Reinforcement Learning in Natural Language Processing (&Search)

Minlie Huang (黄民烈)

Dept. of Computer Science,

Tsinghua University

aihuang@tsinghua.edu.cn

http://coai.cs.tsinghua.edu.cn/hml

Reinforcement Learning

- **Sequential decision:** current decision affects future decision
- **Trial-and-error: just try, do not worry making mistakes**
  - **Explore** (new possibilities)
  - **Exploit** (with the current best policy)
- **Future reward:** maximizing the future rewards instead of just the intermediate rewards at each step

$$q_\pi(s,a) = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \,\middle|\, S_t = s, A_t = a, A_{t+1:\infty} \sim \pi\right]$$

$$q_\pi(s,a) = \mathbb{E}\left[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \,\middle|\, S_t = s, A_t = a, A_{t+1} \sim \pi\right]$$

Applying RL in NLP

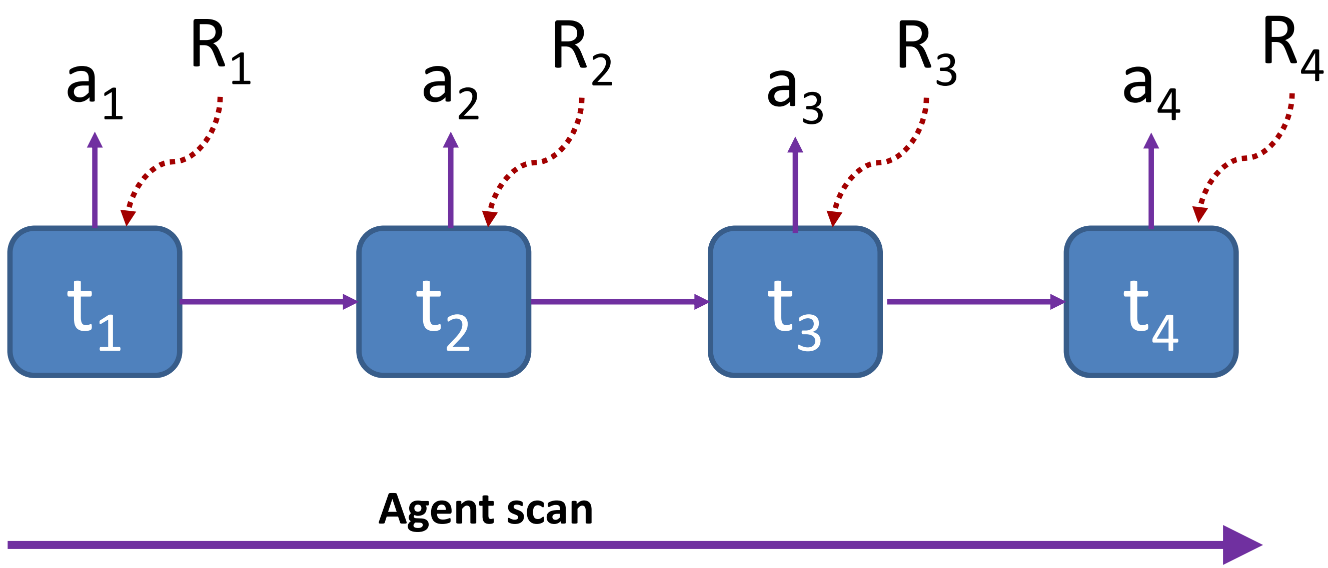
- **Challenges (**Sparse reward, high-dimensional action space, high variance in training**)**
  - **Discrete symbols**
  - **No simulator (or too expensive)**
- **Strengthens of RL**
  - **Weak supervision** without explicit annotations
  - **Trial-and-error**: probabilistic exploring
  - **Accumulative rewards**: encoding expert/prior knowledge in reward design

Why RL in NLP

- Learning to **search and reason**
- Directly optimize the **end metrics** (BLEU, ROUGE, Acc, F_1)
  - Machine translation, language generation, summarization
- Make discrete operations **BP-able**
  - Sampling
  - Argmax
  - Binary operations

Applying RL in NLP

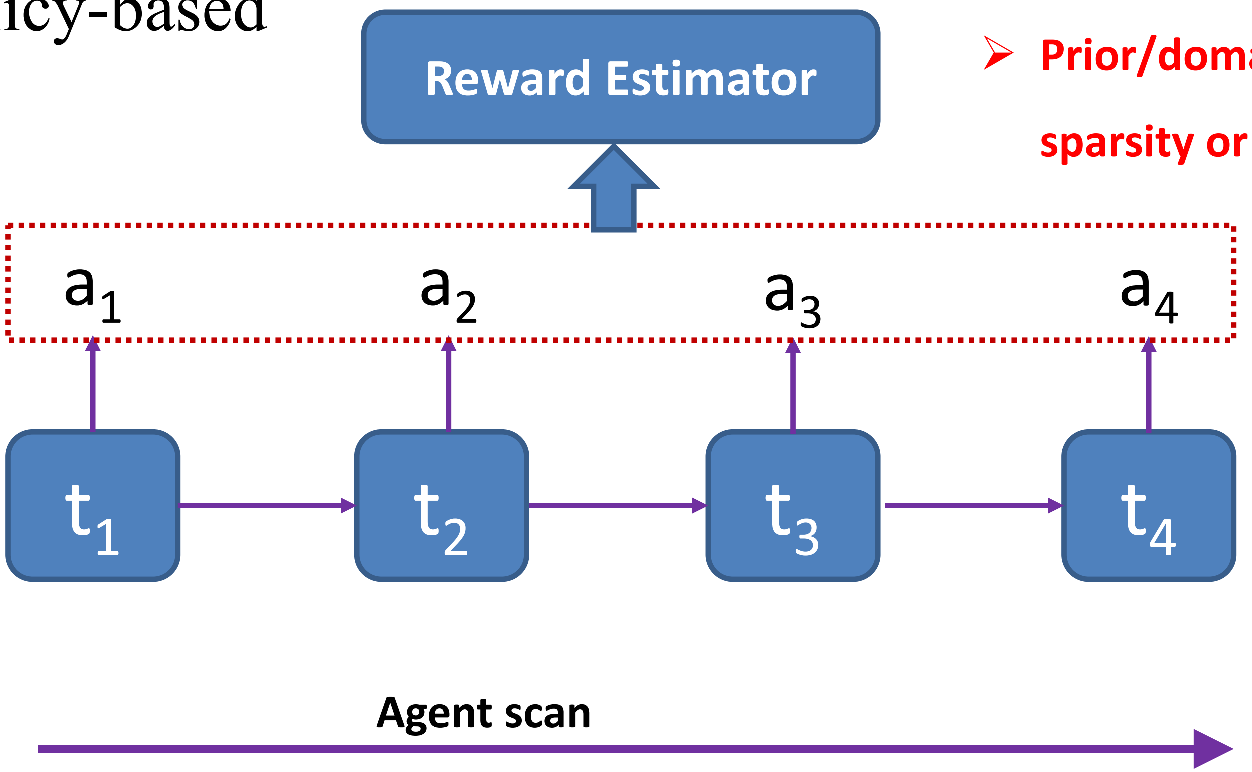
- Immediate rewards: **t** (time step), **a** (action), **R** (reward)
- Deep Q-learning



Applying RL in NLP

- Delayed rewards
- Policy-based

- Comparing with gold-standard: BLEU\ACC\F1
- By classifier: likelihood
- **Prior/domain expertise: sparsity or continuity**



Applications

- **Search and Reasoning**: model structure, text structure, reasoning path, etc.
- **Instance Selection**: unlabeled data selection, data denoising, noisy label correction
- **Strategy Optimization**: ranking, dialogue strategy, language game, negotiation, text compression, language generation

Search and Reasoning

1. **Find optimal model structure**
2. **Search for represent. structure**
3. **Search for reasoning path**

① Andreas, Jacob, et al. Learning to compose neural networks for question answering. NAACL 2016.
② Barret Zoph , Quoc V. Le. Neural Architecture Search with Reinforcement Learning. ICLR 2017.
③ Pham, Hieu, et al. Efficient Neural Architecture Search via Parameter Sharing. arXiv preprint arXiv:1802.03268 (2018).
④ Tianyang Zhang, Minlie Huang, Li Zhao.
Learning Structured Representation for Text Classification via Reinforcement Learning. AAAI 2018, New Orleans, Louisiana, USA.
⑤ Das et al. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. arXiv:1711.05851.

Composing Network Structure (Andreas et al., NAACL2016)

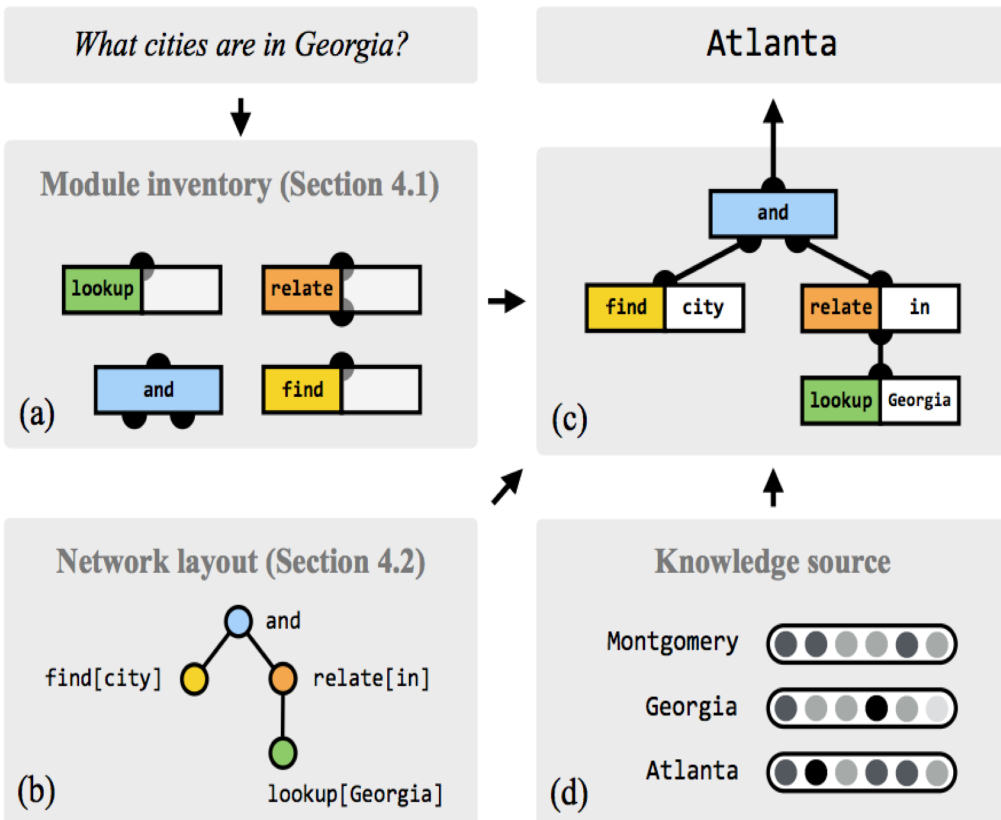


Figure 1: A learned syntactic analysis (a) is used to assemble a collection of neural modules (b) into a deep neural network (c), and applied to a world representation (d) to produce an answer.

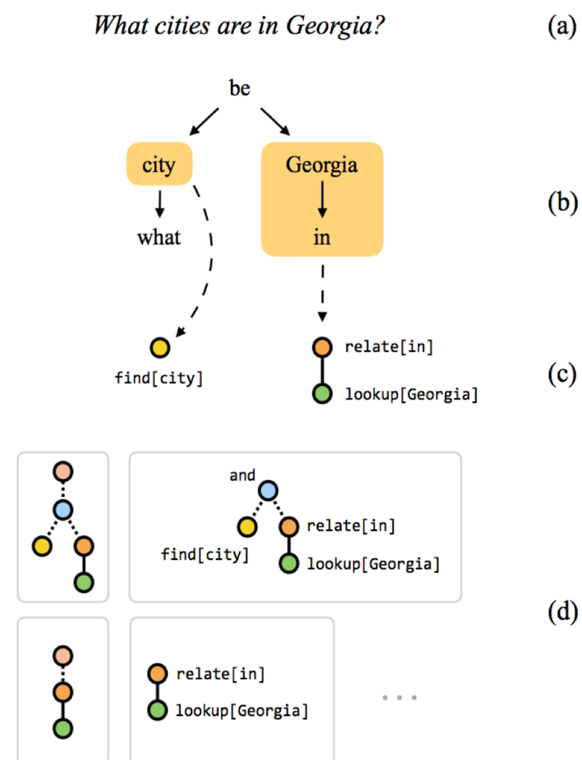
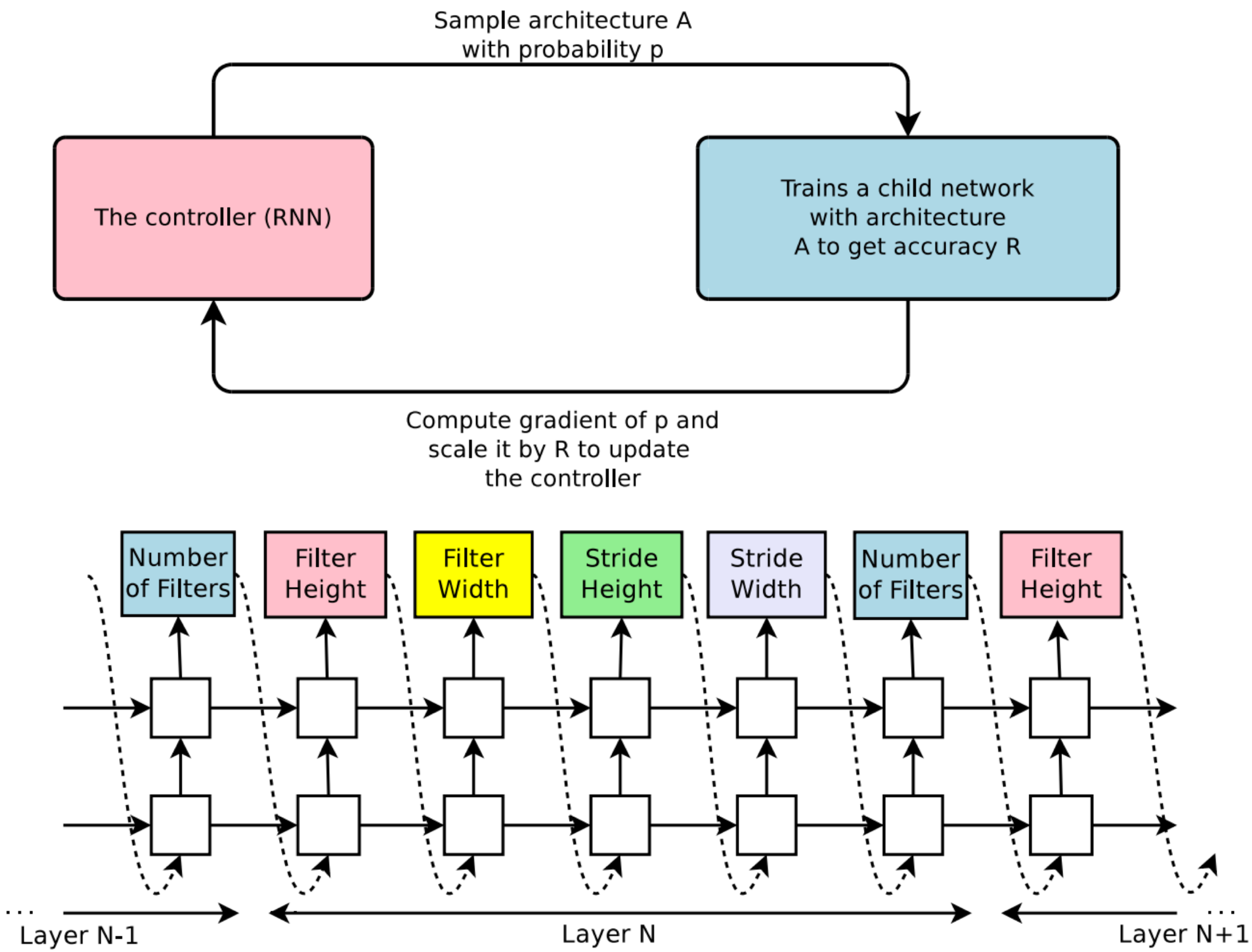


