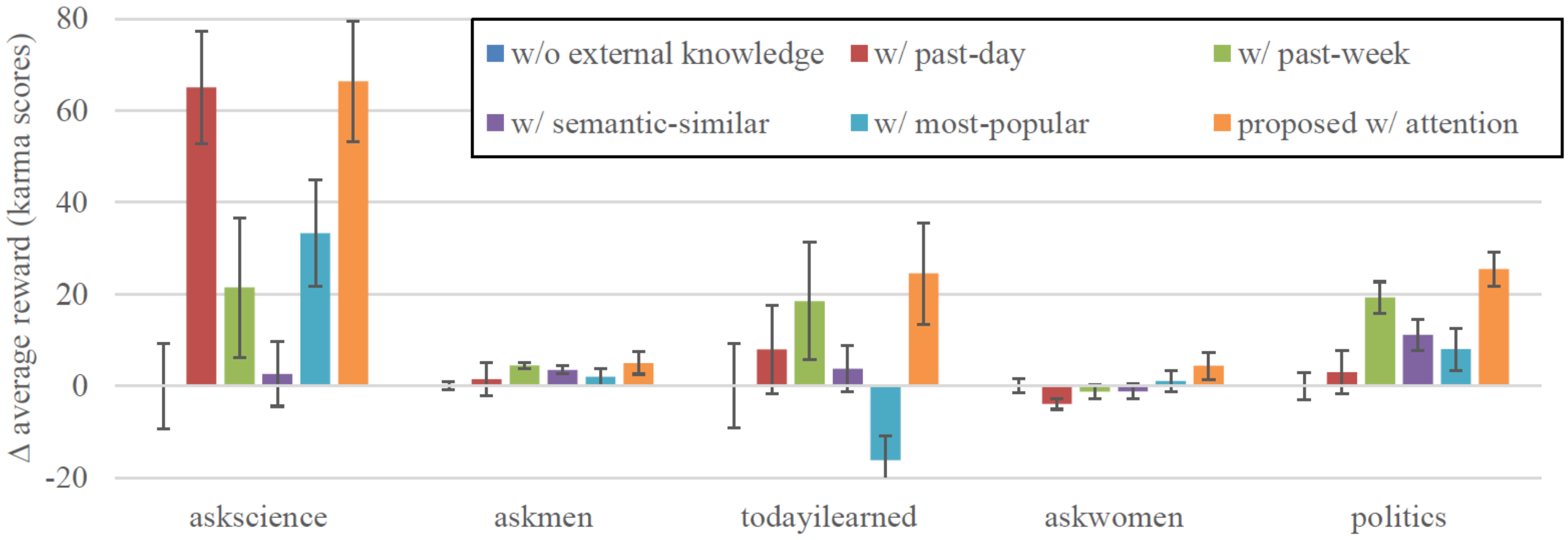
- Timing feature
- Semantic similarity
- Popularity



$$\boldsymbol{p} = \mathrm{Softmax}([\mathbf{1}_{\mathrm{day}}, \mathbf{1}_{\mathrm{wk}}, \boldsymbol{u}_{\mathrm{sem}}, \boldsymbol{u}_{\mathrm{pop}}]\cdot\boldsymbol{\beta})$$

[He, J., Ostendorf, M. and He, X., 2017. Reinforcement Learning with External Knowledge and Two-Stage Q-functions for Predicting Popular Reddit Threads. *arXiv:1704.06217*.]

