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

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

\[
q_\pi(s, a) = \mathbb{E}\left[ R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a, A_{t+1} \sim \pi \right]
\]
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, F₁)
  - 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

Diagram:

- \( a_1 \) to \( t_1 \), reward \( R_1 \)
- \( a_2 \) to \( t_2 \), reward \( R_2 \)
- \( a_3 \) to \( t_3 \), reward \( R_3 \)
- \( a_4 \) to \( t_4 \), reward \( R_4 \)

Agent scan
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

④ Tianyang Zhang, Minlie Huang, Li Zhao. Learning Structured Representation for Text Classification via Reinforcement Learning. AAAI 2018, New Orleans, Louisiana, USA.
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)

Sample architecture $A$ with probability $p$

The controller (RNN)

Trains a child network with architecture $A$ to get accuracy $R$

Compute gradient of $p$ and scale it by $R$ to update the controller

Layer N-1 → Layer N → Layer N+1

Number of Filters, Filter Height, Filter Width, Stride Height, Stride Width, Number of Filters, Filter Height
Neural Architecture Search (Zoph & Le, ICLR2017)

- **Reward R**: the accuracy of the configured model

- **REINFORCE algorithm**

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

\[
J(\theta_c) = E_{P(a_{1:T};\theta_c)} [R]
\]
Discovering Text Structures
(Zhang, Huang, Zhao; AAAI 2018)

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

<table>
<thead>
<tr>
<th>Origin text</th>
<th>Cho continues her exploration of the outer limits of raunch with considerable brio.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID-LSTM</td>
<td><strong>Cho continues her exploration</strong> of the outer limits of <strong>raunch</strong> with <strong>considerable brio</strong>.</td>
</tr>
<tr>
<td>HS-LSTM</td>
<td>Cho continues her exploration of the outer limits of raunch with considerable brio.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Origin text</th>
<th>Much smarter and more attentive than it first sets out to be.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID-LSTM</td>
<td><strong>Much smarter</strong> and <strong>more attentive</strong> than it first sets out to be.</td>
</tr>
<tr>
<td>HS-LSTM</td>
<td>Much smarter and more attentive than it first sets out to be.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Origin text</th>
<th>Offers an interesting look at the rapidly changing face of Beijing.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID-LSTM</td>
<td><strong>Offers an interesting look</strong> at the <strong>rapidly changing face of Beijing</strong>.</td>
</tr>
<tr>
<td>HS-LSTM</td>
<td>Offers an interesting look at the rapidly changing face of Beijing.</td>
</tr>
</tbody>
</table>

Challenges

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

- **Policy Network (PNet):**
  - Samples an action at each state
  - Two models: Information Distilled LSTM, Hierarchically Structured LSTM

- **Structured Representation Model:** transfer action sequence to representation

- **Classification Network (CNet):** 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:

\[
J(\Theta) = \mathbb{E}_{(s_t, a_t) \sim P_\Theta(s_t, a_t)} r(s_1 a_1 \cdots s_L a_L)
\]

\[
= \sum_{s_1 a_1 \cdots s_L a_L} P_\Theta(s_1 a_1 \cdots s_L a_L) R_L
\]

\[
= \sum_{s_1 a_1 \cdots s_L a_L} p(s_1) \prod_t \pi_\Theta(a_t | s_t) p(s_{t+1} | s_t, a_t) R_L
\]

\[
= \sum_{s_1 a_1 \cdots s_L a_L} \prod_t \pi_\Theta(a_t | s_t) R_L.
\]

- Update the policy network with policy gradient:

\[
\nabla_\Theta J(\Theta) = \sum_{t=1}^{L} R_L \nabla_\Theta \log \pi_\Theta(a_t | s_t)
\]
Classification Network (CNet)

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

\[
P(y|X) = \text{softmax}(W_s h_L + b_s),
\]

\[
\mathcal{L} = \sum_{X \in \mathcal{D}} - \sum_{y=1}^{K} \hat{p}(y, X) \log P(y|X)
\]
Information Distilled LSTM (ID-LSTM)

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

\[ P(y|X) = \text{softmax}(W_s h_L + b_s) \]
Information Distilled LSTM (ID-LSTM)

- **Action:** \{Retain, Delete\}
- **States:**
  \[ s_t = c_{t-1} \oplus h_{t-1} \oplus x_t, \]
- **Rewards:**
  \[ R_L = \log P(c_g|X) + \gamma L'/L. \]