Figure 3: Generation of layout candidates. The input sentence (a) is represented as a dependency parse (b). Fragments of this dependency parse are then associated with appropriate modules (c), and these fragments are assembled into full layouts (d).

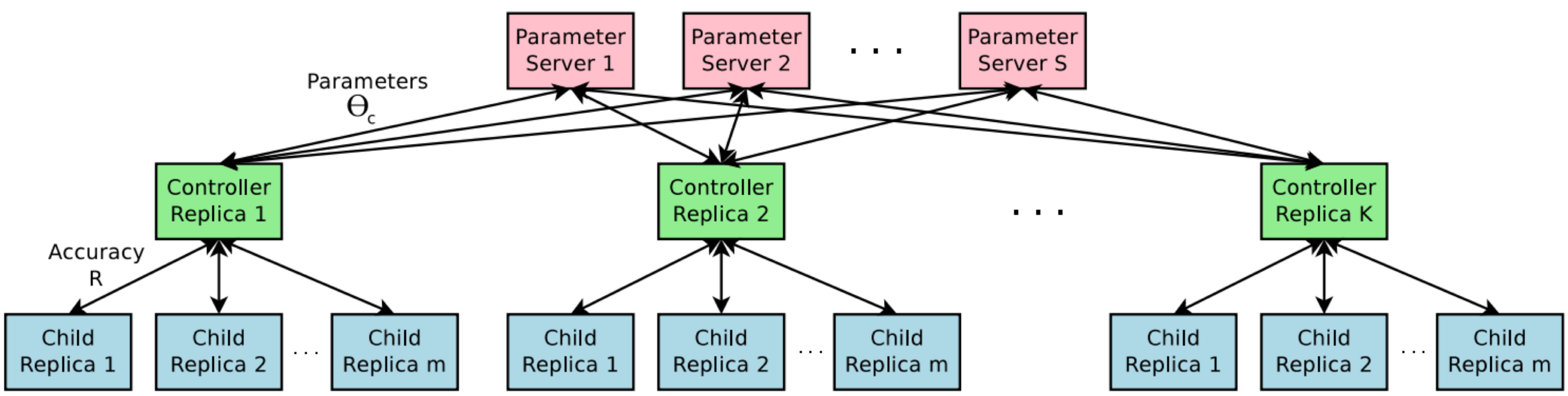
Neural Architecture Search (Zoph&Le, ICLR2017)



Neural Architecture Search (Zoph&Le, ICLR2017)

- **Reward R**: the accuracy of the configured model
- **REINFORCE** algorithm $J(\theta_c) = E_{P(a_{1:T};\theta_c)}[R]$

$$\nabla_{\theta_c} J(\theta_c) = \sum_{t=1}^{T} E_{P(a_{1:T};\theta_c)} \left[\nabla_{\theta_c} \log P(a_t | a_{(t-1):1}; \theta_c) R \right]$$



Discovering Text Structures (Zhang, Huang, Zhao; AAAI 2018)

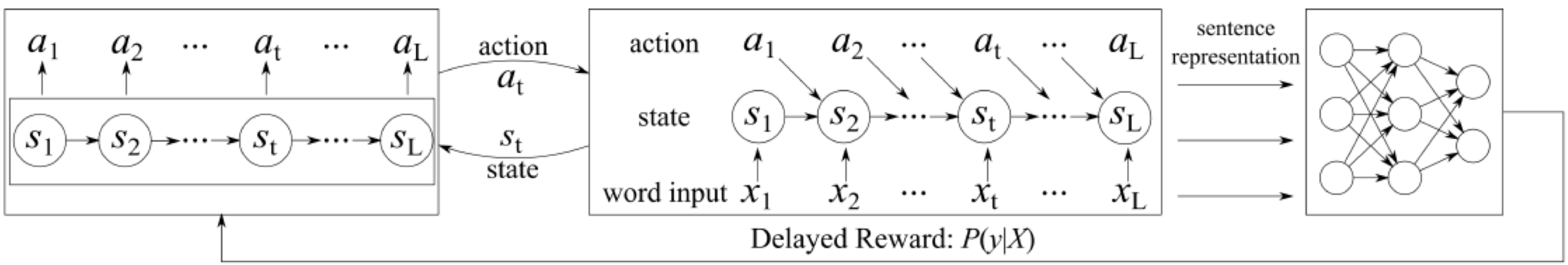
- **How can we identify task-relevant structures without explicit annotations on structure?**

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):

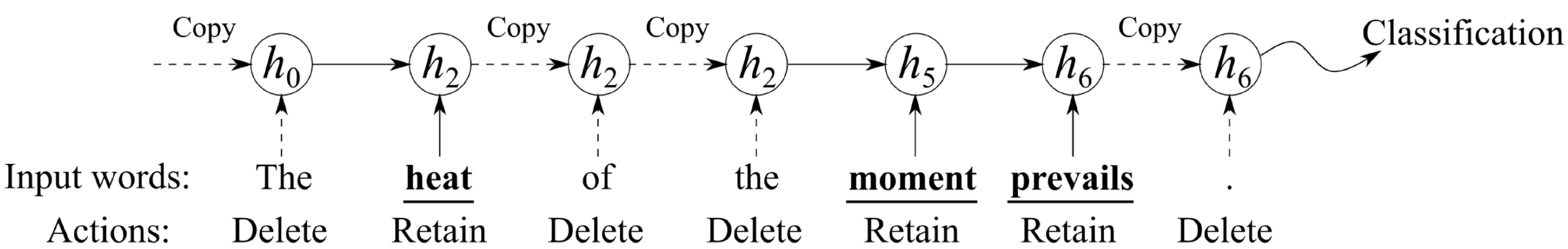
$$P(y|X) = softmax(\mathbf{W_s h_L} + \mathbf{b_s}),$$

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

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:

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

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

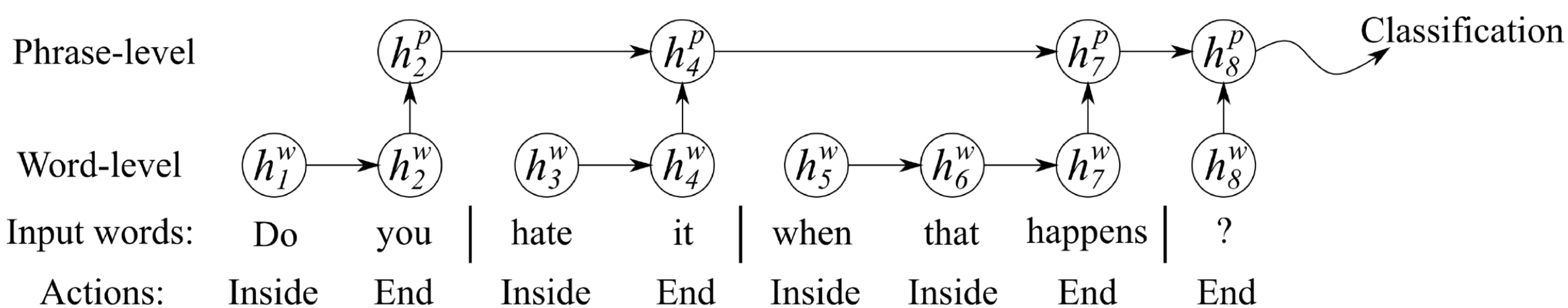
- Rewards:

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

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: $\{\textbf{Inside, End}\}$

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

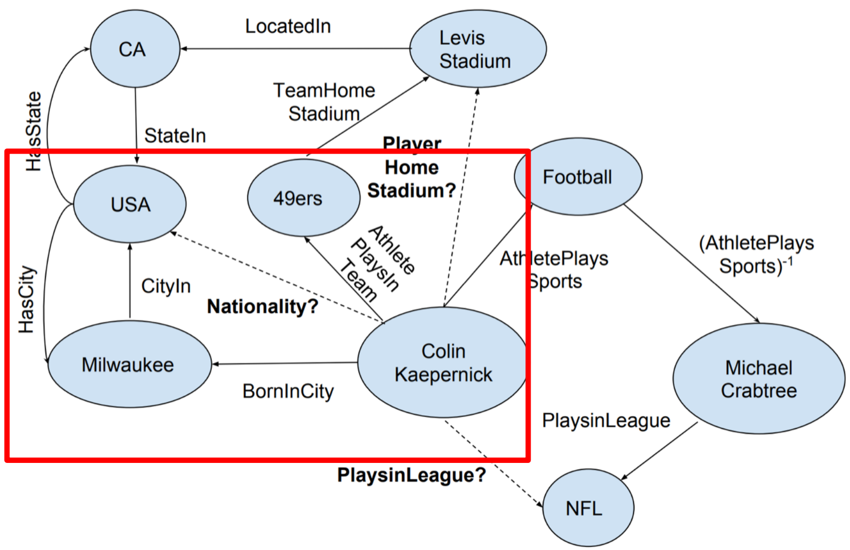
- Classification Results

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>

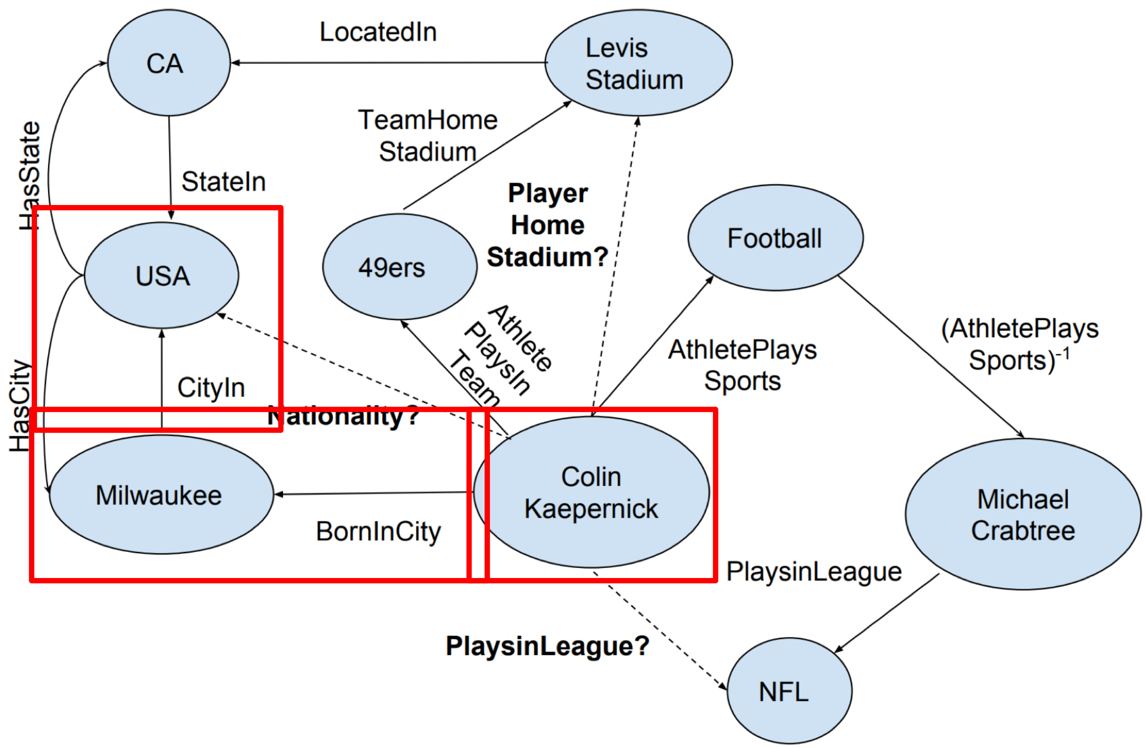
Search for Reasoning Path



- Input: query
  - (e1, r, ?)
  - (Colin Kaepernick, Nationality, ?)
- Output: answer entity
  - e2
  - USA

Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. Das et al., **arXiv:1711.05851.**

Search for Reasoning Path



$$\text{Colin Kaepernick} \xrightarrow{\text{BornInCity}} \text{Milwaukee} \xrightarrow{\text{CityIn}} \text{USA}$$

Model

- States: encodes the query, the answer, the current entity.