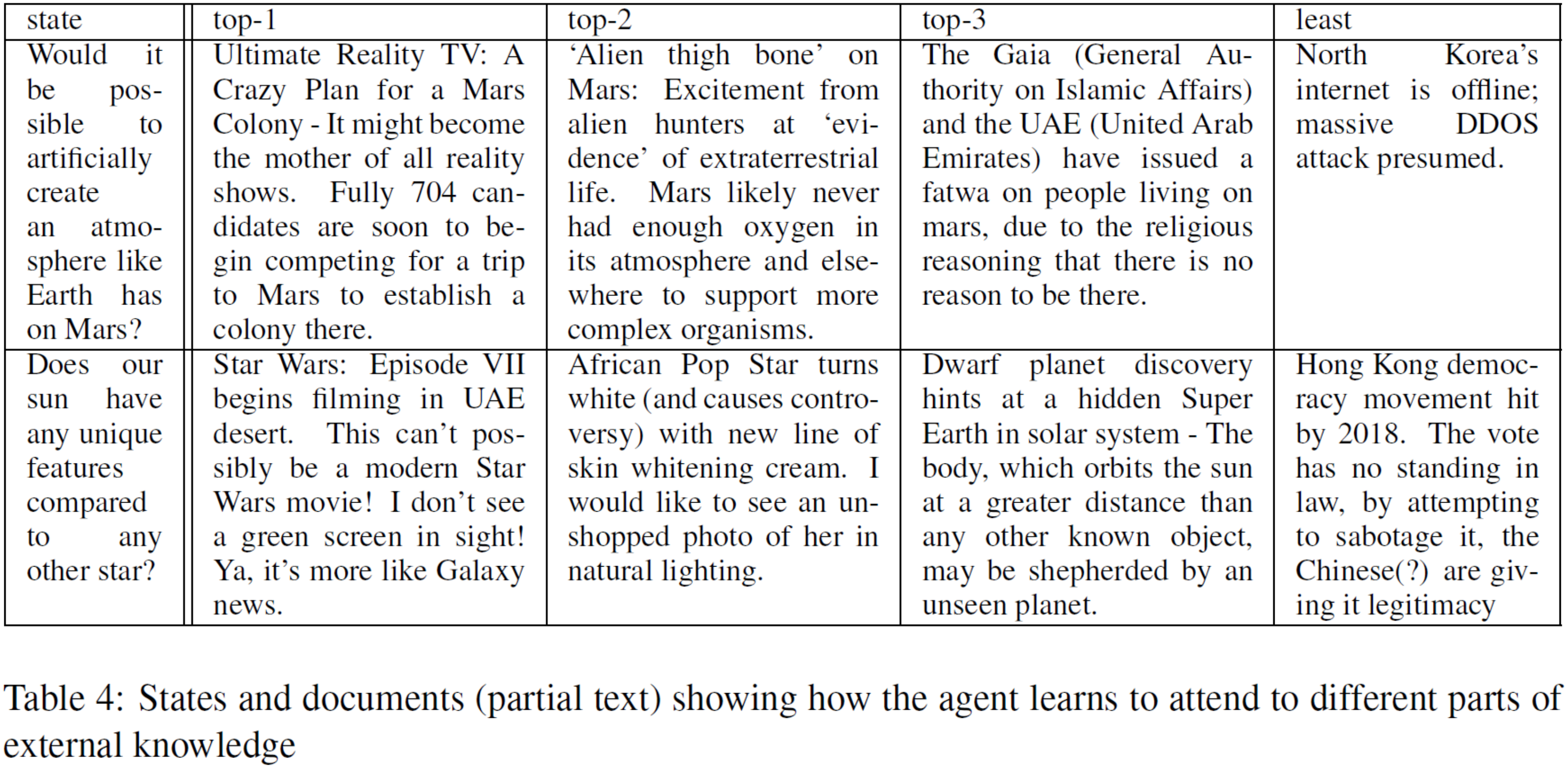
Incorporating external knowledge



DRRN (with different ways of incorporating knowledge) performance gains over baseline DRRN (without external knowledge) across 5 different subreddits

- External knowledge helps in general.
- The most useful knowledge not necessarily the most "semantically similar" knowledge!
- Event based knowledge retrieval is effective

Examples



state	top-1	top-2	top-3	least
Would it be possible to artificially create an atmosphere like Earth has on Mars?	Ultimate Reality TV: A Crazy Plan for a Mars Colony - It might become the mother of all reality shows. Fully 704 candidates are soon to begin competing for a trip to Mars to establish a colony there.	'Alien thigh bone' on Mars: Excitement from alien hunters at 'evidence' of extraterrestrial life. Mars likely never had enough oxygen in its atmosphere and elsewhere to support more complex organisms.	The Gaia (General Authority on Islamic Affairs) and the UAE (United Arab Emirates) have issued a fatwa on people living on mars, due to the religious reasoning that there is no reason to be there.	North Korea's internet is offline; massive DDOS attack presumed.
Does our sun have any unique features compared to any other star?	Star Wars: Episode VII begins filming in UAE desert. This can't possibly be a modern Star Wars movie! I don't see a green screen in sight! Ya, it's more like Galaxy news.	African Pop Star turns white (and causes controversy) with new line of skin whitening cream. I would like to see an unshopped photo of her in natural lighting.	Dwarf planet discovery hints at a hidden Super Earth in solar system - The body, which orbits the sun at a greater distance than any other known object, may be shepherded by an unseen planet.	Hong Kong democracy movement hit by 2018. The vote has no standing in law, by attempting to sabotage it, the Chinese(?) are giving it legitimacy

Table 4: States and documents (partial text) showing how the agent learns to attend to different parts of external knowledge

RL in Long Text Generation Tasks

Generating Recipes

"Grilled Cheese Sandwich"

Ingredients:

4 slices of white bread
2 slices of Cheddar cheese
3 tablespoons butter, divided

Recipe

- Preheat pan over medium heat.
- Generously butter one side of a slice of bread.
- Place bread butter-side-down onto skillet bottom and add 1 slice of cheese.
- Butter a second slice of bread on one side and place butter-side-up on top of sandwich.
- Grill until lightly browned and flip over; continue grilling until cheese is melted.