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

```
Phrase-level

Word-level

Input words:  Do you hate it when that happens ?
Actions:  Inside End Inside End Inside Inside End End
```

Classification
Hierarchically Structured LSTM (HS-LSTM)

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

<table>
<thead>
<tr>
<th>$a_{t-1}$</th>
<th>$a_t$</th>
<th>Structure Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inside</td>
<td>Inside</td>
<td>A phrase continues at $x_t$.</td>
</tr>
<tr>
<td>Inside</td>
<td>End</td>
<td>A old phrase ends at $x_t$.</td>
</tr>
<tr>
<td>End</td>
<td>Inside</td>
<td>A new phrase begins at $x_t$.</td>
</tr>
<tr>
<td>End</td>
<td>End</td>
<td>$x_t$ is a single-word phrase.</td>
</tr>
</tbody>
</table>

- **States:** $s_t = c_{t-1}^p \oplus h_{t-1}^p \oplus c_t^w \oplus h_t^w$

Word-level LSTM: $c_t^w, h_t^w = \begin{cases} 
\Phi^w(0, 0, x_t), & a_{t-1} = \text{End} \\
\Phi^w(c_{t-1}^w, h_{t-1}^w, x_t), & a_{t-1} = \text{Inside} 
\end{cases}$

Phrase-level LSTM: $c_t^p, h_t^p = \begin{cases} 
\Phi^p(c_{t-1}^p, h_{t-1}^p, h_t^w), & a_t = \text{End} \\
c_t^p, h_{t-1}^p, & a_t = \text{Inside} 
\end{cases}$

- **Rewards:**

\[ R_L = \log P(c_g | X) - \gamma \left( \frac{L'}{L} + 0.1L/L' \right) \]

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

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

## Examples by ID-LSTM/HS-LSTM

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

- Input: query
  - (e1, r, ?)
  - (Colin Kaepernick, Nationality, ?)

- Output: answer entity
  - e2
  - USA

Search for Reasoning Path

Colin Kaepernick \( \xrightarrow{\text{BornInCity}} \) Milwaukee \( \xrightarrow{\text{CityIn}} \) USA
Model

- States: encodes the query, the answer, the current entity.
  \[ S = (e_t, e_{1q}, r_q, e_{2q}) \]

- Observations: the complete state of the environment is not observable, as the answer is not observed.
  \[ O(s = (e_t, e_{1q}, r_q, e_{2q})) = (e_t, e_{1q}, r_q) \]
Model

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

\[ A_S = \{(e_t, r, v) \in E : S = (e_t, e_{1q}, r_q, e_{2q}), r \in \mathcal{R}, v \in V\} \cup \{(s, \emptyset, s)\} \]

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

\[ R(S_T) = \mathbb{I}\{e_t = e_{2q}\} \]
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

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.
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

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)
  
  \[
  \text{[Obama]}_{e1} \text{ was born in the [United States]}_{e2}.
  \]
  
  Relation: \textit{BornIn}

- Distant Supervision (noisy labeling problem)
  
  \[
  \text{[Barack Obama]}_{e1} \text{ is the 44th President of the [United States]}_{e2}.
  \]
  
  Triple in knowledge base: <Barack Obama, \textit{BornIn}, United States>

  Relation: \textit{BornIn}
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?
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
- How to train the two modules jointly
Model Structure

Instance Selector

Sentence sequence

Policy function

Update Parameters

Reward function

Action

Relation Classifier

likelihood

Each selected sentence
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

\[
\pi_\Theta(s_i, a_i) = P_\Theta(a_i | s_i) \\
= a_i \sigma(W \cdot F(s_i) + b) \\
+ (1 - a_i)(1 - \sigma(W \cdot F(s_i) + b))
\]

- **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
    \[ J(\Theta) = V_\Theta(s_1 | B) \]
    \[ = E_{s_1,a_1,s_2,\ldots,s_i,a_i,s_{i+1}\ldots} \left[ \sum_{i=0}^{\left| B \right|+1} r(s_i | B) \right] \]
  - Update parameters with the REINFORCE algorithm
    \[ \Theta \leftarrow \Theta + \alpha \sum_{i=1}^{\left| B \right|} v_i \nabla \Theta \log \pi_\Theta(s_i, a_i) \]
Relation Classifier

- A CNN architecture to classify relations

\[ L = \text{CNN}(x) \]

\[ p(r|x; \Phi) = \text{softmax}(W_r \ast \text{tanh}(L) + b_r) \]

- 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

<table>
<thead>
<tr>
<th>Method</th>
<th>Macro $F_1$</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>0.40</td>
<td>0.60</td>
</tr>
<tr>
<td>CNN+Max</td>
<td>0.06</td>
<td>0.34</td>
</tr>
<tr>
<td>CNN+ATT</td>
<td>0.29</td>
<td>0.56</td>
</tr>
<tr>
<td>CNN+RL(ours)</td>
<td><strong>0.42</strong></td>
<td><strong>0.64</strong></td>
</tr>
</tbody>
</table>
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.

