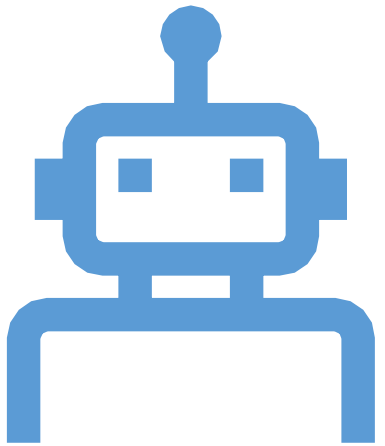
Reinforcement Learning in Extraction Summarization

——陈宏俊

Outline

多智体人工智能实验室
Multi-Agent Artificial Intelligence Laboratory

- Motivation
  - ➢Extraction Summarization
  - ➢Problems remained to solve
- RL in Extraction Summarization
- Conclusion



Motivation

Extraction Summarization

- Classification-based
- **Sequential-Labeling-based**

Problems remained to solve

- The disconnect between the **task definition** and the **training objective**
- The choice of **alternative** summaries
- The **coherency** of the summary
- The **exposure bias**
- The **preference** of earlier sentences

RL in Extraction Summarization

Ranking Sentences for Extractive Summarization with Reinforcement Learning

- Main Architecture

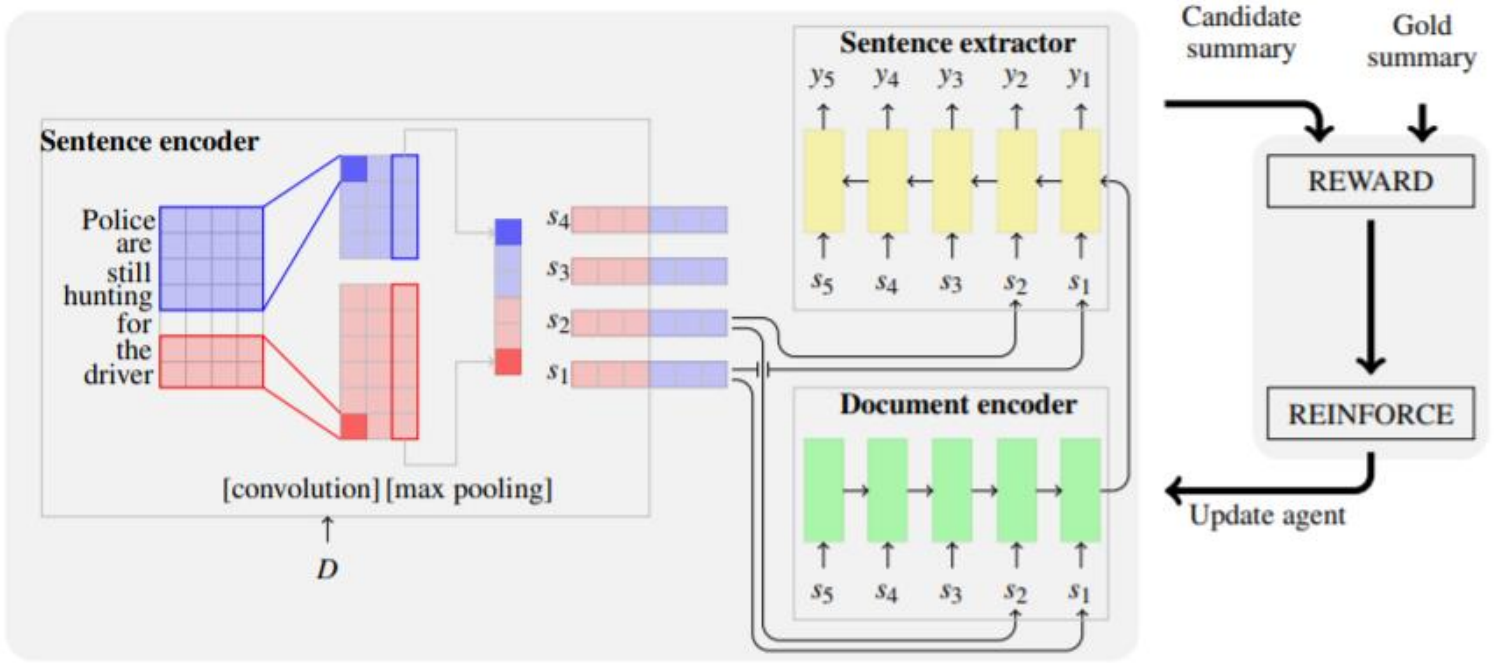


Figure 1: Extractive summarization model with reinforcement learning: a hierarchical encoder-decoder model ranks sentences for their extract-worthiness and a candidate summary is assembled from the top ranked sentences; the REWARD generator compares the candidate against the gold summary to give a reward which is used in the REINFORCE algorithm (Williams, 1992) to update the model.