The challenges:

- Multi-sentence
- Weak correspondence between input and output
- Structural language requires correct order of events and aware of state changes!

Kiddon, Zettlemoyer, Choi. 2016. "Globally coherent text generation with neural checklist models." EMNLP

Challenges in Long Form Text Generation

Sequence to Sequence Training Methods:

- MLE
- RL (Policy gradient)
- GAN (!)

Issues:

- Designed for short form generation (e.g., MT or dialog response)
- Loss functions does not reflect high-level semantics for long form
- **Not direct metric optimization**, **exposure bias, credit assignment**, struggle maintaining **coherence**, objective function balancing, ……

RL has been applied in text generation -- the challenge, however, is to define a global score that can **measure the complex aspects of text quality** beyond local n-gram patterns.

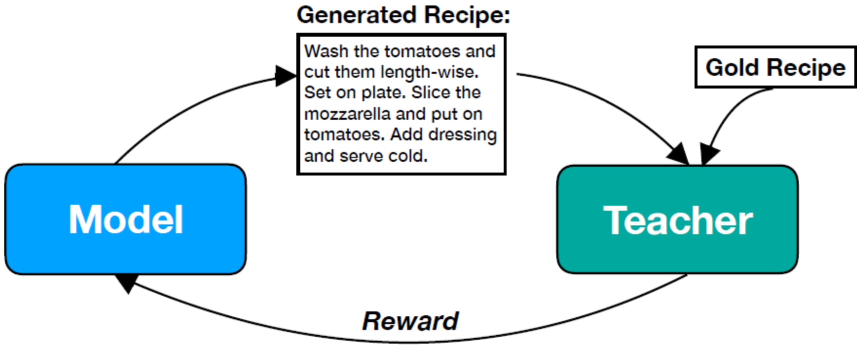
Neural Reward Functions for Long Form Text Generation

Goal:

- Capture individual **semantic properties** of the generation task
- Capture the **coherence** and **long-term dependencies** among sentences
- Generate **temporally correct** text

Approach:

- Use Policy Gradients
- Train Neural Reward functions as teachers
- Generate **task specific rewards**.
- Ensemble of rewards provide a better signal ?


Generated Recipe:
Wash the tomatoes and cut them length-wise. Set on plate. Slice the mozzarella and put on tomatoes. Add dressing and serve cold.
Gold Recipe
Model
Teacher
Reward

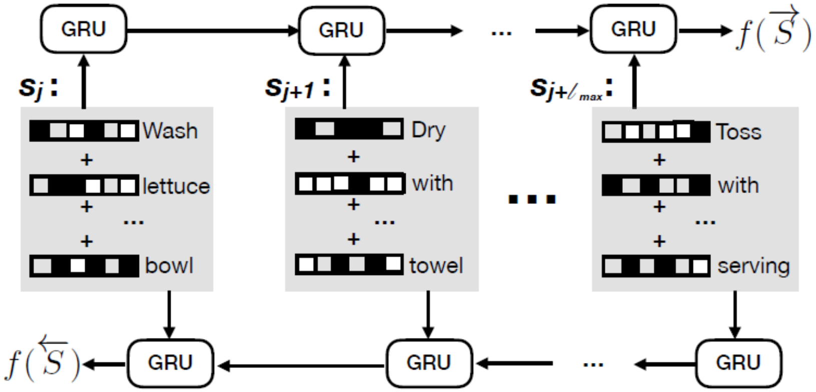
The generator is rewarded for imitating the discourse structure of the gold sequence

Bosselut, Celikyilmaz, Huang, He, Choi, 2018. *"Discourse-Aware Neural Rewards for Coherent Text Generation"*, NAACL

Train the teacher

Two neural teachers that can learn to score an ordered sequence of sentences.