$$S = (\mathrm{e_t}, \mathrm{e_{1q}}, \mathrm{r_q}, \mathrm{e_{2q}})$$

- Observations: the complete state of the environment is not observable, as the answer is not observed

$$\mathcal{O}(s = (\mathrm{e_t}, \mathrm{e_{1q}}, \mathrm{r_q}, \mathrm{e_{2q}})) = (\mathrm{e_t}, \mathrm{e_{1q}}, \mathrm{r_q})$$

Model

- Actions: the set of possible actions consists of all outgoing edges of the current vertex

$$\mathcal{A}_S = \{(\mathrm{e}_t, r, v) \in E : S = (\mathrm{e}_t, \mathrm{e}_{1q}, \mathrm{r}_q, \mathrm{e}_{2q}), r \in \mathcal{R}, v \in V\} \cup \{(s, \varnothing, s)\}$$

- Rewards: only have a terminal reward of +1 if the current location is the correct answer at the end and 0 otherwise

$$R(S_T) = \mathbb{I}\{\mathrm{e}_t = \mathrm{e}_{2q}\}$$

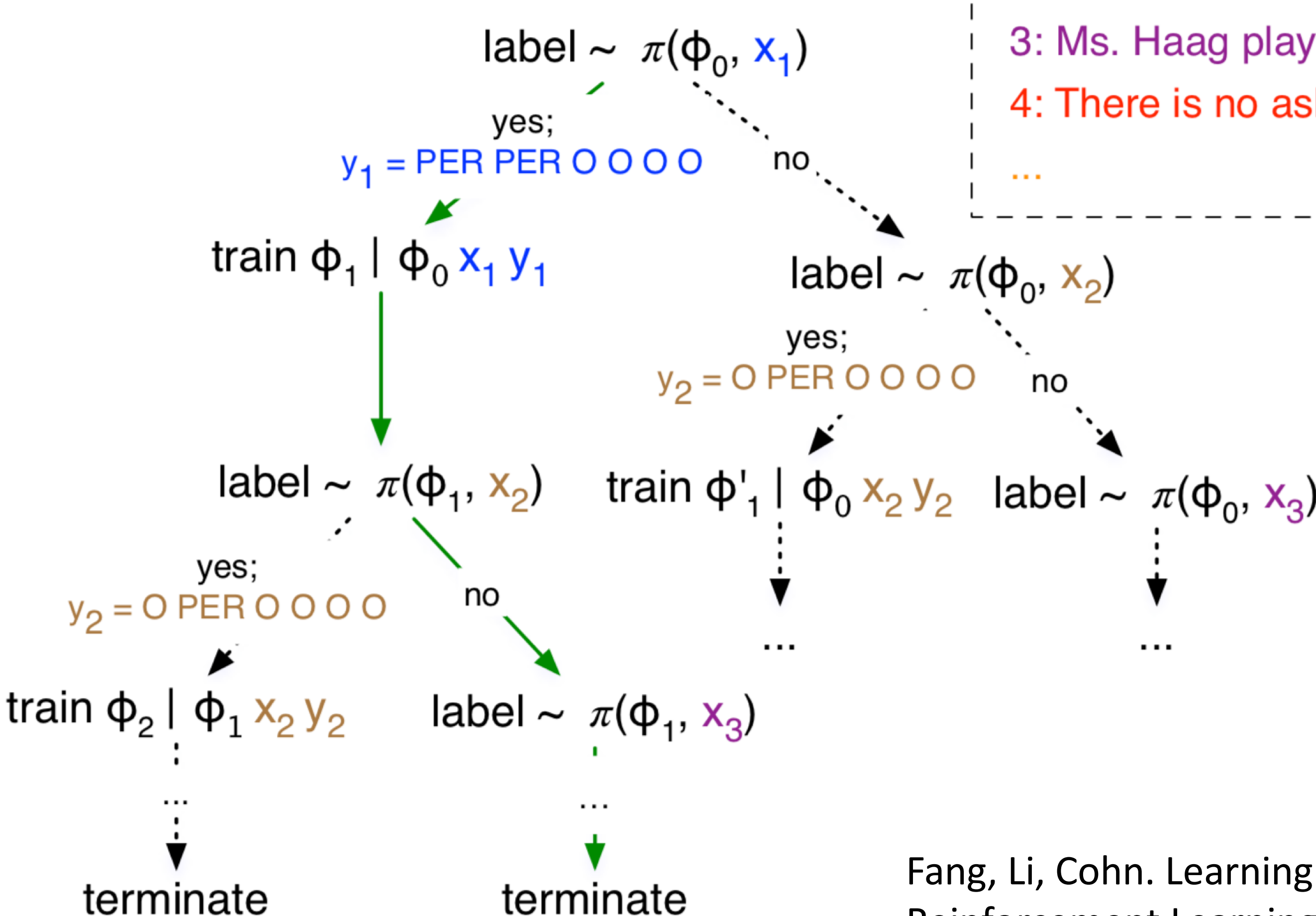
Instance Selection

1. Selecting unlabeled data in SSL or co-training
2. Selecting mini-batch order in SGD
3. Data denoising (removing noisy instances)
4. Label correction in noisy labeling

① Meng Fang, Yuan Li, Trevor Cohn. Learning how to Active Learn: A Deep Reinforcement Learning Approach. EMNLP 2017.
② Yang Fan, Fei Tian, Tao Qin, Jiang Bian, Tie-Yan Liu. Learning What Data to Learn.
③ Jiawei Wu, Lei Li, Willian Yang Wang. Reinforced Co-Training. NAACL 2018.
④ Jun Feng, Minlie Huang, Li Zhao, Yang Yang, Xiaoyan Zhu. Reinforcement Learning for Relation Classification from Noisy Data. AAAI 2018.
⑤ Zeng et al., Large Scaled Relation Extraction with Reinforcement Learning. AAAI 2018.
⑥ Ryuichi Takanobu, Minlie Huang, Zhongzhou Zhao, Fenglin Li, Haiqing Chen, Xiaoyan Zhu, Liqiang Nie. A Weakly Supervised Method for Topic Segmentation and Labeling in Goal-oriented Dialogues via Reinforcement Learning. IJCAI-ECAI 2018.

Unlabeled Data Selection (Fang et al., EMNLP2017)

1: Pierre Vinken will join the board
2: Mr. Vinken is chairman of Elsevier
3: Ms. Haag plays Elianti
4: There is no asbestos in our products
...

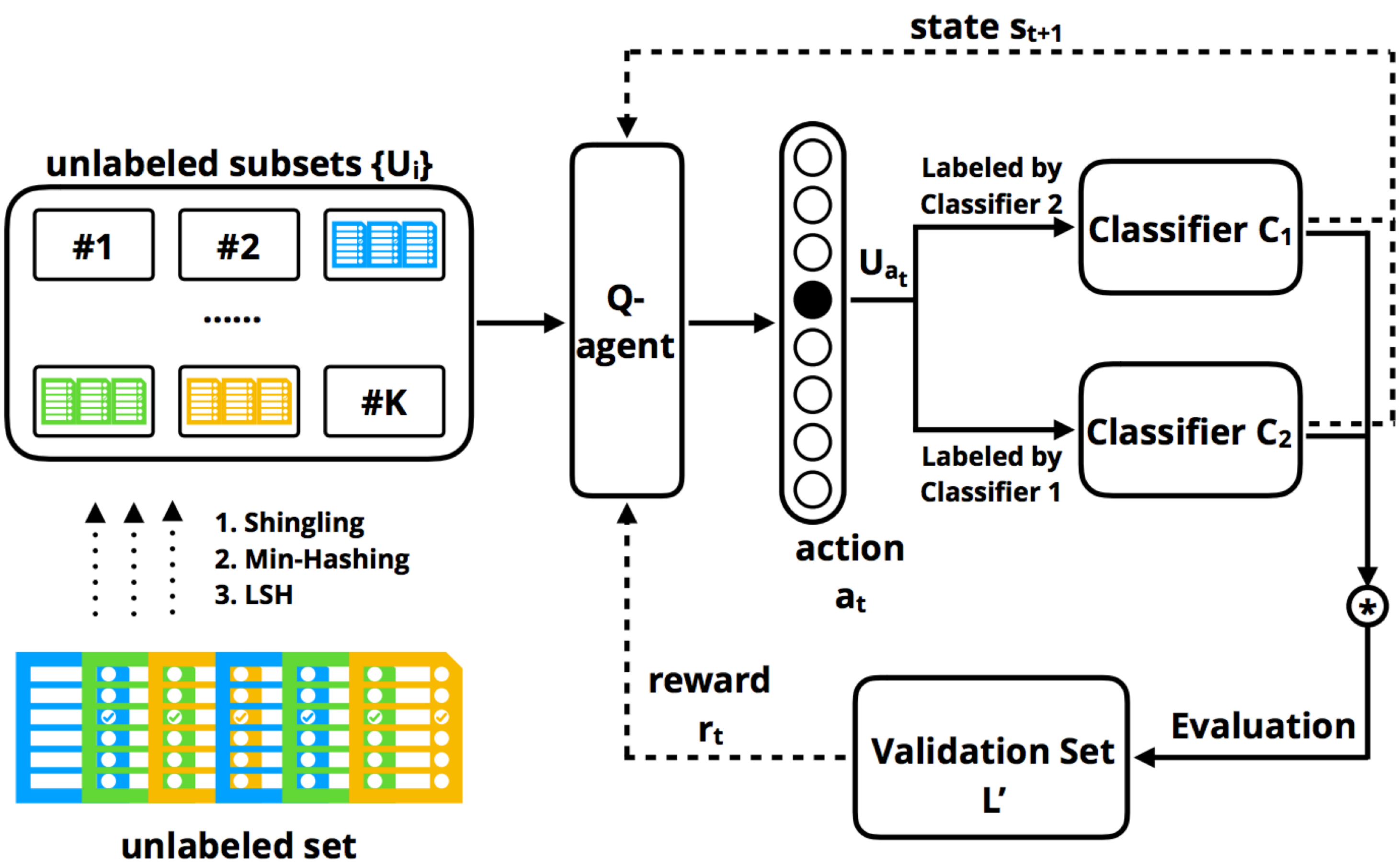


Fang, Li, Cohn. Learning how to Active Learn: A Deep Reinforcement Learning Approach. EMNLP 2017.

Unlabeled Data Selection (Fang et al., EMNLP2017)

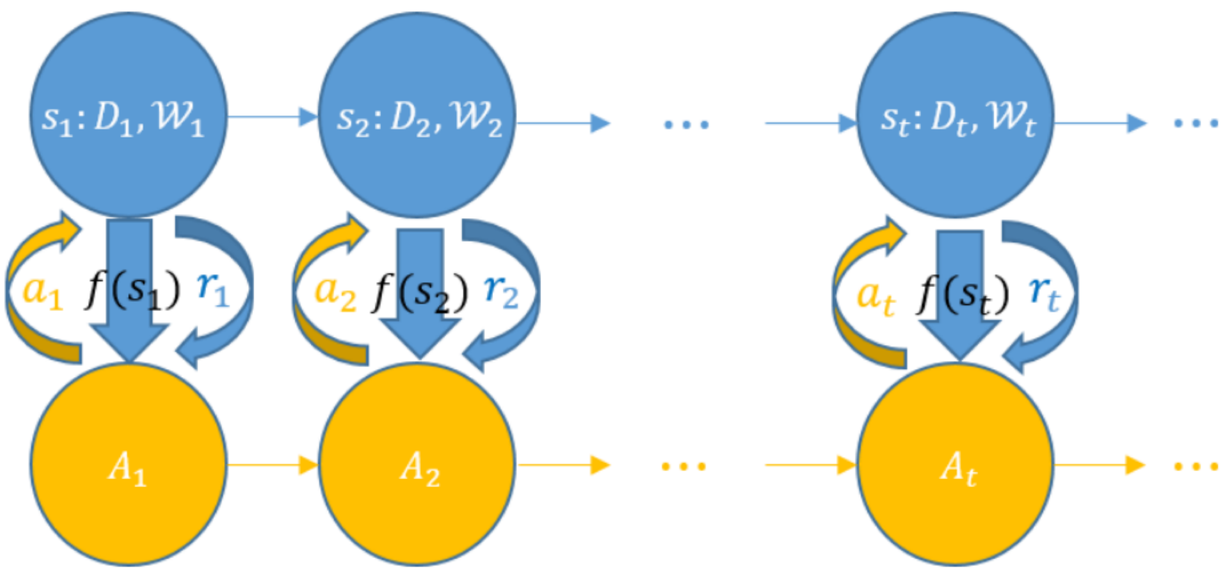
- **State**: the candidate instance being considered for annotation and the labelled dataset constructed in steps 1,2,3,…, i
- **Action**: 0/1, whether to use x_i for training
- **Reward**: the accuracy margin in two model updates.
- **Optimization**: deep Q-learning

Reinforced CoTraining (Wu et al., NAACL2018)



Mini-Batch Selection in SGD (Fan et al., 2017)

- In SGD, the order of data batch in model update is important
- State: data feature, base model feature, combination of the two



Instance Denoising
(Feng et al., AAAI 2018)

- Relation Classification (or extraction)

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

Relation: *BornIn*

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

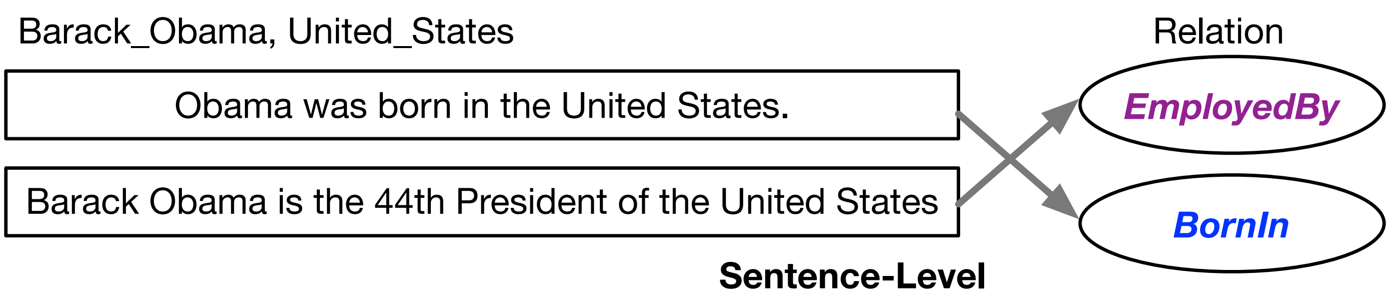
[**Barack Obama**]$_{e1}$ is ~~**the 44th President of**~~ the [**United States**]$_{e2}$.