Ranking Sentences for Extractive Summarization with Reinforcement Learning

- The Pitfalls of Cross-Entropy Loss $L(\theta) = -\sum_{i=1}^{n} \log p\left(y_i | s_i, D, \theta\right)$

sent. pos.	CNN article Sentences	Sent-level ROUGE	Individual Oracle	Collective Oracle	Multiple Collective Oracle
0	A debilitating, mosquito-borne virus called Chikungunya has made its way to North Carolina, health officials say.	21.2	1	1	(0,11,13) : 59.3 (0,13) : 57.5
1	It's the state's first reported case of the virus.	18.1	1	0	(11,13) : 57.2
2	The patient was likely infected in the Caribbean, according to the Forsyth County Department of Public Health.	11.2	1	0	(0,1,13) : 57.1 (1,13) : 56.6
3	Chikungunya is primarily found in Africa, East Asia and the Caribbean islands, but the Centers for Disease Control and Prevention has been watching the virus,+ for fear that it could take hold in the United States – much like West Nile did more than a decade ago.	35.6	1	0	(3,11,13) : 55.0 (13) : 54.5 (0,3,13) : 54.2
4	The virus, which can cause joint pain and arthritis-like symptoms, has been on the U.S. public health radar for some time.	16.7	1	0	(3,13) : 53.4 (1,3,13) : 52.9
5	About 25 to 28 infected travelers bring it to the United States each year, said Roger Nasci, chief of the CDC's Arboviral Disease Branch in the Division of Vector-Borne Diseases.	9.7	0	0	(1,11,13) : 52.0 (0,9,13) : 51.3 (0,7,13) : 51.3
6	"We haven't had any locally transmitted cases in the U.S. thus far," Nasci said.	7.4	0	0	(0,12,13) : 51.0
7	But a major outbreak in the Caribbean this year – with more than 100,000 cases reported – has health officials concerned.	16.4	1	0	(9,11,13) : 50.4 (1,9,13) : 50.1
8	Experts say American tourists are bringing Chikungunya back home, and it's just a matter of time before it starts to spread within the United States.	10.6	0	0	(12,13) : 49.3 (7,11,13) : 47.8
9	After all, the Caribbean is a popular one with American tourists, and summer is fast approaching.	13.9	1	0	(0,10,13) : 47.8 (11,12,13):47.7
10	"So far this year we've recorded eight travel-associated cases, and seven of them have come from countries in the Caribbean where we know the virus is being transmitted," Nasci said.	18.4	1	0	(7,13) : 47.6 (9,13) : 47.5 (1,7,13) : 46.9
11	Other states have also reported cases of Chikungunya.	13.4	0	1	(3,7,13) : 46.0
12	The Tennessee Department of Health said the state has had multiple cases of the virus in people who have traveled to the Caribbean.	15.6	1	0	(3,12,13) : 46.0 (3,9,13) : 45.9
13	The virus is not deadly, but it can be painful, with symptoms lasting for weeks.	54.5	1	1	(10,13) : 45.5
14	Those with weak immune systems, such as the elderly, are more likely to suffer from the virus' side effects than those who are healthier.	5.5	0	0	(4,11,13) : 45.3 ...

Story Highlights
• North Carolina reports first case of mosquito-borne virus called Chikungunya • Chikungunya is primarily found in Africa, East Asia and the Caribbean islands • Virus is not deadly, but it can be painful, with symptoms lasting for weeks

Ranking Sentences for Extractive Summarization with Reinforcement Learning

- RL in Sentence Ranking
  - Agent: Model
  - Environment: Document
  - Policy: getting the sentences input and predicting the score of relevance
  - Reward: Similarity between gold summary and generated one

$$\nabla L(\theta) = -\mathbb{E}_{\hat{y}\sim\ p_\theta}[r(\hat{y})\nabla \log p(\hat{y}|D,\theta)]$$

  - Better exploration in the search space of summary
  - Directly optimizing the evaluation metric
  - Better at discriminating among sentences for the final summary