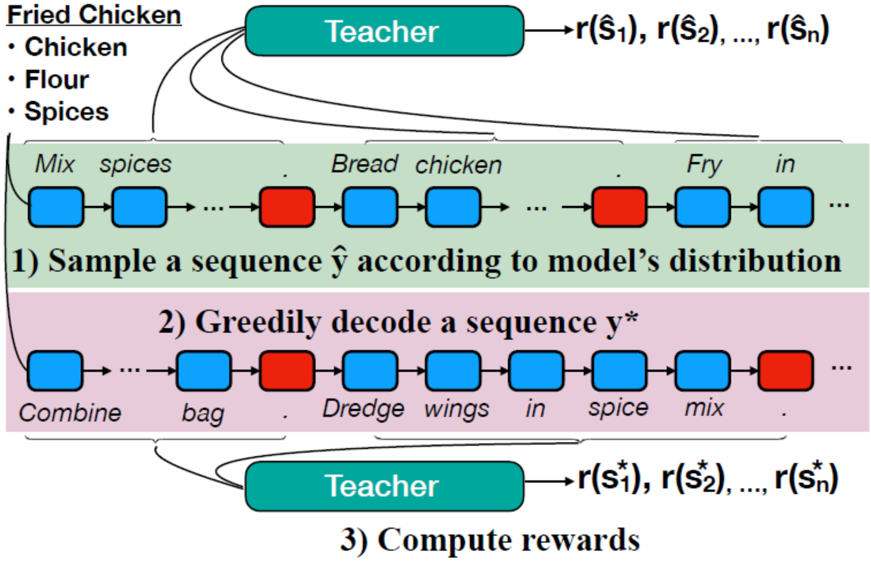
1. Absolute Order Teacher
    1. evaluates the temporal coherence of the entire generation
2. Relative Order Teacher
    1. reward how a sentence fits with surrounding sentences
3. A DSSM like architecture is used in implementation



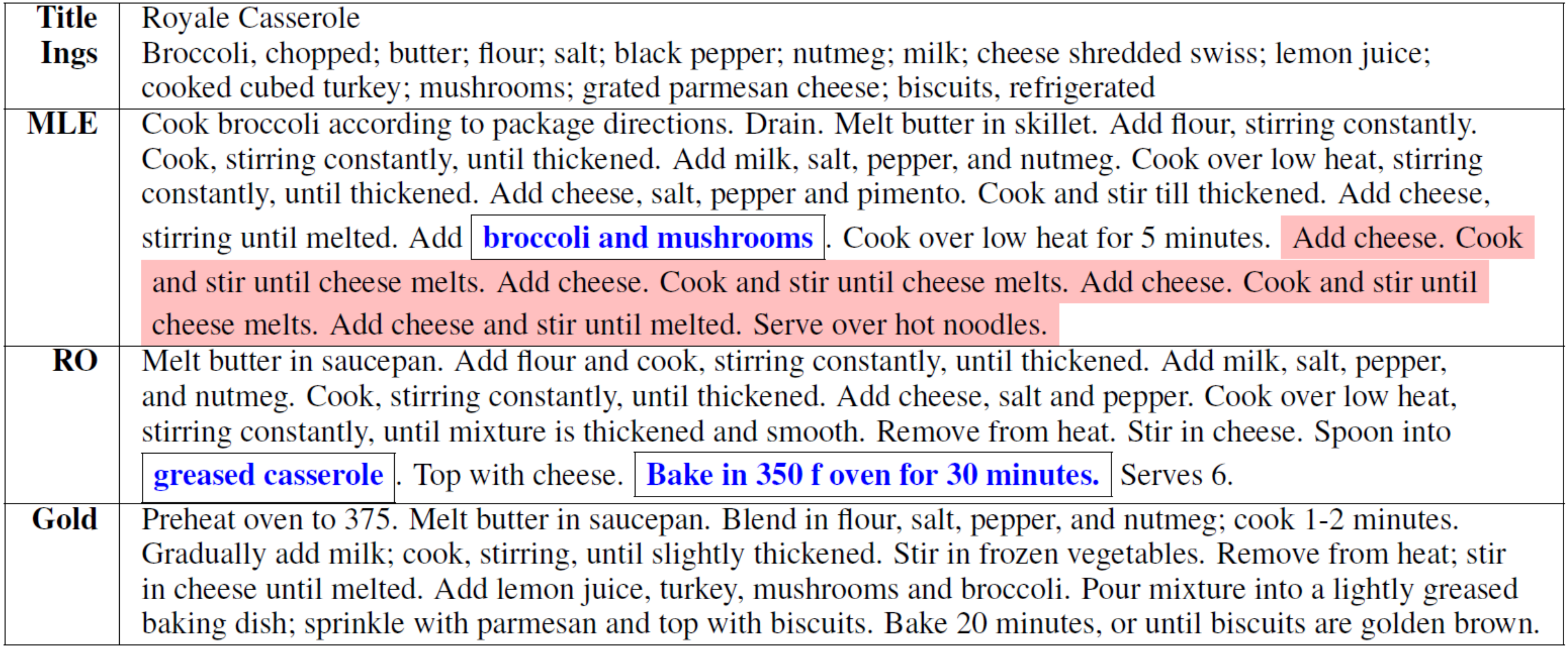
The teacher encodes the sentences of the document in the forward and reverse order