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

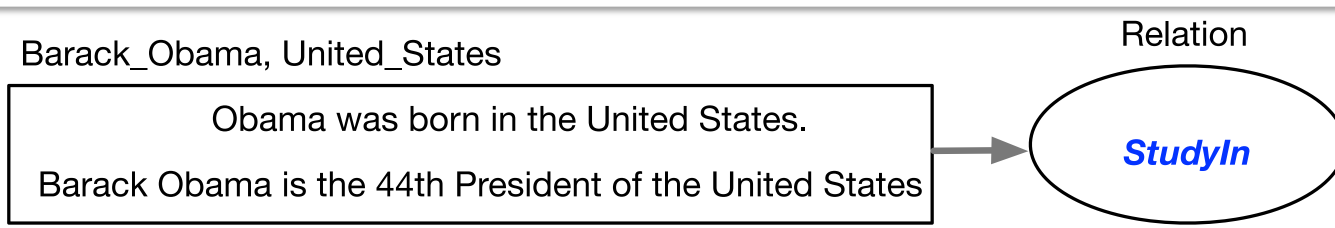
Relation: ***BornIn***

Instance Denoising
(Feng et al., AAAI 2018)

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

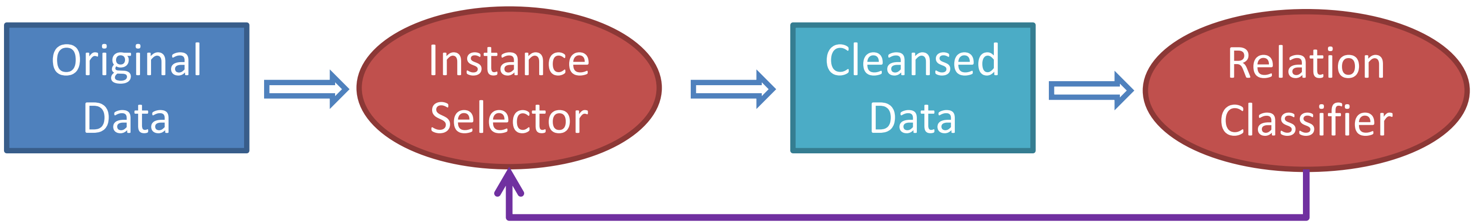


How can we remove noisy data to improve relation extraction without explicit annotations?



Model Structure

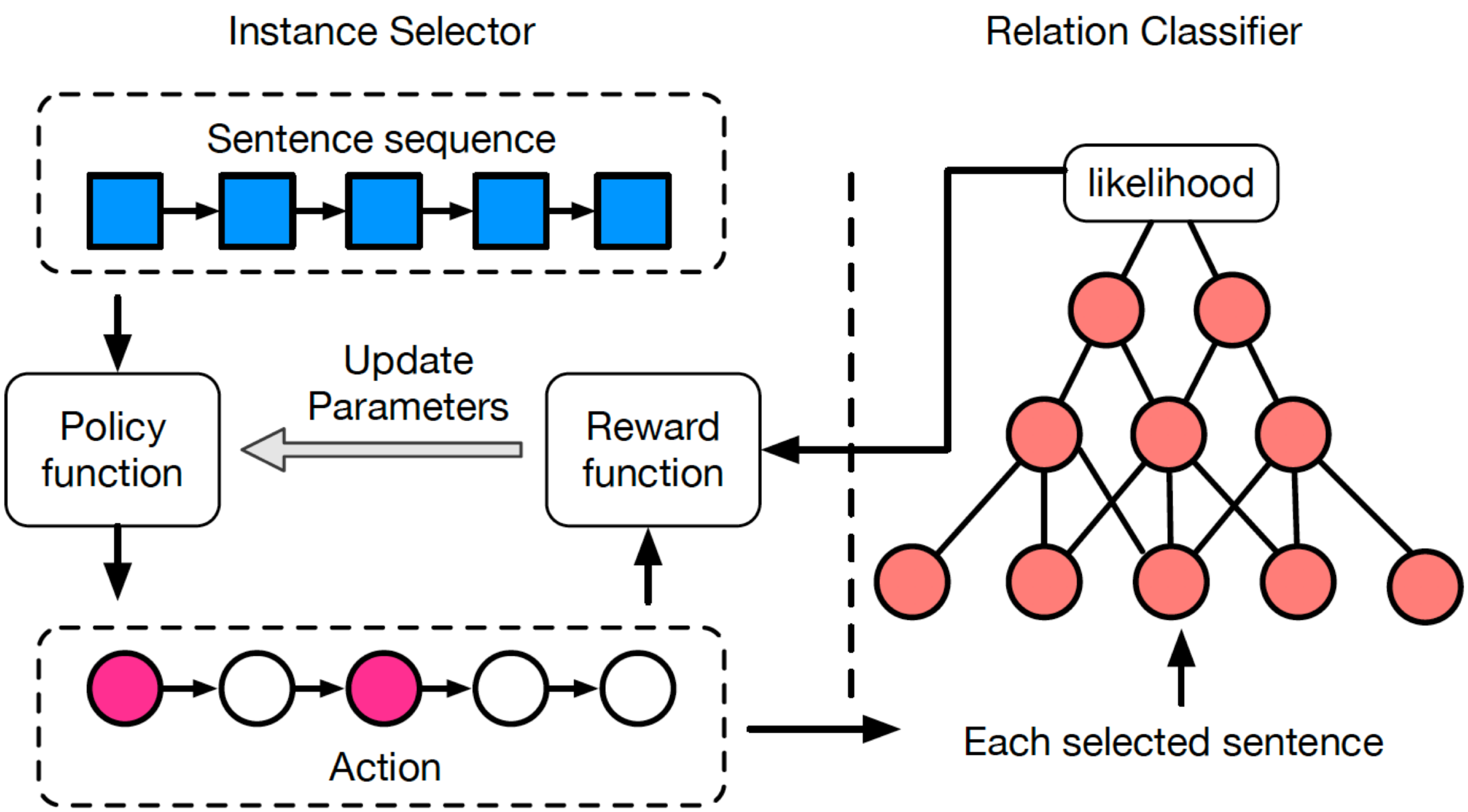
- The model consists of an **instance selector** and **a relation classifier**



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

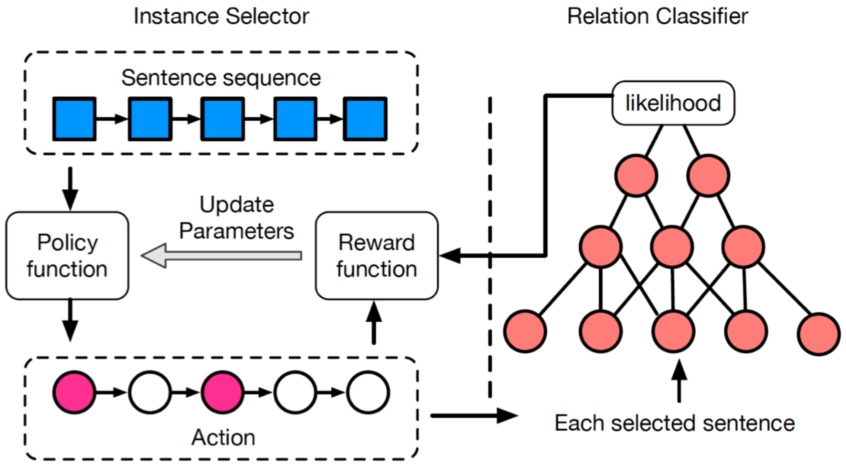
    → Reinforcement Learning
  - How to train the two modules jointly

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



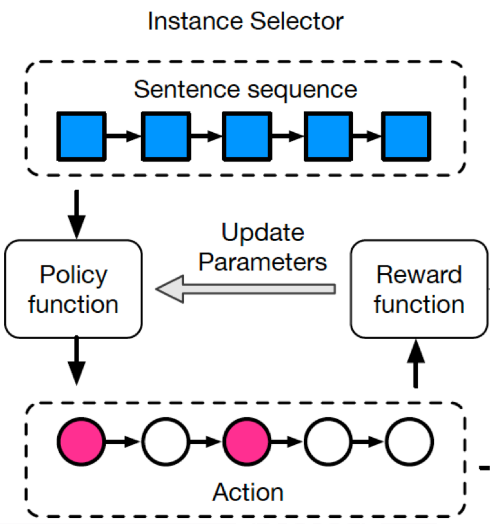
Instance Selector

- Instance selection as a reinforcement learning problem
  - **State**: ***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

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

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

- 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

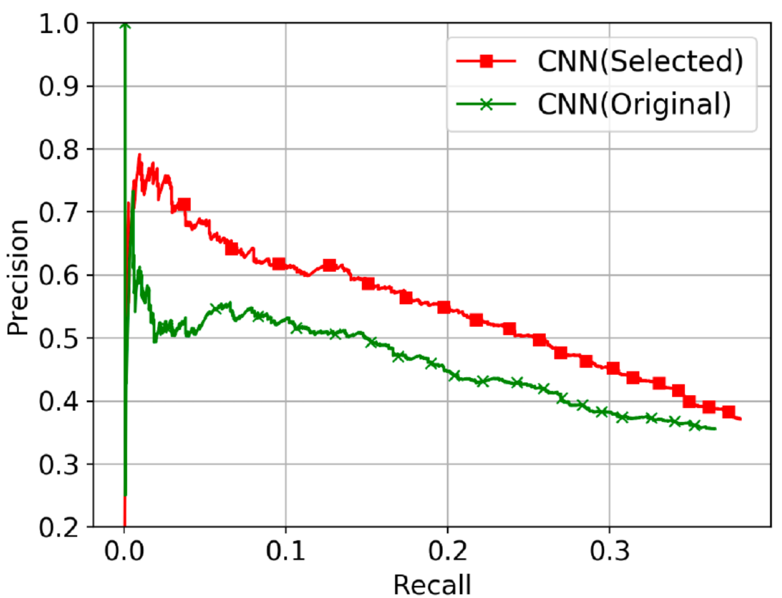
Experiment

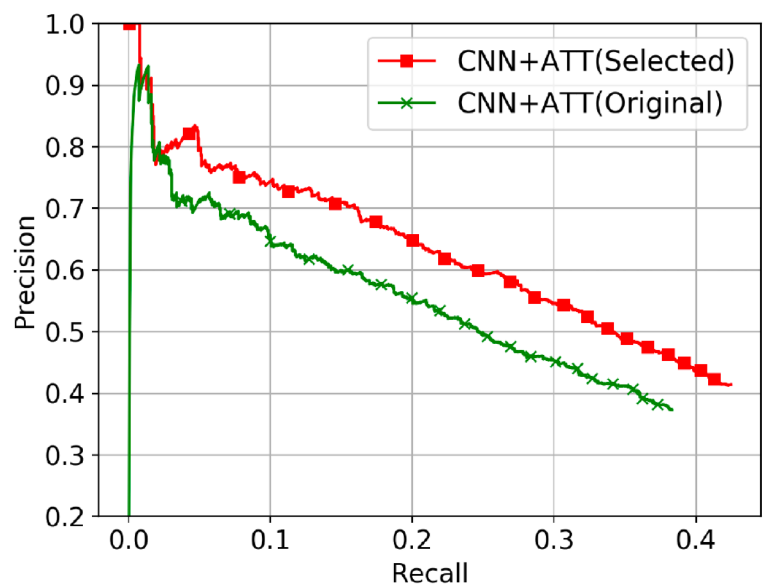
- Dataset
  - NYT and developed by (Riedel, Yao, and McCallum 2010)
- Baselines
  - CNN: is a sentence-level classification model. It does not consider the noisy labeling problem.
  - CNN+Max: assumes that there is one sentence describing the relation in a bag and chooses the most correct sentence in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





Noisy Label Correction (Takanobu et al., IJCAI 2018)

Product-info	A:	The release date of ⟨ MODEL ⟩???
	B:	⟨ MODEL ⟩ will be available for pre-order on 19 April and launch on 26.
	A:	How long can the battery last?
	B:	It's equipped with a 4,000 mAh battery up to 8 hours of HD video playing or 10 hours of web browsing.
Promotion	A:	Can I use a coupon?
	B:	When entering your payment on the checkout page, click *Redeem a coupon* below your payment method.
	B:	You can check here for more details: ⟨ URL ⟩.
Payment	A:	OK. Support payment by installments?
	B:	Sure. We provide an interest-free installment option for up to 6 months.

Table 1: An example of customer service dialogues, translated from Chinese. Utterances in the same color are of the same topic.