Ranking Sentences for Extractive Summarization with Reinforcement Learning

- Training with High Probability Samples

$$\nabla L(\theta) \approx -r(\hat{y}) \nabla \log p(\hat{y}|D,\theta)$$

$$\approx -r(\hat{y}) \sum_{i=1}^{n} \nabla \log p(\hat{y}_i|s_i,D,\theta)$$

Learning to Extract Coherent Summary via Deep Reinforcement Learning

- NES

Since NES is trained with supervised learning, ground truth extraction labels $(\mathring{y}_1, \cdots, \mathring{y}_n)$ are available during training. Then the representation of sentences selected before or at time t is

$$\mathbf{g}_t = \mathbf{g}_{t-1} + \mathring{y}_t \tanh(\mathbf{W}_g \overleftrightarrow{\mathbf{h}}_t).$$

The NES model is pretrained by minimizing the negative log-likelihood of the ground truth extraction labels

$$L_{\text{pretrain}}(\Theta) = -\sum_{i=1}^{N}\sum_{t=1}^{N_i} \left[\mathring{y}_t^i \log \Pr(y_t^i = 1 | X_i, \mathring{y}_{1:t-1}^i) + (1 - \mathring{y}_t^i) \log \Pr(y_t^i = 0 | X_i, \mathring{y}_{1:t-1}^i)\right].$$

Learning to Extract Coherent Summary via Deep Reinforcement Learning

- RNES

We use the REINFORCE algorithm to train our RNES model. It is a kind of policy gradient method proposed by (Williams 1992), and it maximizes the performance of the agent by updating its policy parameters. The policy is defined as the probability of taking an action at time t given a state, which is parameterized by Θ:

$$\pi(a|s_t, \Theta) \overset{\text{def}}{=} \Pr(y_t = a|s_t, \Theta) \overset{\text{def}}{=} \Pr(y_t = a|X, y_{1:t-1}, \Theta).$$

In our case, Θ represents all the parameters in the RNES model. We use a shorthand π_Θ to denote the policy π parameterized by Θ. By applying Equation 1, we have

$$\pi_\Theta(a = 1|s_t) = \text{MLP}_\Theta(\overleftrightarrow{\mathbf{h}}_t, \mathbf{g}_{t-1}, \mathbf{d}).$$

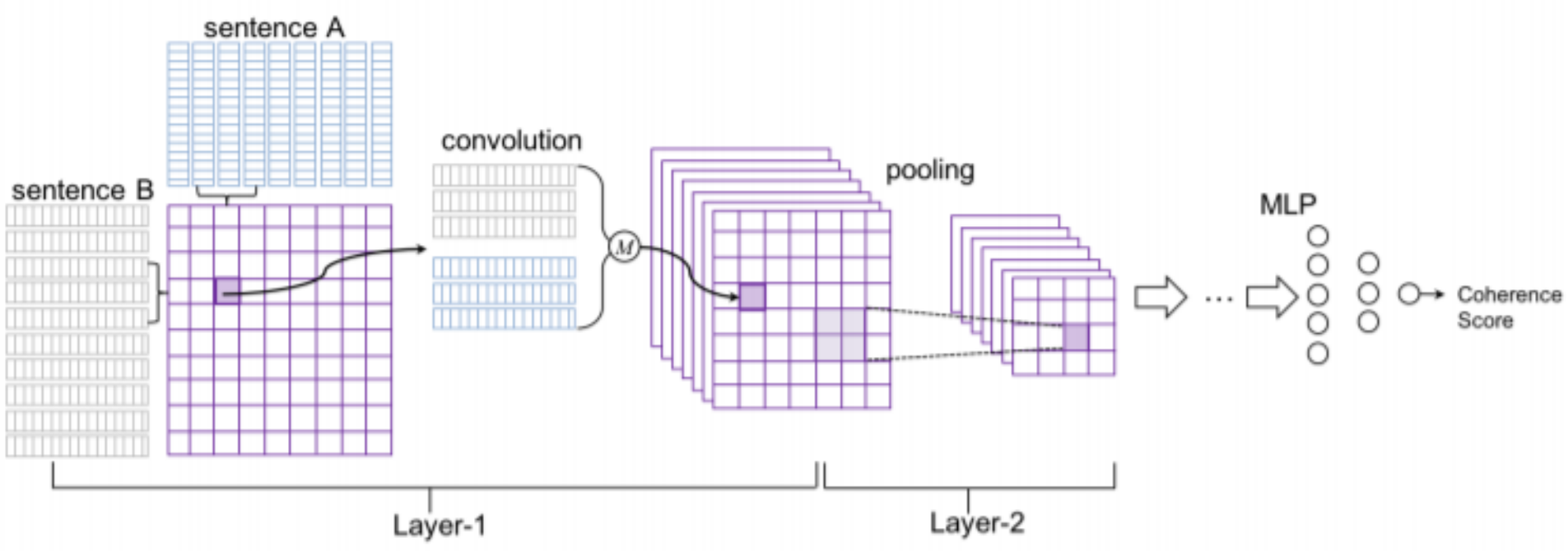
Maximizing the weight sum:

$$v_\pi(s_0) \overset{\text{def}}{=} \mathbb{E}_{\pi_\Theta}[R_0|s_0] = \mathbb{E}_{\tilde{y}_t, \tilde{s}_t \sim \pi_\Theta}[\tilde{r}_{-1} + \lambda \sum_{t=1}^{n} \tilde{r}_t|s_0]$$

$$= \mathbb{E}_{\pi_\Theta}[\text{ROUGE}(\tilde{G}) + \lambda \text{Coherence}(\tilde{G})|X]$$