Policy Learning to optimize the reward

- The model generates a recipe by sampling
- Also greedily decodes a baseline recipe.
- the teacher yields a reward for each sentence


Fried Chicken
• Chicken
• Flour
• Spices
Teacher
r(ŝ1), r(ŝ2), ..., r(ŝn)
Mix spices . Bread chicken . Fry in ...
1) Sample a sequence ŷ according to model's distribution
2) Greedily decode a sequence y*
Combine bag . Dredge wings in spice mix .
Teacher
r(s*1), r(s*2), ..., r(s*n)
3) Compute rewards

Results



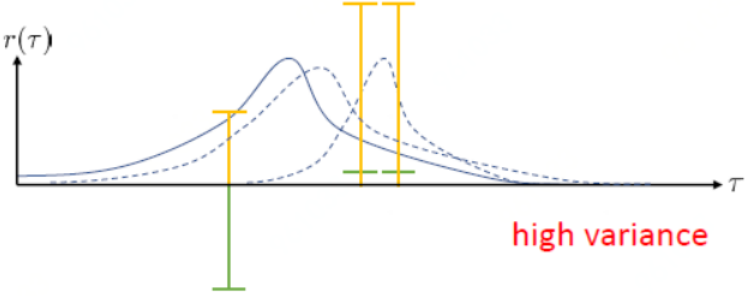
Title	Royale Casserole
Ings	Broccoli, chopped; butter; flour; salt; black pepper; nutmeg; milk; cheese shredded swiss; lemon juice; cooked cubed turkey; mushrooms; grated parmesan cheese; biscuits, refrigerated
MLE	Cook broccoli according to package directions. Drain. Melt butter in skillet. Add flour, stirring constantly. Cook, stirring constantly, until thickened. Add milk, salt, pepper, and nutmeg. Cook over low heat, stirring constantly, until thickened. Add cheese, salt, pepper and pimento. Cook and stir till thickened. Add cheese, stirring until melted. Add **broccoli and mushrooms** . Cook over low heat for 5 minutes. Add cheese. Cook and stir until cheese melts. Add cheese. Cook and stir until cheese melts. Add cheese. Cook and stir until cheese melts. Add cheese and stir until melted. Serve over hot noodles.
RO	Melt butter in saucepan. Add flour and cook, stirring constantly, until thickened. Add milk, salt, pepper, and nutmeg. Cook, stirring constantly, until thickened. Add cheese, salt and pepper. Cook over low heat, stirring constantly, until mixture is thickened and smooth. Remove from heat. Stir in cheese. Spoon into **greased casserole** . Top with cheese. **Bake in 350 f oven for 30 minutes.** Serves 6.
Gold	Preheat oven to 375. Melt butter in saucepan. Blend in flour, salt, pepper, and nutmeg; cook 1-2 minutes. Gradually add milk; cook, stirring, until slightly thickened. Stir in frozen vegetables. Remove from heat; stir in cheese until melted. Add lemon juice, turkey, mushrooms and broccoli. Pour mixture into a lightly greased baking dish; sprinkle with parmesan and top with biscuits. Bake 20 minutes, or until biscuits are golden brown.

Challenges and opportunities

- Open questions in RL that are important to NLP
  - Sample complexity
  - Model-based RL vs. Model-free RL
  - Acquiring rewards for many NLP tasks

Reducing Sample Complexity

- One of the core problems of RL: estimation with sampling.
- The problem: High variance and slow convergence