Noisy Label Correction (Takanobu et al., IJCAI 2018)

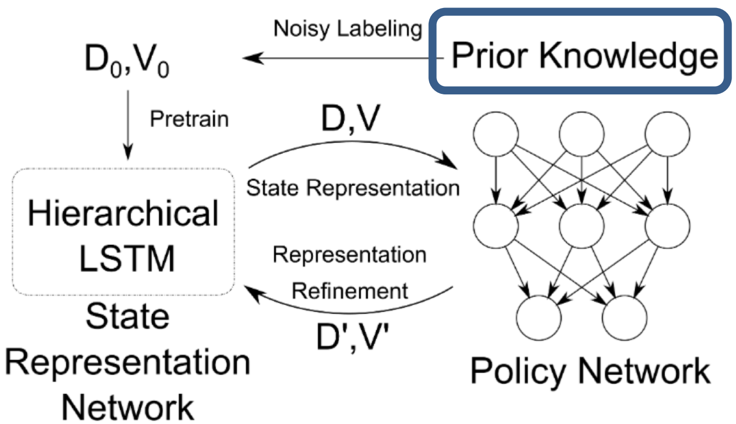
Datasets	SmartPhone	Clothing
# Topic category	7	10
# Training session	12,315	10,000
# Training utterance	430,462	338,534
# Gold-standard session	300	315
# Gold-standard utterance	10,888	10,962

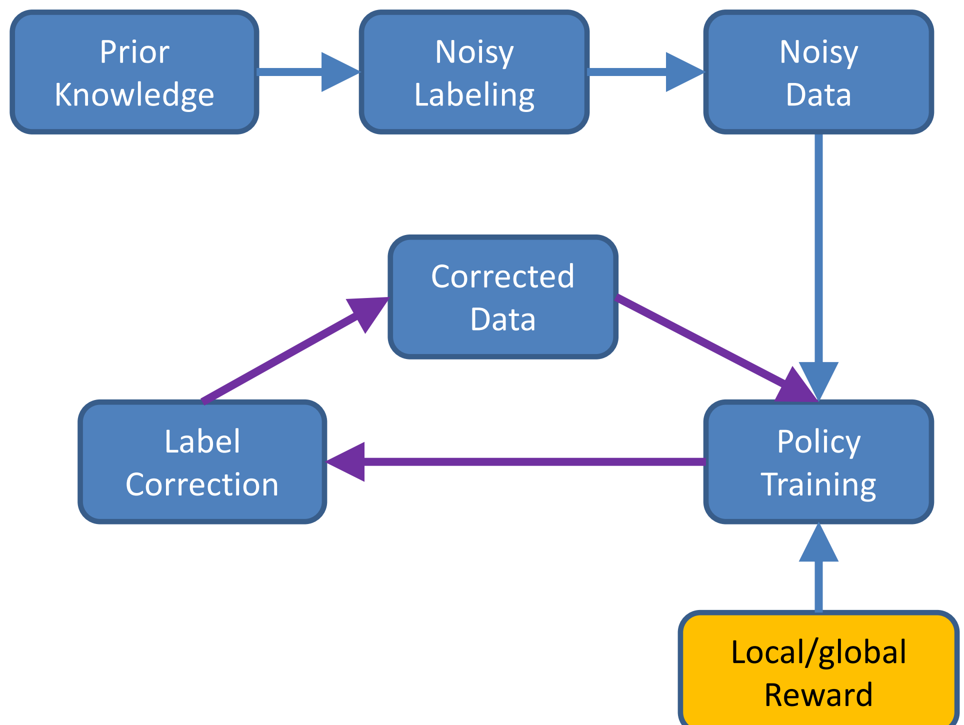
Table 2: Statistics of the corpus.

How can we do topic labeling on these large-scale dialogues without much annotation efforts?

Central Idea

- Noisy labeled data → learn policies with reward → refine data → learn better policies → refine more data





Learning from weakly annotated data

Model Structure

- State Representation Network
- Policy Network

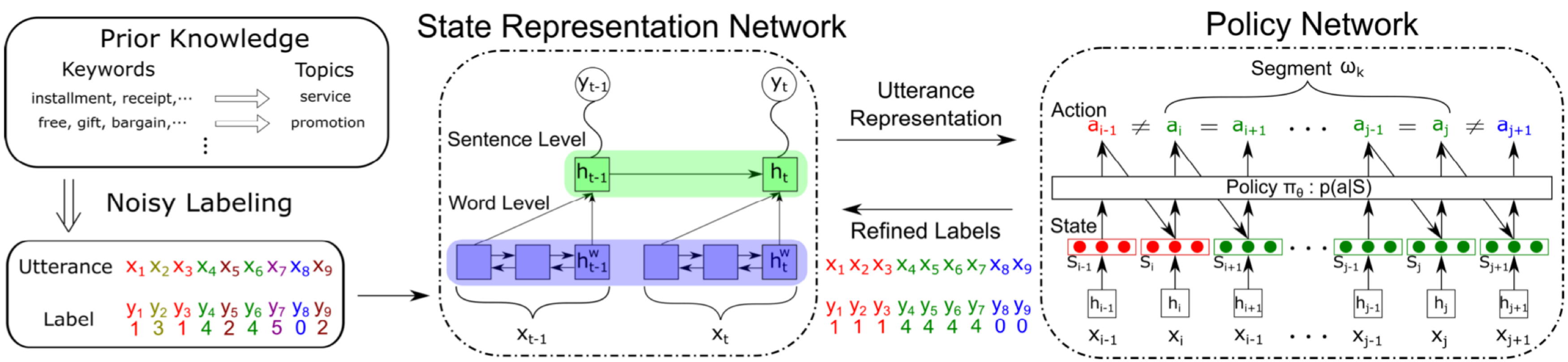


Figure 1: Illustration of the model. SRN adopts a hierarchical LSTM to represent utterances and provides state representations to PN. Data labels are refined to retrain SRN and PN to learn better state representations and policies. The label y and the action a are in the same space.

Model Structure

- **Local topic continuity**: the same topic will continue in a few dialogue turns

$$r_{int} = \frac{1}{L-1} sign(a_{t-1} = a_t) \cos(\mathbf{h}_{t-1}, \mathbf{h}_t)$$

- **Global topic structure**: high content similarity within segments but low between segments

$$r_{delayed} = \frac{1}{N} \sum_{\omega \in X} \frac{1}{|\omega|} \sum_{X_t \in \omega} \cos(\mathbf{h}_t, \boldsymbol{\omega}) - \frac{1}{N-1} \sum_{(\omega_{k-1}, \omega_k) \in X} \cos(\boldsymbol{\omega}_{k-1}, \boldsymbol{\omega}_k)$$

Experiment

(a) Topic Segmentation (MAE and WD)

Model	SmartPhone		Clothing	
	MAE	WD	MAE	WD
TextTiling(TT)	13.09	.802	16.32	.948
TT+Embedding	3.59	.564	3.17	.567
STM	4.37	.505	8.85	.669
NL+HLSTM	8.25	.632	16.26	.925
Our method	**2.69**	**.415**	**2.74**	**.446**

(b) Topic Labeling (Accuracy)

Model	SmartPhone	Clothing
Keyword Matching	39.8	31.8
NL	51.4	39.0
NL+LSTM	49.6	35.5
NL+HLSTM	52.6	40.1
Our method	**62.2**	**48.0**

(a)

Model	# Keywords per topic		
	3	6	9
NL	45.0	51.4	48.0
NL+HLSTM	46.6	52.6	48.8
Our method	**55.3**	**62.2**	**58.2**

(b)

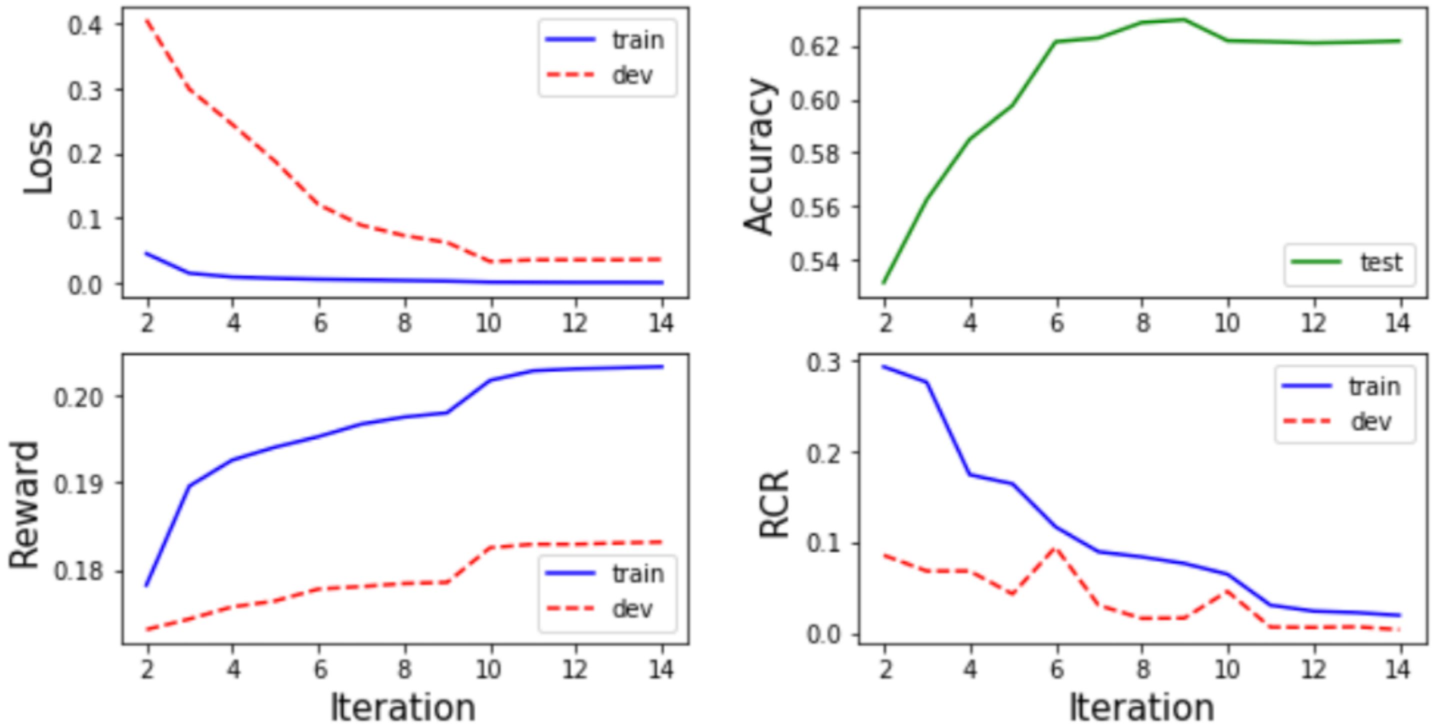
SubSets	KM	1-NN
Utterances	3,503	7,385
NL	78.7	38.4
NL+HLSTM	78.6	40.2
Our method	**79.0**	**54.2**

(c)

Model Setting	Segmentation		Labeling
	MAE	WD	Acc
RL + r_{int}	3.04	.449	59.5
RL + $r_{delayed}$	3.89	.490	60.4
RL + r_{int} + $r_{delayed}$	**2.69**	**.415**	**62.2**

Experiment

- Training converges well (loss, reward, accuracy, relative data change)

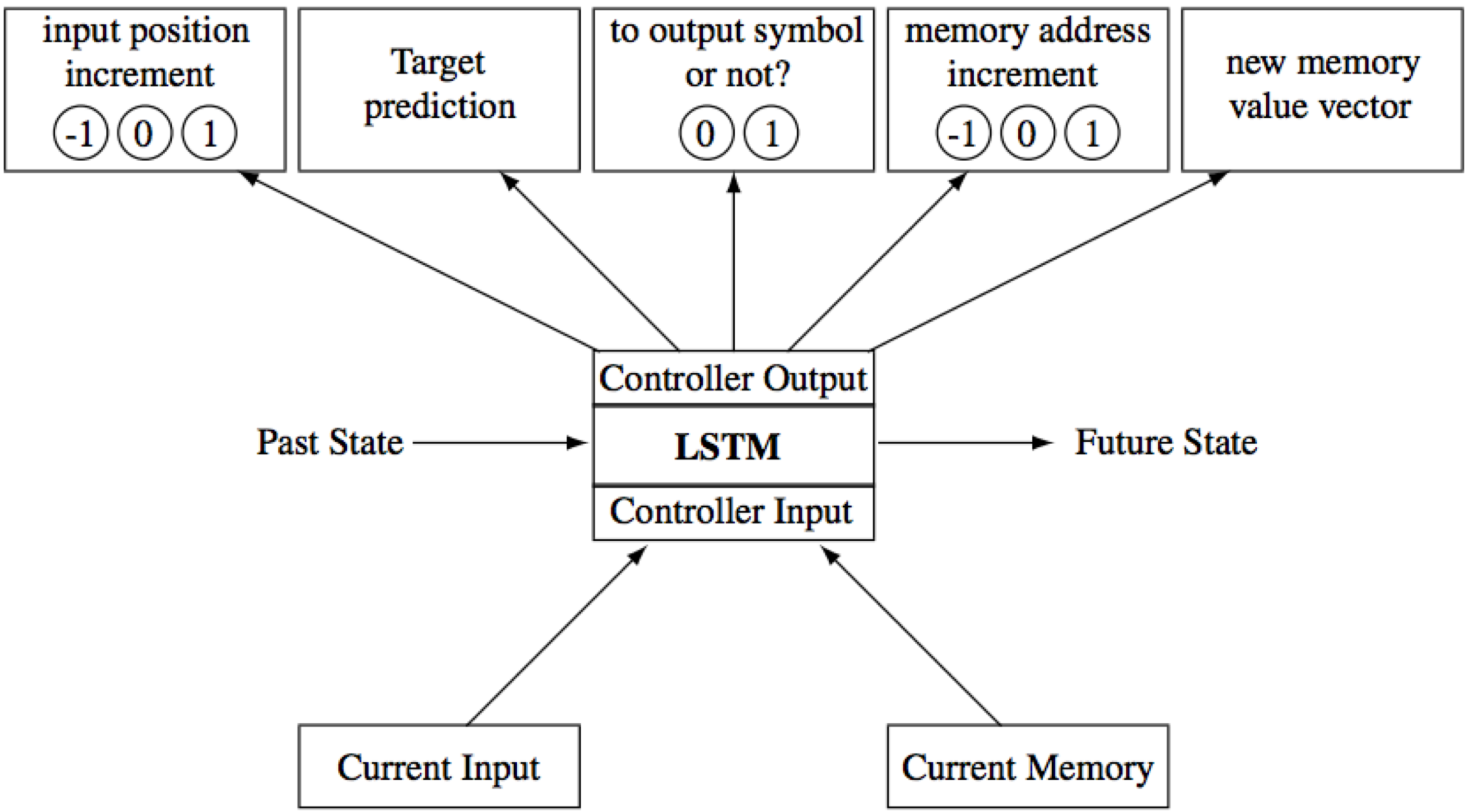


Strategy Optimization

1. **Language Generation**
2. **Dialogue Strategy**
3. **Ranking Optimization in Search**

