LLMR: Knowledge Distillation with a Large Language Model-Induced Reward

Dongheng Li dongheng@ualberta.ca

Yongchang Hao yongcha1@ualberta.ca

Lili Mou doublepower.mou@gmail.com

Introduction

Prompt generation with LLMs

- An efficient way to use LLM to perform downstream tasks:

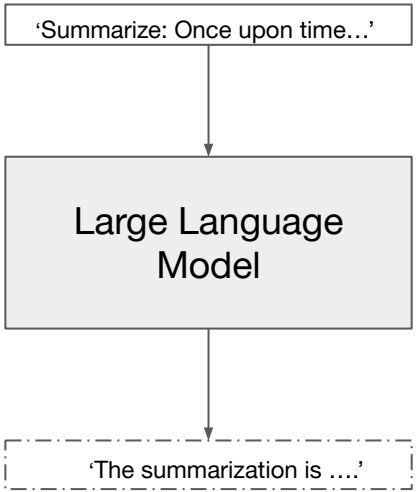
  Dialogue generation:

  'What is the dialogue response: $x_1, x_2 \ldots, x_t$'

  Text summarization:

  'Summarize the following text: $x_1, x_2 \ldots, x_t$'

- LLMs are very expensive!



Introduction

Knowledge distillation towards LLMs

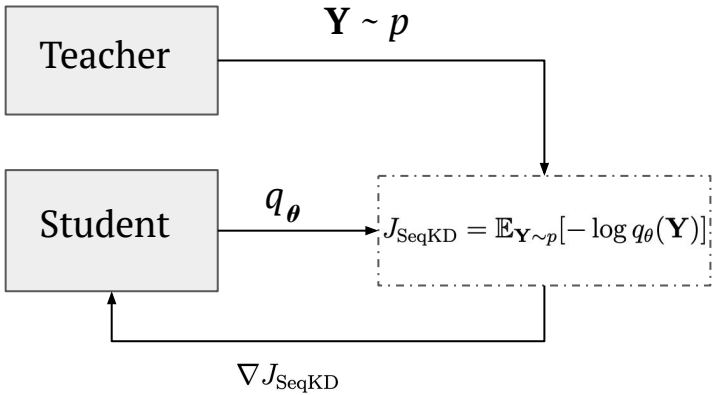
- Knowledge Distillation (Hinton et al.,2015)

  SeqKD (Kim and Rush, 2016)

  F-divergence KD (Wen et al., 2024)

- Exposure bias (Ranzato et al., 2016):

Error accumulations in sequential decisions due to discrepancy between training and inference.



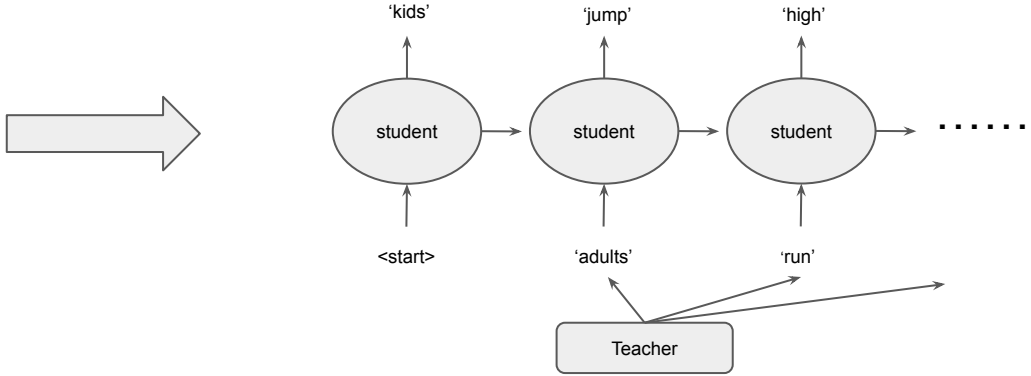
Introduction

Exposure bias in KD of LLMs

In knowledge distillation, where teacher (P) and student (Q).