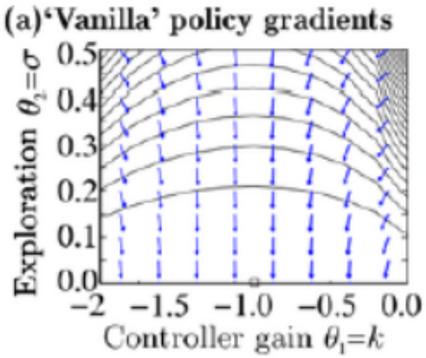


high variance

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \log \pi_\theta(\tau)\, r(\tau)$$

$$\log \pi_\theta(a_t|s_t) = -\frac{1}{2\sigma^2}(ks_t - a_t)^2 + const$$

$$r(s_t, a_t) = -s_t^2 - a_t^2$$



slow convergence

(image from Peters & Schaal 2008)

Reducing Sample Complexity

- Variance reduction using value function

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \, log \, \pi_\theta(\tau)[r(\tau) - b]$$

Subtracting a baseline is unbiased in expectation but reduce variance greatly!

$$E[\nabla_\theta \, log \, \pi_\theta(\tau) b] = \int \pi_\theta(\tau) \nabla_\theta \pi_\theta(\tau) b d\tau = \int \nabla_\theta \pi_\theta(\tau) b d\tau = b \nabla_\theta \int \pi_\theta(\tau) d\tau = b \nabla_\theta 1 = 0$$

Various forms for b

(1) $b = \frac{1}{N}\sum_{i=1}^{N} r(\tau)$

(2) $b = V^{\pi,\gamma}(s_t)$: $\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t^n | s_t^n) \, A^{\pi,\gamma}(s_t, a_t)$

[GAE, John Schulman et al.2016]

(3) $b = b(s_t, a_t)$: $\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t^n | s_t^n) \left(\hat{Q}_{n,t} - b(s_t^{\mathrm{n}}, a_t^n)\right) + \frac{1}{N}\sum_{n}^{N}\sum_{t}^{N} \nabla_\theta E_{a \sim \pi_\theta(a_t | s_t^n)}[b(s_t^n, a_t^n)]$

[Q-prop, Gu et al.2016]

Reducing Sample Complexity

Improve convergence rate via constrained optimization

$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$ $\pi_\theta(a_t|s_t)$

Problems with direct SGD: some parameters change probabilities a lot than others

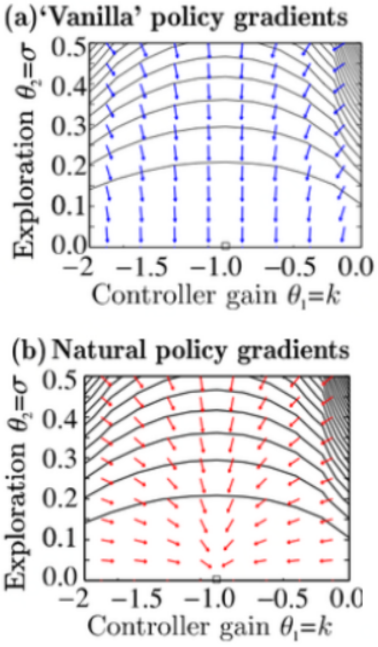
- ➢ Rescale the gradient with constraint divergence

$$\theta' \leftarrow \arg\max_{\theta'} (\theta' - \theta)^T \nabla_\theta J(\theta) \ \text{ s.t. } D_{KL}(\pi_{\theta'}, \pi_\theta) \leq \epsilon$$

$$D_{KL}(\pi_{\theta'}, \pi_\theta) = E_{\pi_{\theta'}}[\log \pi_{\theta'} - \log \pi_\theta] \approx (\theta' - \theta)^T F (\theta' - \theta)$$

$$F = E_{\pi_\theta}[\log \pi_\theta(a|s) \log \pi_\theta(a|s)^T]$$ ← Fisher-information matrix

Equivalence with natural gradient ! **[TRPO, Schulman et al.2015]**


(a)'Vanilla' policy gradients

(b) Natural policy gradients

- ➢ Use penalty instead of constraint

$$\min_\theta \sum_{n=1}^{N} \frac{\pi_\theta(a_n|s_n)}{\pi_{\theta_{old}}(a_n|s_n)} \hat{A}_n - \beta D_{KL}[\pi_{\theta_{old}}, \pi_\theta]$$

Increase/decrease β if KL is too high/low

[PPO, Schulman et al.2017]

Model-based v.s. Model-free RL

- Improve sample-efficiency via fast model-based RL

	Pros	Cons
Model-free RL	Handling arbitrary dynamic systems with minimal bias	Substantially less sample-efficient
Model-based RL	Sample-efficient planning when given accurate dynamics	Cannot handle unknown dynamical systems that might be hard to model

Can we combine model-based and model-free RL?