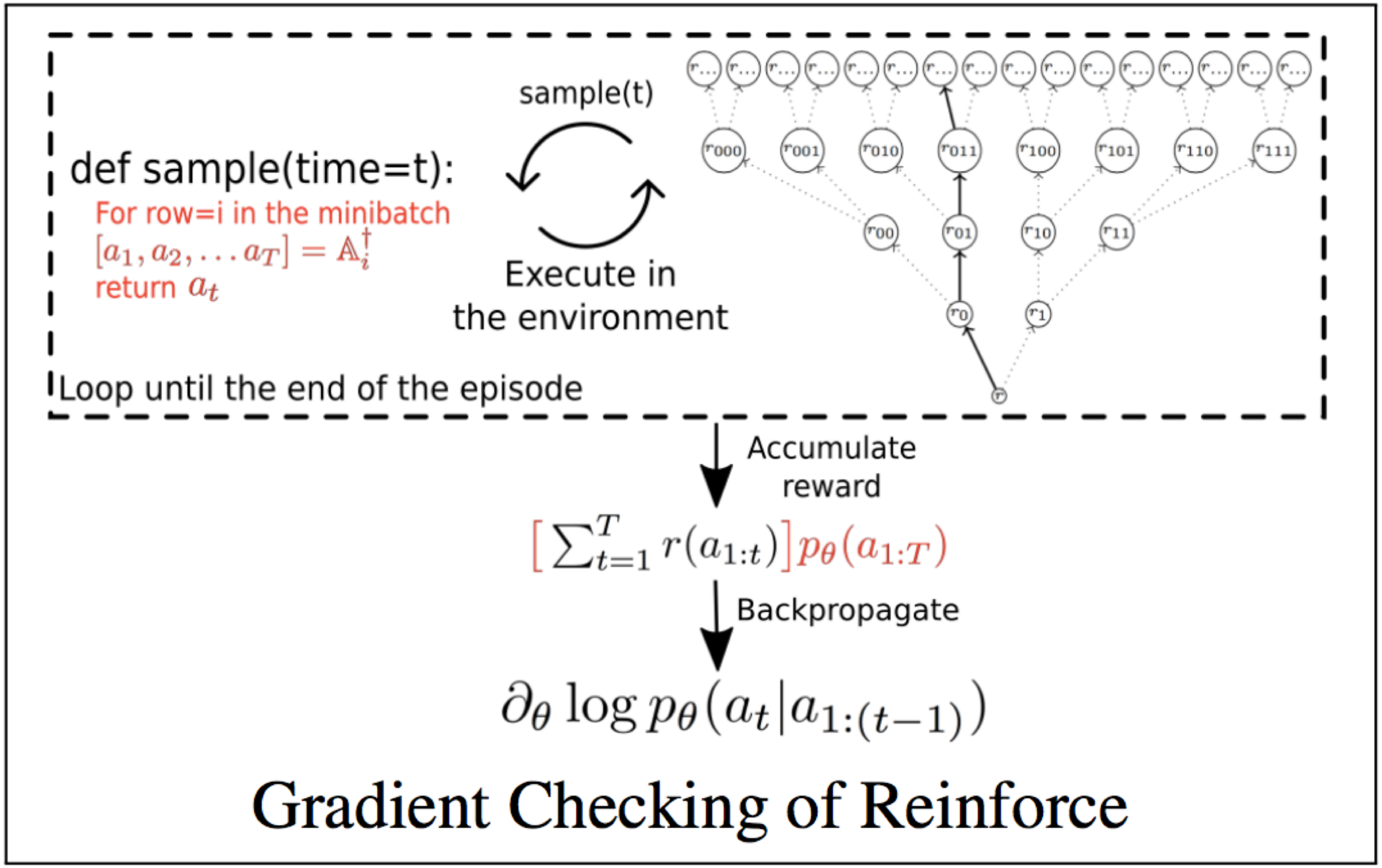
① Zaremba, Wojciech, and Ilya Sutskever. Reinforcement learning neural turing machines-revised. arXiv preprint arXiv:1505.00521 (2015).
② Xingxing Zhang and Mirella Lapata. Sentence Simplification with Deep Reinforcement Learning. EMNLP 2017.
③ Li et al. Deep Reinforcement Learning for Dialogue Generation. EMNLP 2016.
④ Peng et al. Composite Task-Completion Dialogue Policy Learning via Hierarchical Deep Reinforcement Learning. EMNLP 2017.
⑤ Zhang et al. Exploring Implicit Feedback for Open Domain Conversation Generation. AAAI 2018.
⑥ Fent et al. Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning. WWW 2018.

Reinforce Learning NTM (Zaremba et al. 2015)



Reinforce Learning NTM

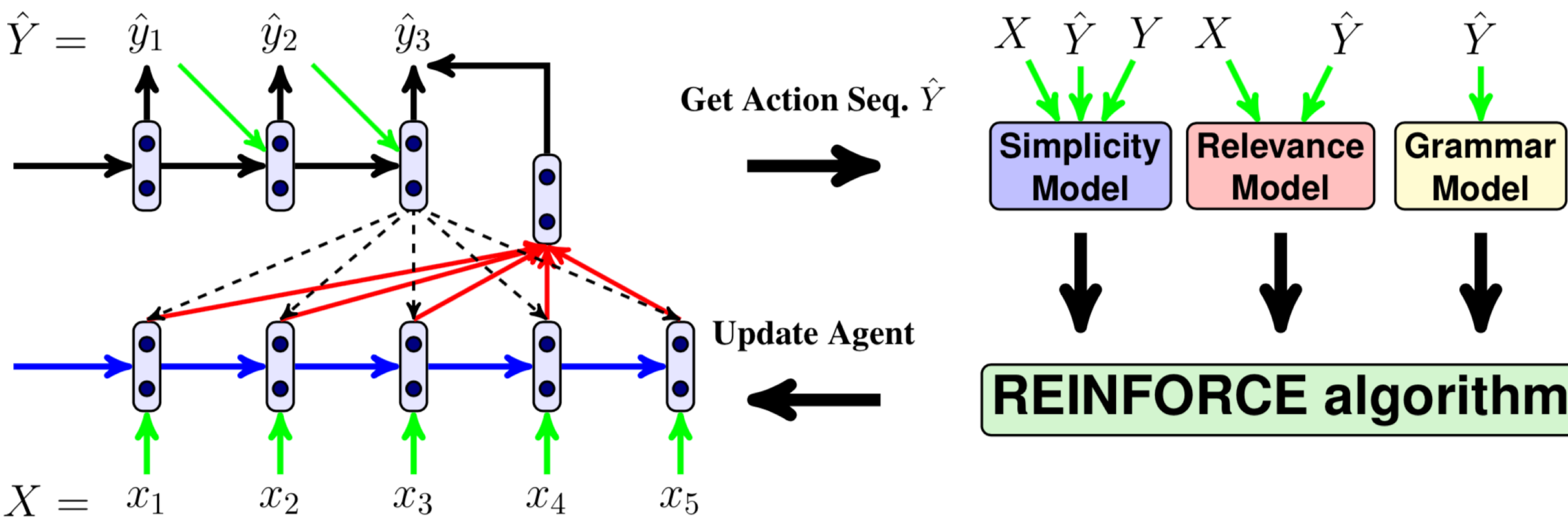
(Zaremba et al. 2015)



Gradient Checking of Reinforce

$$J(\theta) = \sum_{[a_1, a_2, \ldots, a_T] \in \mathbb{A}^\dagger} p_\theta(a_1, a_2, \ldots, a_T) R(a_1, a_2, \ldots, a_T) = \sum_{a_{1:T} \in \mathbb{A}^\dagger} p_\theta(a_{1:T}) R(a_{1:T})$$

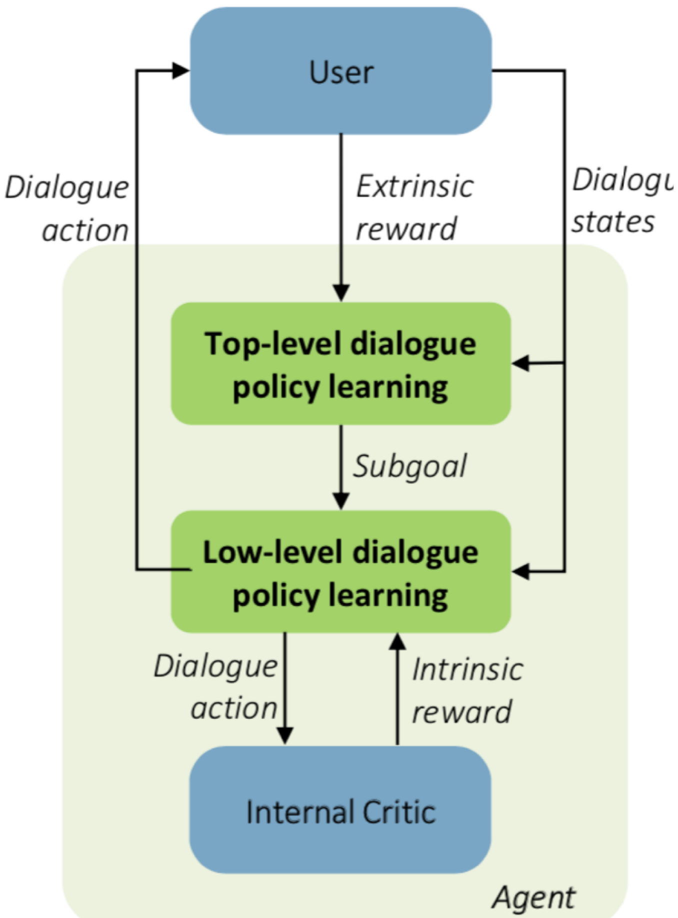
Language Generation (Zhang&Lapata, EMNLP2107)



$$\mathcal{L}(\theta) = -\mathbb{E}_{(\hat{y}_1,\ldots,\hat{y}_{|\hat{Y}|})\sim P_{RL}(\cdot|X)}[r(\hat{y}_1,\ldots,\hat{y}_{|\hat{Y}|})]$$

$$\nabla\mathcal{L}(\theta) \approx$$
$$\sum_{t=1}^{|\hat{Y}|} \nabla \log P_{RL}(\hat{y}_t|\hat{y}_{1:t-1}, X)[r(\hat{y}_{1:|\hat{Y}|}) - b_t]$$

High-level Dialogue Strategy (Peng et al. EMNLP2017)



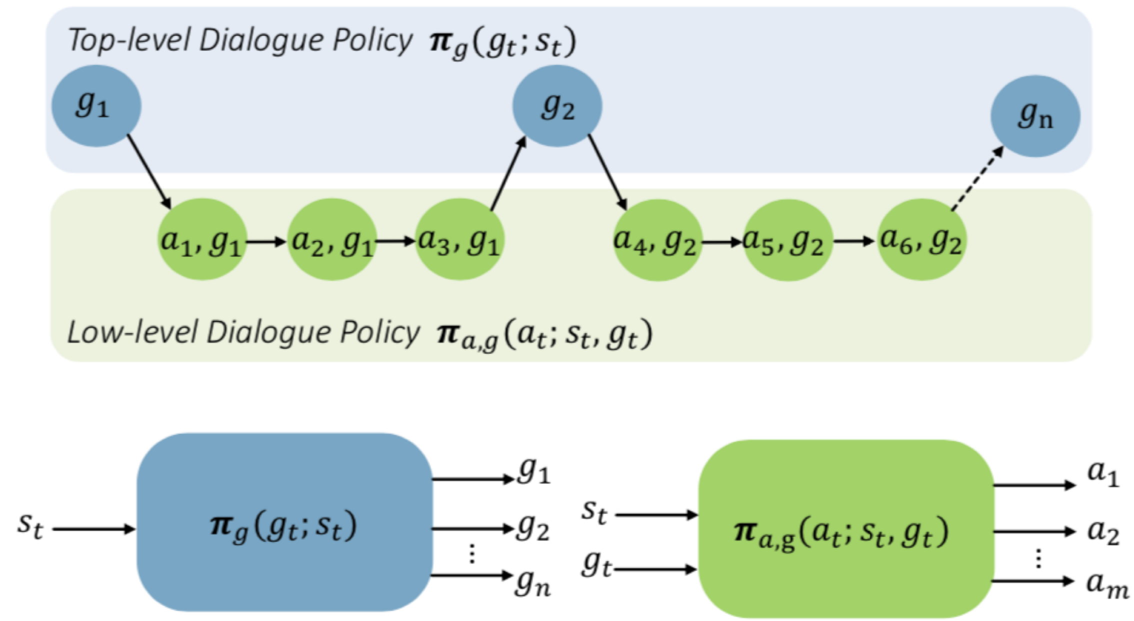


Figure 2: Illustration of a two-level hierarchical dialogue policy learner.

Many Other Applications

- **Negotiation** ("Deal or No Deal? End-to-End Learning for Negotiation Dialogues")
- **Language game** ("Language Understanding for Text-based Games using Deep Reinforcement Learning")
- **Information extraction** ("Improving Information Extraction by Acquiring External Evidence with Reinforcement Learning")

Reinforcement Learning in Search

- Usually **multi-turn interactions**
  - Could be natural **sequential decision** problems
  - For instance, search result diversification
- **No direct supervision** on which you should do at each step
- Only **implicit feedbacks** from user behavior data
  - Not necessarily as **direct supervision**
  - Good as **reward signals** for RL
- Totally **dynamic** systems (online training with real-time interactions)

Reinforcement Learning in Search

- **Query reformulation** (Nogueira & Cho, 2017; Buck et al., ICLR 2018)
- **Search results diversification** (Xia et al., SIGIR 2017)
- **Layout optimization** (Oosterhuis & Rijke, SIGIR 2018)
- **Ranking optimization** (Feng et al., WWW 2018)
- Tutorial by Jun Xu & Liang Pang:
  - **"Deep and reinforcement learning for information retrieval"**

Ranking Opti. In Search (Feng et al., WWW2018)