Payment-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 ⟩.
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)

<table>
<thead>
<tr>
<th>Datasets</th>
<th>SmartPhone</th>
<th>Clothing</th>
</tr>
</thead>
<tbody>
<tr>
<td># Topic category</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td># Training session</td>
<td>12,315</td>
<td>10,000</td>
</tr>
<tr>
<td># Training utterance</td>
<td>430,462</td>
<td>338,534</td>
</tr>
<tr>
<td># Gold-standard session</td>
<td>300</td>
<td>315</td>
</tr>
<tr>
<td># Gold-standard utterance</td>
<td>10,888</td>
<td>10,962</td>
</tr>
</tbody>
</table>

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 $\rightarrow$ learn policies with reward $\rightarrow$ refine data $\rightarrow$ learn better policies $\rightarrow$ 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
  \[
  r_{int} = \frac{1}{L-1} \text{sign}(a_{t-1} = a_t) \cos(h_{t-1}, h_t).
  \]

- **Global topic structure**: high content similarity within segments but low between segments
  \[
  r_{delayed} = \frac{1}{N} \sum_{\omega \in X} \frac{1}{|\omega|} \sum_{X_t \in \omega} \cos(h_t, \omega)
  \]
  \[
  - \frac{1}{N-1} \sum_{(\omega_{k-1}, \omega_k) \in X} \cos(\omega_{k-1}, \omega_k)
  \]
## Experiment

### (a) Topic Segmentation (MAE and WD)

<table>
<thead>
<tr>
<th>Model</th>
<th>Smartphone: MAE</th>
<th>Smartphone: WD</th>
<th>Clothing: MAE</th>
<th>Clothing: WD</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextTiling(TT)</td>
<td>13.09</td>
<td>.802</td>
<td>16.32</td>
<td>.948</td>
</tr>
<tr>
<td>TT+Embedding</td>
<td>3.59</td>
<td>.564</td>
<td>3.17</td>
<td>.567</td>
</tr>
<tr>
<td>STM</td>
<td>4.37</td>
<td>.505</td>
<td>8.85</td>
<td>.669</td>
</tr>
<tr>
<td>NL+HLSTM</td>
<td>8.25</td>
<td>.632</td>
<td>16.26</td>
<td>.925</td>
</tr>
<tr>
<td>Our method</td>
<td><strong>2.69</strong></td>
<td><strong>.415</strong></td>
<td><strong>2.74</strong></td>
<td><strong>.446</strong></td>
</tr>
</tbody>
</table>

### (b) Topic Labeling (Accuracy)

<table>
<thead>
<tr>
<th>Model</th>
<th>Smartphone</th>
<th>Clothing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword Matching</td>
<td>39.8</td>
<td>31.8</td>
</tr>
<tr>
<td>NL</td>
<td>51.4</td>
<td>39.0</td>
</tr>
<tr>
<td>NL+LSTM</td>
<td>49.6</td>
<td>35.5</td>
</tr>
<tr>
<td>NL+HLSTM</td>
<td>52.6</td>
<td>40.1</td>
</tr>
<tr>
<td>Our method</td>
<td><strong>62.2</strong></td>
<td><strong>48.0</strong></td>
</tr>
</tbody>
</table>

### (a) # Keywords per topic

<table>
<thead>
<tr>
<th>Model</th>
<th>3</th>
<th>6</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>45.0</td>
<td>51.4</td>
<td>48.0</td>
</tr>
<tr>
<td>NL+HLSTM</td>
<td>46.6</td>
<td>52.6</td>
<td>48.8</td>
</tr>
<tr>
<td>Our method</td>
<td><strong>55.3</strong></td>
<td><strong>62.2</strong></td>
<td><strong>58.2</strong></td>
</tr>
</tbody>
</table>