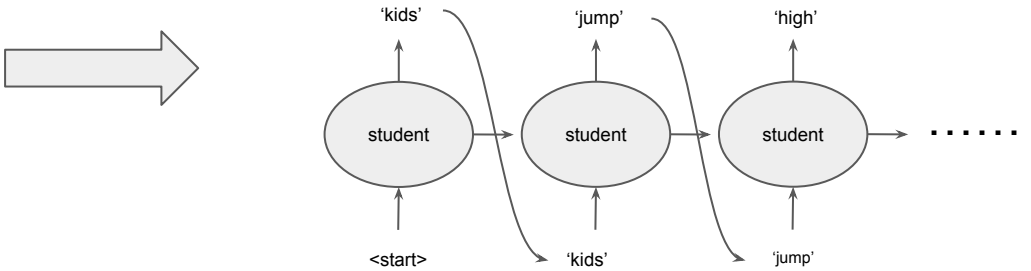
- During training:

$$y_{<t} \sim P$$



- During inference:

$$y_{<t} \sim Q$$



Intuition

- High exposure bias leads to generation degradation (Chiang and Chen, 2021; Ranzato et al., 2016).

- Our idea:

  Make the student participate during training

Approach

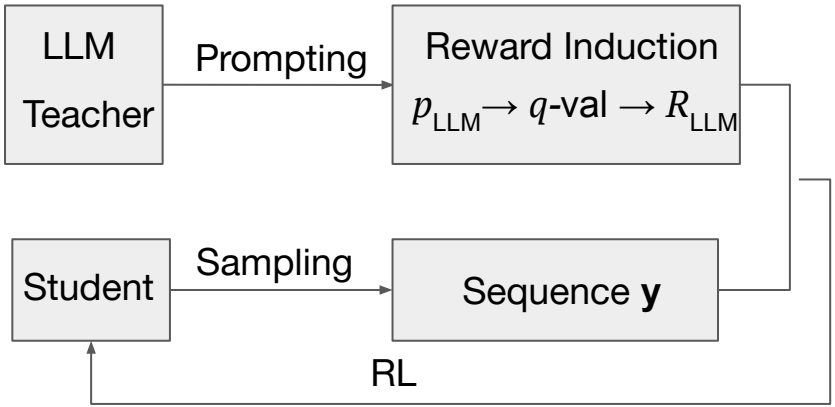
Reinforcement learning based KD

- Knowledge distillation with reinforcement learning.

  Teacher : LLM ($\mathrm{p}_{\mathrm{LLM}}$)

  Student : small language model ($\mathrm{q}_{\boldsymbol{\theta}}$)

- Objective for the KD:



$$\text{maximize}_{\boldsymbol{\theta}}\, \mathbb{E}_{\mathbf{y}\sim q_\theta}[R_{\mathrm{LLM}}(\mathbf{y})]$$

Approach

Reinforcement learning based KD

Problem formulation:

Markov decision process (MDP) formulation for general text generation:

Text generation as MDP process:

- (S,A,T,R)
- S: Partial sequence so far
- A: Next token
- T: Probability of current state to the next state
- R: A function that defines goodness based on a state and an action

Approach

Reinforcement learning based KD

Problem formulation:

Markov decision process (MDP) formulation for knowledge distillation:

Text generation as MDP process:

- (S,A,T,R)
- S: Partial sequence so far
- A: Next token
- T: Probability of current state to the next state
- R: Reward function induced from teacher model

Approach

Reward design

- Prompt-based reward induction consider task $\mathcal{T}$

  Dialogue generation:

  $\texttt{pmt}_{\mathcal{T}}(\mathbf{x})$ = ***'What is the dialogue response:*** $x_1, x_2 \ldots, x_t$'