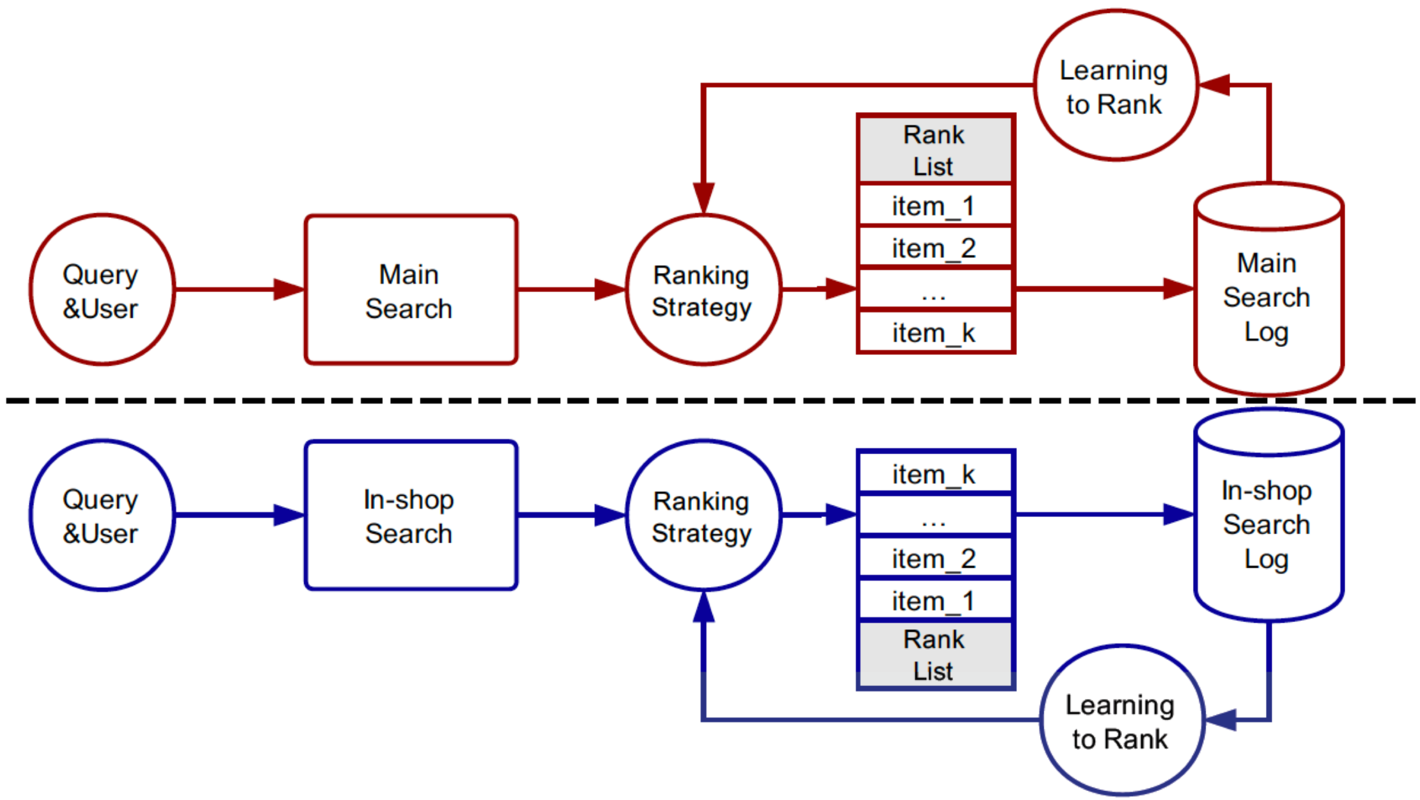
- **Multi-scenario Ranking:** most large-scale online platforms or mobile Apps have multiple scenarios

Main-search

In-shop Search

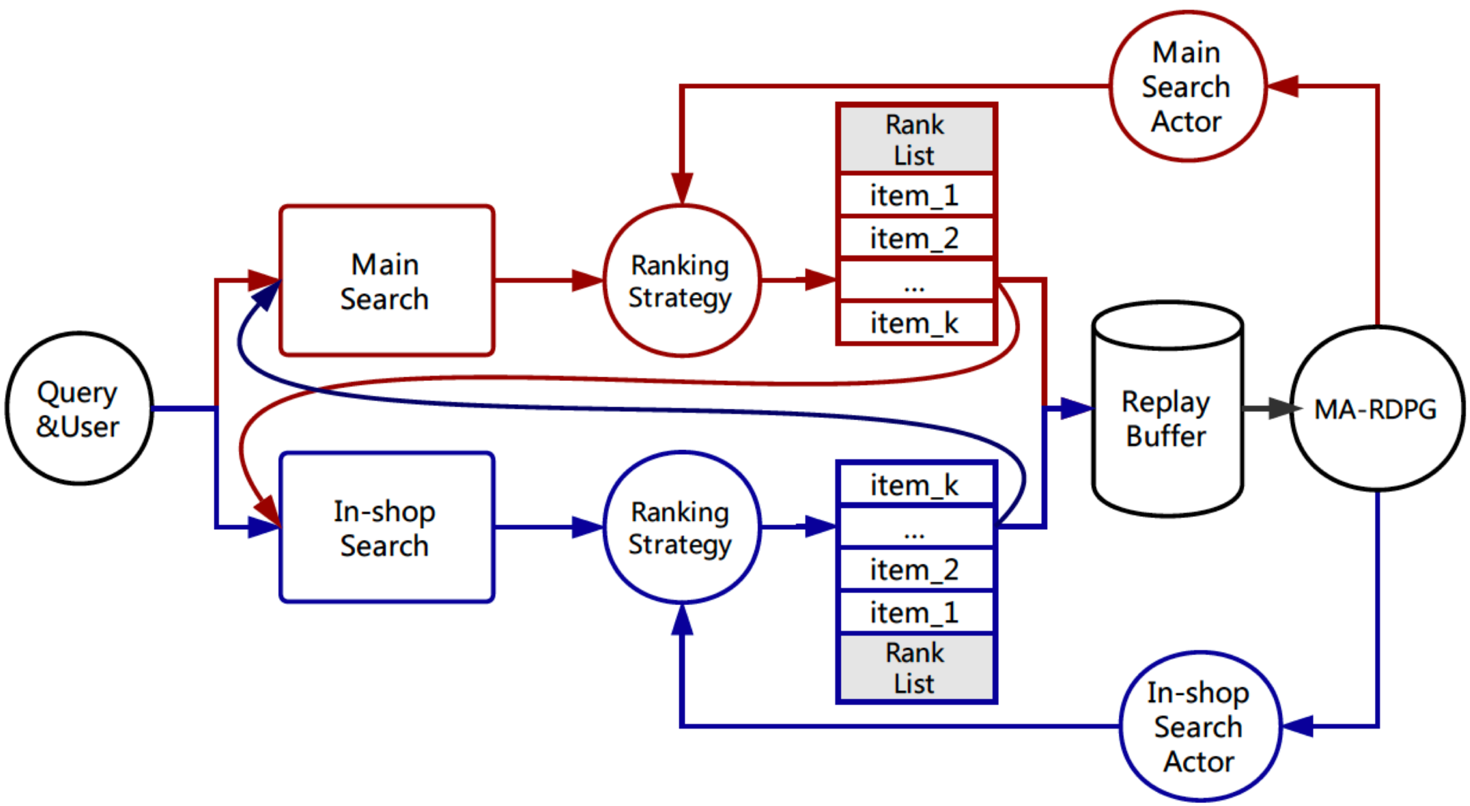
Ranking Opti. In Search (Feng et al., WWW2018)

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



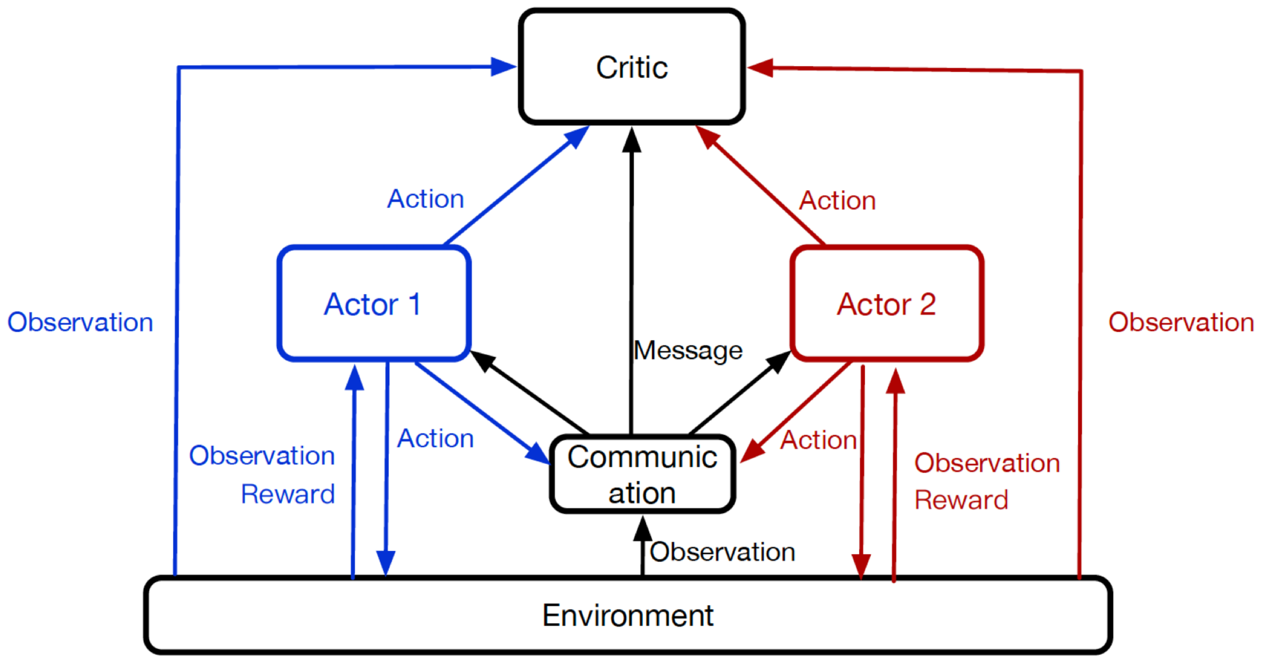
Ranking Opti. In Search (Feng et al., WWW2018)