### (b) SubSets

<table>
<thead>
<tr>
<th>SubSets</th>
<th>KM</th>
<th>1-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utterances</td>
<td>3,503</td>
<td>7,385</td>
</tr>
</tbody>
</table>

### (c) Model Setting

<table>
<thead>
<tr>
<th>Model Setting</th>
<th>Segmentation</th>
<th>Labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAE</td>
<td>WD</td>
</tr>
<tr>
<td>RL + ( r_{int} )</td>
<td>3.04</td>
<td>.449</td>
</tr>
<tr>
<td>RL + ( r_{delayed} )</td>
<td>3.89</td>
<td>.490</td>
</tr>
<tr>
<td>RL + ( r_{int} + r_{delayed} )</td>
<td><strong>2.69</strong></td>
<td><strong>.415</strong></td>
</tr>
</tbody>
</table>
Experiment

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

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

Reinforce Learning NTM (Zaremba et al. 2015)
Reinforce Learning NTM
(Zaremba et al. 2015)

\[
J(\theta) = \sum_{[a_1, a_2, \ldots, a_T] \in A^t} p_\theta(a_1, a_2, \ldots, a_T)R(a_1, a_2, \ldots, a_T) = \sum_{a_{1:T} \in A^t} p_\theta(a_{1:T})R(a_{1:T})
\]
Language Generation
(Zhang & Lapata, EMNLP 2007)

\[ \hat{Y} = \hat{y}_1, \hat{y}_2, \hat{y}_3 \]

Get Action Seq. \( \hat{Y} \)

Update Agent

\[ X = x_1, x_2, x_3, x_4, x_5 \]

REINFORCE algorithm

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

\[
\nabla \mathcal{L}(\theta) \approx 
\sum_{t=1}^{|\hat{Y}|} \nabla \log P_{RL}(\hat{y}_t | \hat{y}_{1:t-1}, X)[r(\hat{y}_{1:|\hat{Y}|}) - b_t]
\]
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”
Multi-scenario Ranking: most large-scale online platforms or mobile Apps have multiple scenarios
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.

\[
a^i_t = \mu^i_t(s_t; \theta^i_t) \approx \mu^i_t(h_{t-1}, o^i_t; \theta^i_t)
\]

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

\[
Q(s_t, a^1_t, a^2_t, \ldots, a^N_t; \phi) = r_t + Q(s_{t+1}, a^1_{t+1}, a^2_{t+2}, \ldots, a^N_{t+1}; \phi)
\]
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

\[
J(\theta_{i_t}) = \mathbb{E}_{h_{t-1}, o_t} [Q(h_{t-1}, o_t, a_t; \phi) | a=\mu_{i_t}(h_{t-1}, o_t; \theta_{i_t})]
\]
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**

\[
a^i_t = \mu^i_t(s_t; \theta^i_t) \approx \mu^i_t(h_{t-1}, o^i_t; \theta^i_t)
\]
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

<table>
<thead>
<tr>
<th>day</th>
<th>EW + L2R</th>
<th>L2R + EW</th>
<th>L2R + L2R</th>
<th>MA-RDPG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>main</td>
<td>in-shop</td>
<td>total</td>
<td>main</td>
</tr>
<tr>
<td>1</td>
<td>0.04%</td>
<td>1.78%</td>
<td>0.58%</td>
<td>5.07%</td>
</tr>
<tr>
<td>2</td>
<td>0.01%</td>
<td>1.98%</td>
<td>0.62%</td>
<td>4.96%</td>
</tr>
<tr>
<td>3</td>
<td>0.08%</td>
<td>2.11%</td>
<td>0.71%</td>
<td>4.82%</td>
</tr>
<tr>
<td>4</td>
<td>0.09%</td>
<td>1.89%</td>
<td>0.64%</td>
<td>5.12%</td>
</tr>
<tr>
<td>5</td>
<td>-0.08%</td>
<td>2.24%</td>
<td>0.64%</td>
<td>4.88%</td>
</tr>
<tr>
<td>6</td>
<td>0.14%</td>
<td>2.23%</td>
<td>0.79%</td>
<td>5.07%</td>
</tr>
<tr>
<td>7</td>
<td>-0.06%</td>
<td>2.12%</td>
<td>0.62%</td>
<td>5.21%</td>
</tr>
<tr>
<td>avg.</td>
<td>0.03%</td>
<td>2.05%</td>
<td>0.66%</td>
<td>5.02%</td>
</tr>
</tbody>
</table>

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