Guided policy search

$$\min_{\theta} \sum_{t=1}^{T} E_{\pi_\theta(x_t,u_t)}[l(x_t,u_t)]$$

$$\Downarrow$$

$$\min_{\theta,q_1,\dots,q_N} \sum_{i=1}^{N} \sum_{t=1}^{T} E_{q_i\,(x_t,u_t)}[l(x_t,u_t)] \;\; s.t\; q_i(u_t|x_t) = \pi_\theta(u_t|x_t) \;\; \forall x_t, u_t, t, i$$

$$\Uparrow$$

$$q_i(u_t|x_t) \sim N(k_t + K_t x_t | Q_{uut}^{-1})$$

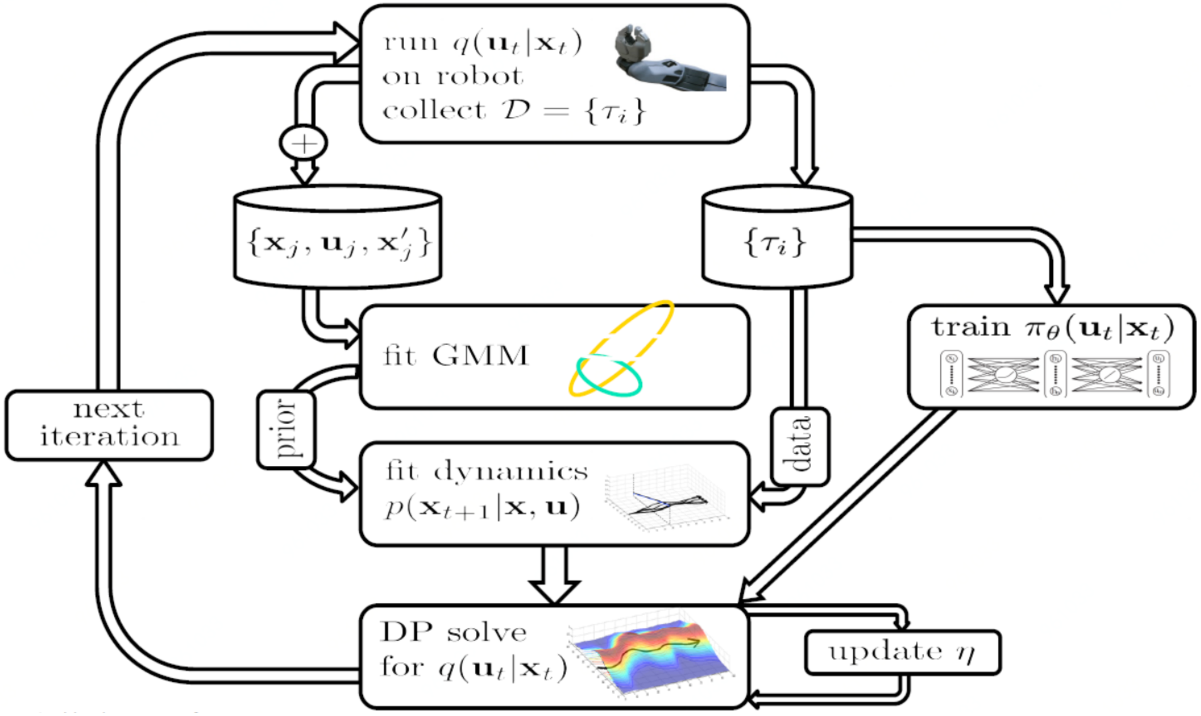
Planning $q_i(u_t|x_t)$ through a local approximate dynamics $p(x_{t+1}|x_t,u_t)$.

$$\Downarrow$$

Differentiable Dynamic Programming(DDP)

Model-based v.s. Model-free RL

- Improve sample-efficiency via fast model-based RL



run $q(\mathbf{u}_t|\mathbf{x}_t)$ on robot collect $\mathcal{D} = \{\tau_i\}$
$+$
$\{\mathbf{x}_j, \mathbf{u}_j, \mathbf{x}'_j\}$
$\{\tau_i\}$
fit GMM
train $\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)$
next iteration
prior
data
fit dynamics $p(\mathbf{x}_{t+1}|\mathbf{x}, \mathbf{u})$
DP solve for $q(\mathbf{u}_t|\mathbf{x}_t)$
update η

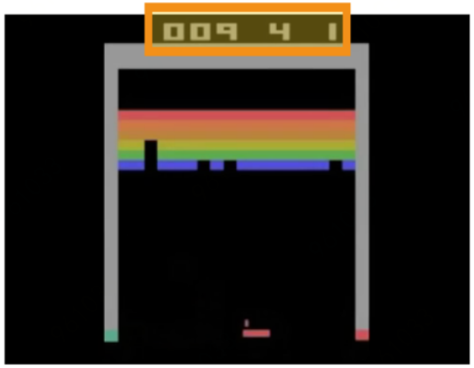
[Levine&Abbeel, NIPS 2014]

Acquiring Rewards

- How can we rewards for complex real-world tasks?