- Joint Optimization of Multi-scenario Ranking



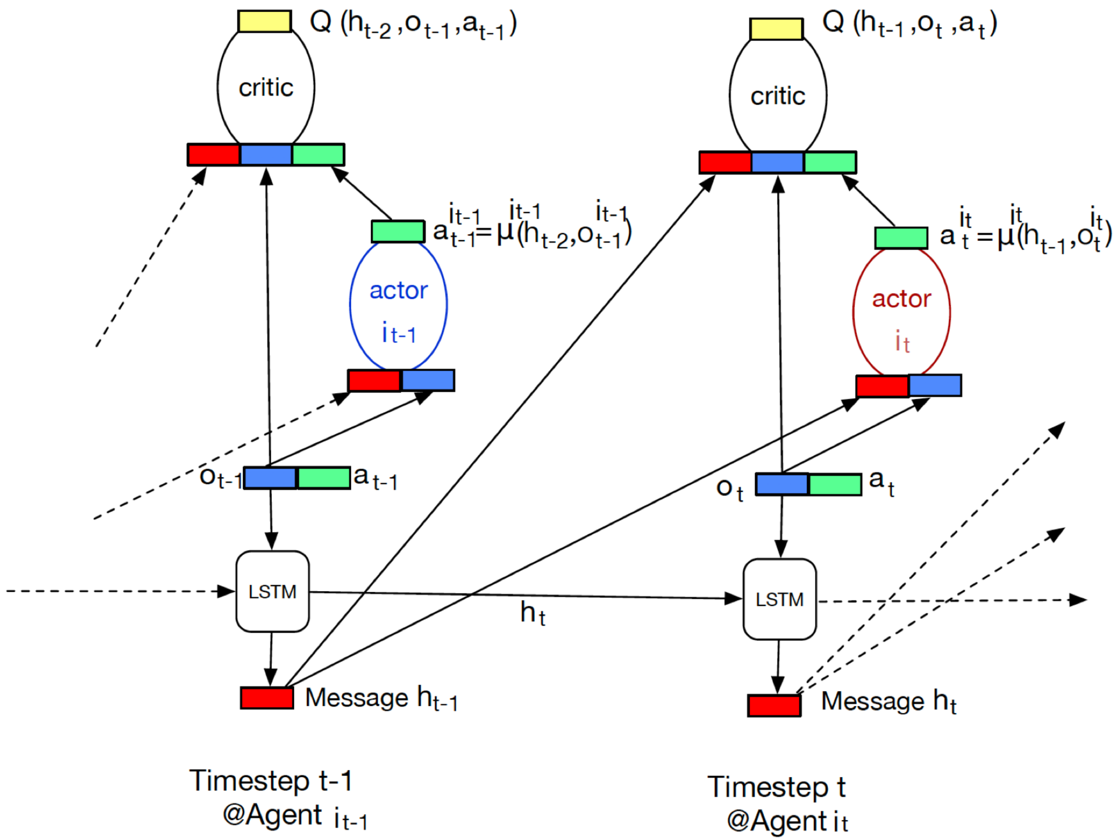
Model Overview

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



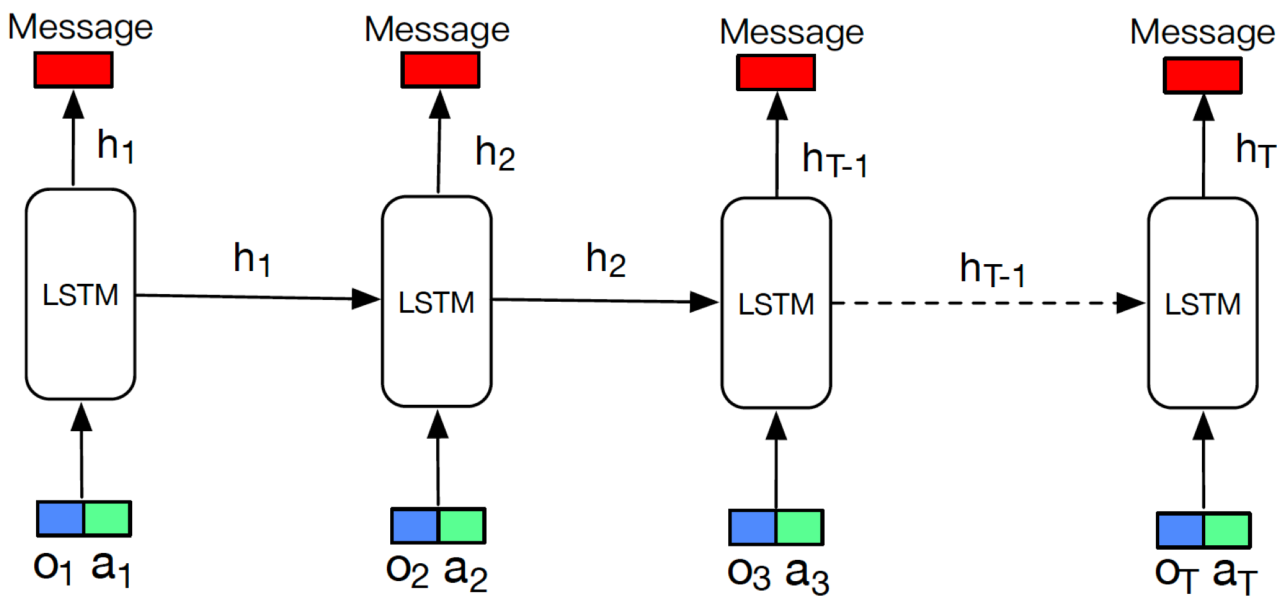
Model Structure

- 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

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

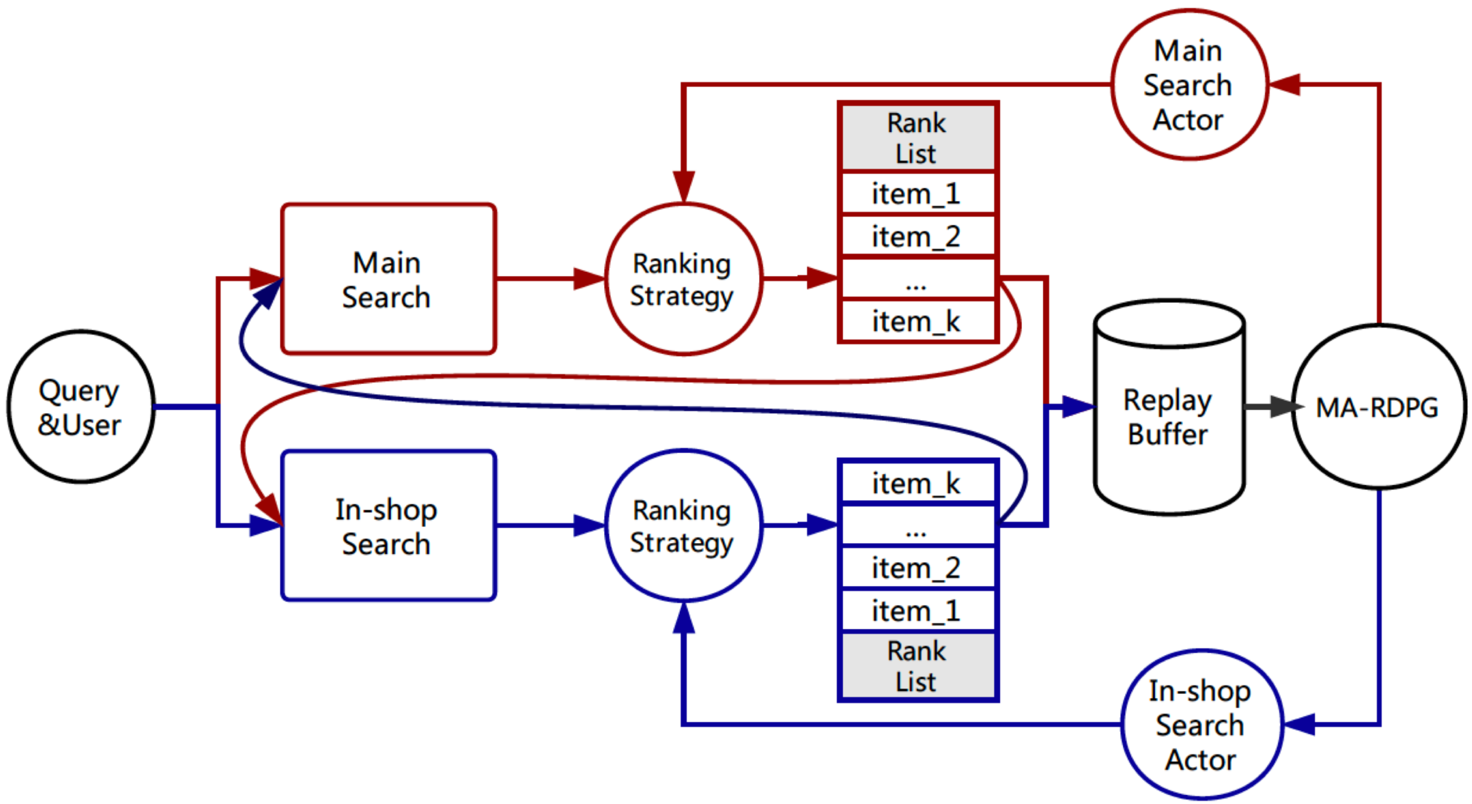
$$y_t = r_t + \gamma Q(h_t, o_{t+1}, \mu^{i_{t+1}}(h_t, o_{t+1}); \phi)$$

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

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

Application in Search

- 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

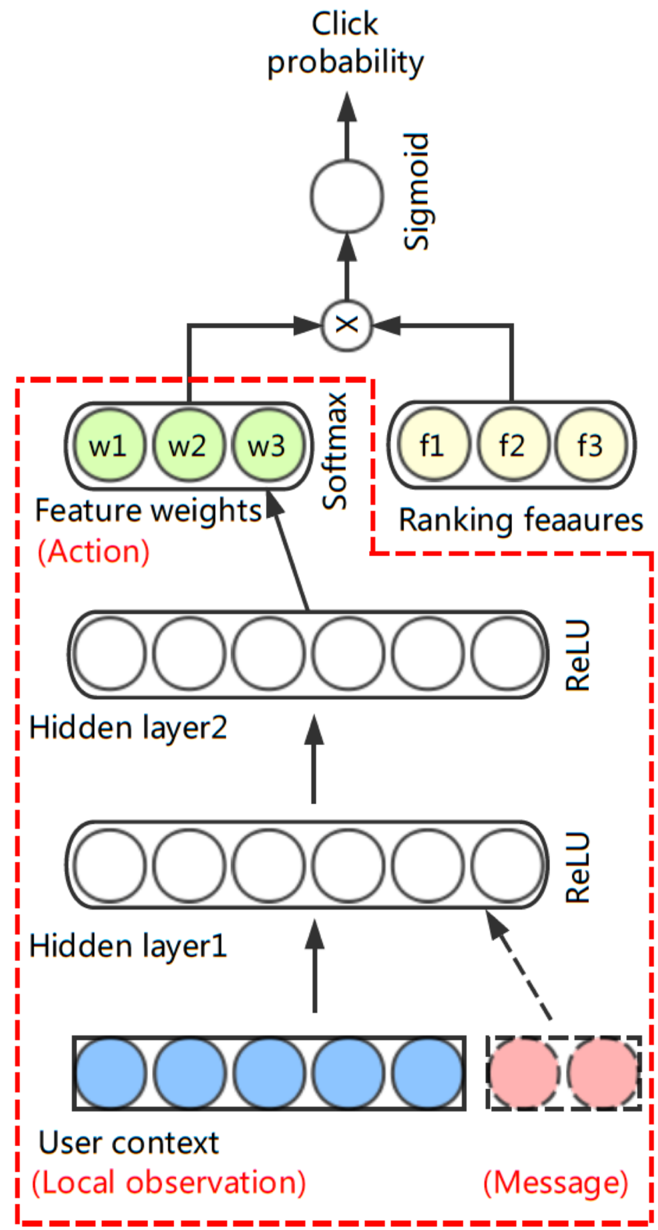
Application in Search

- 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

- GMV gap evaluated on an online Taobao platform

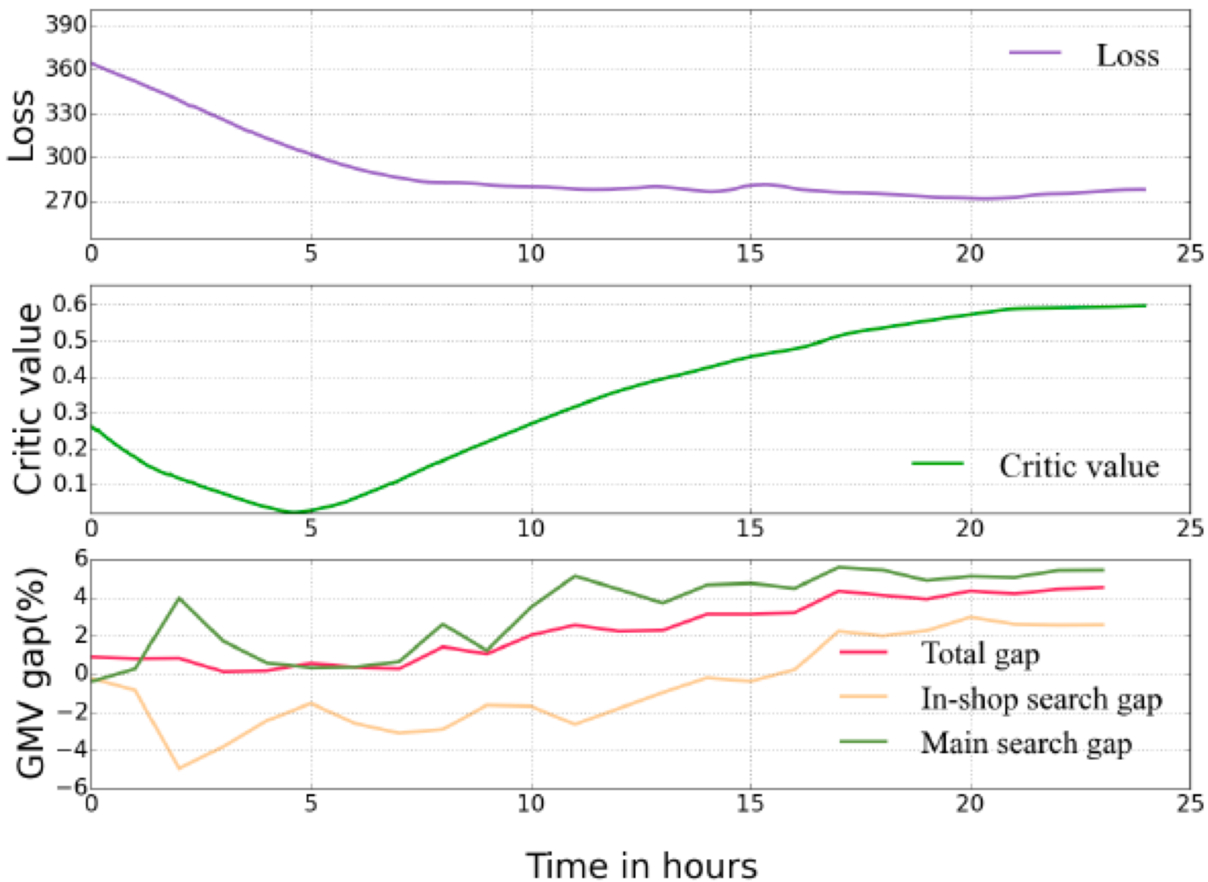
Relative improvement against EW+EW

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

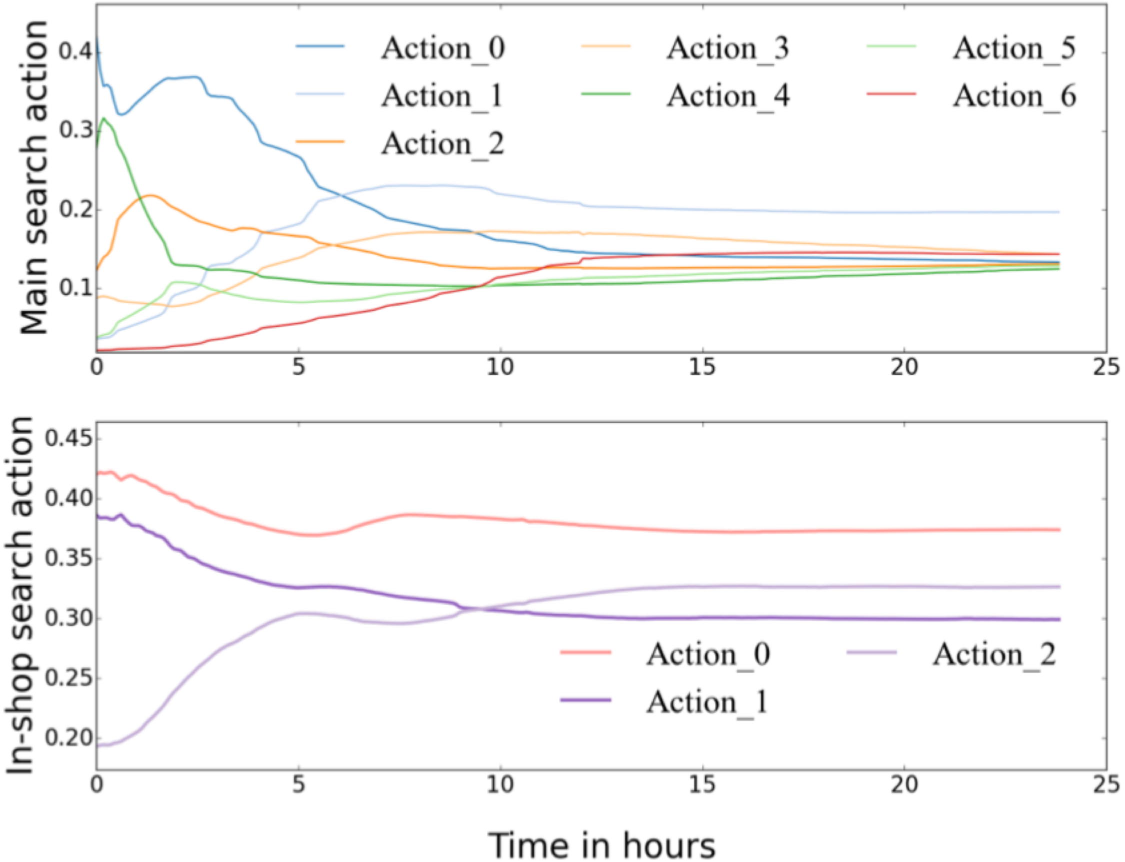
Experiment Results

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



Experiment Results

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



Summary

- **Search and Reasoning**: model structure, text structure, reasoning path, etc.
- **Instance Selection**: unlabeled data selection, data denoising, noisy label correction
- **Strategy Optimization**: ranking, dialogue strategy, language game, negotiation, text compression, language generation
- **How RL can facilitate NLP and search**

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 **domain 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**

Future Directions

- **Hierarchical DRL**: with planning ability
- **Inverse DRL**: estimate rewards from data
- **Sample-efficiency**: finding optimal solutions more efficiently

Thanks for Your Attention

- Minlie Huang, Tsinghua University
- aihuang@tsinghua.edu.cn
- http://coai.cs.tsinghua.edu.cn/hml