- Given any candidate output $\mathbf{y} = (y_1, \cdots, y_T)$ from the student, we wish there is a $R_{\text{LLM}}(\mathbf{y})$ that evaluates the 'goodness' of $\mathbf{y}$

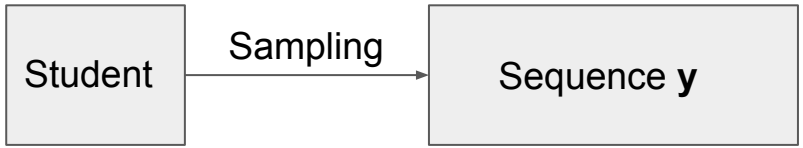
Approach

Reward design

Step 1:

- Sample the candidate output $\mathbf{y} = (y_1, \cdots, y_T)$ from the student model ($q_{\boldsymbol{\theta}}$):

$$\mathbf{y} \sim q_{\boldsymbol{\theta}}$$



Approach

Induce a reward based on LLM:

Step 2.1:
Get a policy:

- Querying the LLM step-by-step to obtain next word probability (policy):

$$\pi(a|s) = p_{\text{LLM}}(\mathbf{y}_t|\mathbf{y}_{<t}, \texttt{pmt}_{\mathcal{T}}(\mathbf{x}))$$

Step 2.2:
Induce a q-value function ($q\texttt{-val}$):

- **Assumption 1**: Optimal policy are parameterised by a Boltzmann distribution (Ahmed and Devanbu, 2022):

$$\pi(a|s) = \frac{\exp\{q(s,a)\}}{\sum_{a'} \exp\{q(s,a')\}}$$

Approach

Reward design

Step 2.2:

- Induce q-value function $q\text{-}\mathtt{val}$ by reparameterization:

$$p_{\mathrm{LLM}}(\mathrm{y}_t|\mathbf{y}_{<t}, \mathtt{pmt}_{\mathcal{T}}(\mathbf{x})) = \frac{\exp\{q\text{-}\mathtt{val}(\mathrm{y}_t; \mathbf{y}_{<t})\}}{\sum_{\mathrm{y}'} \exp\{q\text{-}\mathtt{val}(\mathrm{y}'; \mathbf{y}_{<t})\}}$$

- We get the q-value function represented by LLM's logits function:

$$q\text{-}\mathtt{val}(\mathrm{y}_t; \mathbf{y}_{<t}) = f_{\mathrm{LLM}}(\mathrm{y}_t; \mathbf{y}_{<t})$$

Approach

Reward design

Step 2.3:

Induce a reward from q-value:

- Bellman optimality equation:

$$q^*(s,a) = r(s,a) + \sum_{s' \in \mathcal{S}} T(s'|s,a) \max_{a'} q^*(s',a')$$
$$= r(s,a) + \max_{a'} q^*(s',a')$$

- **Overall we get the reward function:**

$$R_{\text{LLM}}(\mathrm{y}_t; \mathbf{y}_{<t}) = q\text{-}\texttt{val}(\mathrm{y}_t; \mathbf{y}_{<t}) - \max_{\mathrm{y}'} q\text{-}\texttt{val}(\mathrm{y}'; \mathbf{y}_{<t+1})$$
$$= f_{\text{LLM}}(\mathrm{y}_t; \mathbf{y}_{<t}) - \max_{\mathrm{y}'} f_{\text{LLM}}(\mathrm{y}'; \mathbf{y}_{<t+1})$$

Approach

REINFORCE

- With the reward function induced from prompt $R_{\text{LLM}}(\mathrm{y}_t; \mathbf{y}_{<t})$:

$$R_{\text{LLM}}(\mathbf{y}) = \sum\nolimits_t R_{\text{LLM}}(\mathrm{y}_t; \mathbf{y}_{<t})$$

- REINFORCE algorithm is applied:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \mathbb{E}_{\pi_\theta} \left[\sum_t R_{\text{LLM}}(\mathrm{y}_t; \mathbf{y}_{<t}) \right]$$