Learning to Extract Coherent Summary via Deep Reinforcement Learning

- Main Architecture



Learning to Extract Coherent Summary via Deep Reinforcement Learning

- Training for Coherence Model

For the training of the neural coherence model, we use a pair-wise training strategy with a large margin objective. Suppose we are given the following triples $(S_A, {S_B}^+, {S_B}^-)$, we adopt the ranking-based loss as objective:

$$L_\Theta(S_A, {S_B}^+, {S_B}^-) = \max(0, 1 + \mathrm{Coh}(S_A, {S_B}^-) - \mathrm{Coh}(S_A, {S_B}^+)).$$

BANDITSUM: Extractive Summarization as a Contextual Bandit

A policy for extractive summarization is a neural network $p_\theta(\cdot|d)$, parameterized by a vector θ, which, for each input document d, yields a probability distribution over index sequences. Our goal is to find parameters θ which cause $p_\theta(\cdot|d)$ to assign high probability to index sequences that induce extractive summaries that a human reader would judge to be of high-quality. We achieve this by maximizing the following objective function with respect to parameters θ:

$$J(\theta) = E\left[R(i,a)\right] \tag{1}$$

where the expectation is taken over documents d paired with gold-standard abstractive summaries a, as well as over index sequences i generated according to $p_\theta(\cdot|d)$.

Policy Gradient Reinforcement Learning

$$\nabla_\theta J(\theta) = E\left[\nabla_\theta \log p_\theta(i|d) R(i,a)\right] \tag{2}$$

where the expectation is taken over the same variables as (1).

Since we typically do not know the exact document distribution and thus cannot evaluate the expected value in (2), we instead estimate it by sampling. We found that we obtained the best performance when, for each update, we first sample one document/summary pair (d, a), then sample B index sequences $i^1, \ldots, i^B$ from $p_\theta(\cdot|d)$, and finally take the empirical average:

$$\nabla_\theta J(\theta) \approx \frac{1}{B} \sum_{b=1}^{B} \nabla_\theta \log p_\theta(i^b|d) R(i^b,a) \tag{3}$$

BANDITSUM: Extractive Summarization as a Contextual Bandit

Structure of $p_\theta(\cdot|d)$

$$p_\theta(\cdot|d) = \mu(\cdot|\pi_\theta(d))$$

At each step of sampling-without-replacement, we also include a small probability ϵ of sampling uniformly from all remaining sentences. This is used to achieve adequate exploration during training, and is similar to the ϵ-greedy technique from reinforcement learning.

BANDITSUM:
Extractive Summarization as a Contextual Bandit

Baseline for Variance Reduction

Using a baseline $\overline{r}$, our sample-based estimate of $\nabla_\theta J(\theta)$ becomes:

$$\frac{1}{B}\sum_{i=1}^{B} \nabla_\theta \log p_\theta(i^b|d)(R(i^b,a) - \overline{r}) \qquad (6)$$

BANDITSUM:
Extractive Summarization as a Contextual Bandit

Reward Function

$$R(i,a) = \frac{1}{3}(\text{ROUGE-1}_f(i,a) + \text{ROUGE-2}_f(i,a) + \text{ROUGE-L}_f(i,a)).$$

BANDITSUM: Extractive Summarization as a Contextual Bandit

Model

In this section, we discuss the concrete instantiations of the neural network π_θ that we use in our experiments. We break π_θ up into two components: a document encoder $f_{\theta 1}$, which outputs a sequence of sentence feature vectors $(h_1, \ldots, h_{N_d})$ and a decoder $g_{\theta 2}$ which yields sentence affinities:

$$h_1, \ldots, h_{N_d} = f_{\theta 1}(d) \tag{9}$$

$$\pi_\theta(d) = g_{\theta 2}(h_1, \ldots, h_{N_d}) \tag{10}$$

Encoder-Decoder

Thank you

多智体人工智能实验室
Multi-Agent Artificial Intelligence Laboratory

- End