Computer Games

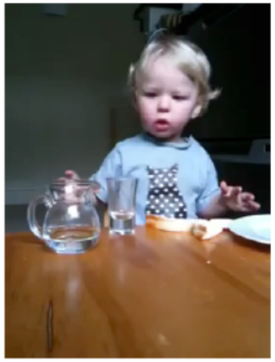
reward



Mnih et al. '15

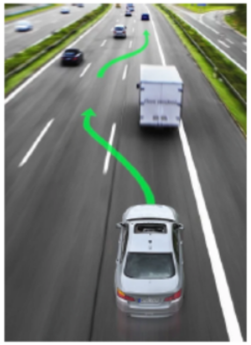
Real World Scenarios

robotics



dialogue

autonomous driving



*Many tasks are easier to provide expert data instead of reward function

Inverse RL: infer reward function from roll-outs of expert policy

Acquiring Rewards

- Inverse RL: infer reward function from demonstrations

[Kalman ’64, Ng & Russell ’00]

given:
- state & action space
- roll-out from π^*
- dynamics model[sometimes]

goal:
- Recover reward function
- then use reward to get policy

Challenges:
- underdefined problem
- difficult to evaluate a learned reward
- demonstrations may not be precisely optimal

- Newest works: combined with generative adversarial networks

Similar to inverse RL, GANs learn an objective for generative modeling

Zhu et al. ’17

Arjovsky et al. ’17

[Finn*, Christiano*, et al. ’16]

Acquiring Rewards

- Generative adversarial inverse RL

Inverse RL

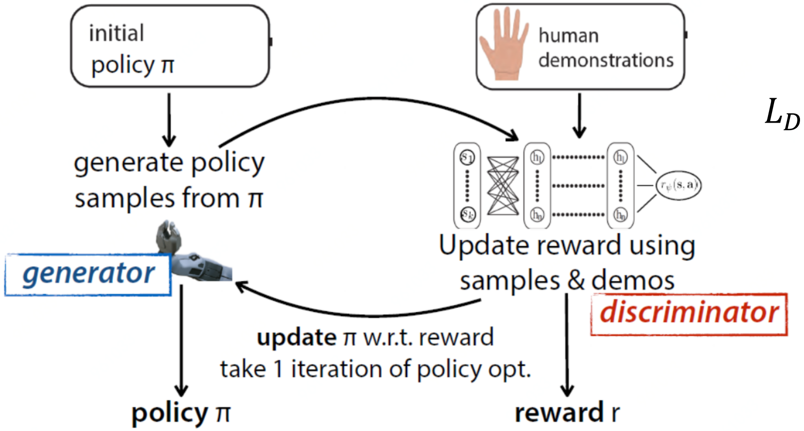
trajectory τ ⟷ **sample x**

policy π~q(τ) ⟷ **generator** G

reward r ⟶ **discriminator** D

data distribution p

GANs



update reward in inner loop of policy optimization

Reward/discriminator optimization

$$L_D(\psi) = E_{\tau\sim p}\left[-\log D_\psi(\tau)\right] + E_{\tau\sim q}\left[-\log(1 - D_\psi(\tau))\right]$$

Policy/Generator optimization

$$L_P(\theta) = E_{\tau\sim q}\left[\log(1 - D_\psi(\tau)) - \log(D_\psi(\tau))\right]$$

[Guided cost learning, Finn et al. ICML '16]
[GAIL, Ho & Ermon NIPS '16]

Session Summary

- Learn Q function in a common vector space for states and actions
- Add external knowledges to help NL understanding
- The reward could be learned to reflect the goal of long form text generation
- Open questions in RL that are important to NLP
  - Sample complexity
  - Model-based RL vs. Model-free RL
  - Acquiring rewards for many NLP tasks

Conclusion

- Deep Reinforcement Learning is a very natural solution for many NLP applications.
- DRL can be interpreted in many different ways.
- We have seen many exciting research directions.
- In particular, DRL for dialog is a very promising direction.
- Opportunities and challenges are ahead of us.

Questions?