Approach

REINFORCE

Step 3:

- REINFORCE algorithm is applied:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \mathbb{E}_{\pi_\theta} \left[\sum_t R_{\text{LLM}}(\mathrm{y}_t; \mathbf{y}_{<t}) \right]$$

- The gradient of the expected reward:

$$\nabla_\theta \mathbb{E}_{\pi_\theta} \left[\sum_t R_{\text{LLM}}(\mathrm{y}_t; \mathbf{y}_{<t}) \right] = \mathbb{E}_{\pi_\theta} \left[\sum_t G_t(\mathbf{y}) \nabla_\theta \log \pi_\theta(\mathrm{y}_t; \mathbf{y}_{<t}) \right]$$

where $G_t(\mathbf{y}) := \sum_{t^*=t}^{T} R_{\text{LLM}}(\mathrm{y}_{t^*}; \mathbf{y}_{<t^*})$

Approach

Summary

- Our LLMR is an improved KD method in a few aspects:
    - Student participates in training
    - Task-agnostic reward function for RL induced from prompting
- LLMR can achieve a better generation performance and a lower exposure bias.

Experiments

General setting

- Setting:
  - Teacher: prompt T0-3b(3 billion)
  - Student: T5-base(220 million)
- Tasks:
  - Dialogue generation:
    - Daily Dialogue
    - Open Subtitles
  - Prompt: ***'What is the dialogue response of*** 'x_1,x_2…,x_t***':***
  - Summarization:
    - CNN/Daily Mail
  - Prompt: ***'Summarize*** 'x_1,x_2…,x_t***':***

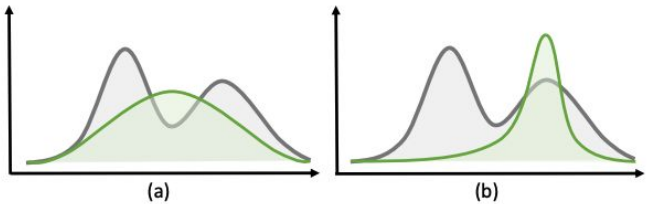
Baseline selections (Wen et al., 2023)

Kullback–Leibler distillation (KL):

$$J_{\mathrm{KL}} = D_{\mathrm{KL}}(p \| q_\theta) = \mathbb{E}_{\mathbf{Y} \sim p} \left[\log \frac{p(\mathbf{Y})}{q_\theta(\mathbf{Y})} \right]$$

Reverse KL distillation (RKL):

$$J_{\mathrm{RKL}} = D_{\mathrm{KL}}(q_\theta \| p) = \mathbb{E}_{\mathbf{Y}' \sim q_\theta} \left[\log \frac{q_\theta(\mathbf{Y}')}{p(\mathbf{Y}')} \right]$$



Jenson–Shannon distillation (JS):

$$J_{\mathrm{JS}} = \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{Y} \sim p} \left[\log \frac{p(\mathbf{Y})}{m(\mathbf{Y})} \right] + \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{Y}' \sim q_\theta} \left[\log \frac{q_\theta(\mathbf{Y}')}{m(\mathbf{Y}')} \right]$$

Main Results (general generation ability)

	Model		DailyDialog		OpenSubtitles		CNN/DailyMail		
			BLEU2	BLEU4	BLEU2	BLEU4	ROUGE-1	ROUGE-2	ROUGE-L
1	Prompting Teacher		5.57	1.49	4.67	1.51	36.16	14.99	24.05
2	Prompting Student		1.35	0.31	1.21	0.25	21.23	6.73	17.88
3	Distilled Students	SeqKD	6.19	1.71	3.87	1.35	35.46	14.52	23.68
4		KL	5.03	1.40	3.84	1.33	34.11	14.21	22.83
5		RKL	5.02	1.29	4.12	1.36	32.07	13.77	22.87
6		JS	6.60	1.73	3.64	0.87	35.88	14.72	23.97
7		Our LLMR	**7.00**	**1.88**	**5.13**	**1.85**	**36.42**	**15.21**	**24.83**

Table 1: Main results on dialogue generation and summarization tasks.

Diversity analysis

	DailyDialog		OpenSubtitles		CNN/DailyMail	
Model	Dist1	Dist2	Dist1	Dist2	Dist1	Dist2
SeqKD	4.93	27.37	4.78	23.15	3.86	33.59
KL	4.76	26.77	4.99	24.00	3.76	33.59
RKL	5.76	29.01	5.38	23.72	4.07	32.27
JS	5.84	32.25	4.44	19.21	3.83	31.47
Our LLMR	**6.02**	**34.83**	**5.82**	**27.21**	**4.20**	**35.38**

Quantification of text generational diversity (Li et al., 2016):

- Distinct n-gram measure (Distinct-n)

$$\text{Distinct-}n = \frac{\text{Number of unique } n\text{-grams}}{\text{Total number of } n\text{-grams}}$$

Exposure Analysis

Quantification of exposure bias (Arora et al., 2022)

- Excess Error Percentage (ExError%)

In our case:

$$\text{ExError\%}(l) = \frac{D_s(l) - D_t(l)}{D_t(l)} \times 100\%$$

Exposure Analysis

Quantification of exposure bias (Arora et al., 2022)

- Excess Error Percentage (ExError%)

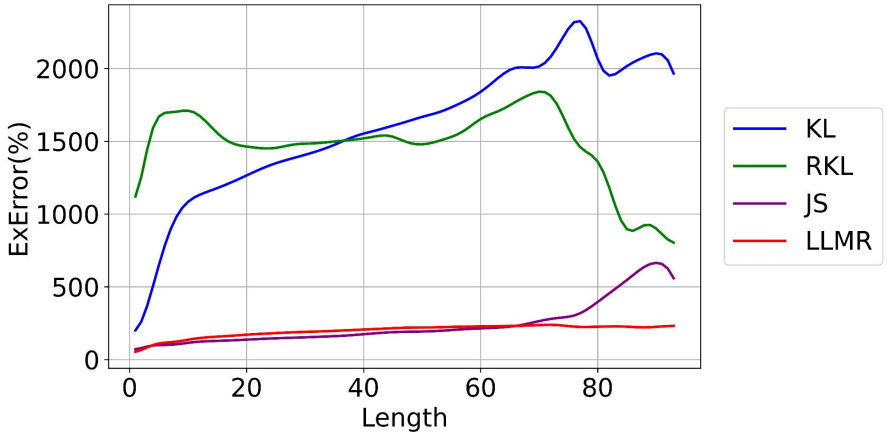
In our case:

$$\text{ExError\%}(l) = \frac{D_s(l) - D_t(l)}{D_t(l)} \times 100\%$$

$$D_s(l) = \sum_{t=1}^{T} \mathbb{E}_{\substack{\mathsf{y}_{<t} \sim q_\theta(\cdot|\mathsf{x}) \\ \mathsf{y}_t \sim p(\cdot|\mathbf{y}_{<t}, \mathsf{x})}} \left[\log \frac{p(\mathsf{y}_t|\mathbf{y}_{<t}, \mathsf{x})}{q_\theta(\mathsf{y}_t|\mathbf{y}_{<t}, \mathsf{x})} \right]$$

$$D_t(l) = \sum_{t=1}^{T} \mathbb{E}_{\substack{\mathsf{y}_{<t} \sim p(\cdot|\mathsf{x}) \\ \mathsf{y}_t \sim p(\cdot|\mathbf{y}_{<t}, \mathsf{x})}} \left[\log \frac{p(\mathsf{y}_t|\mathbf{y}_{<t}, \mathsf{x})}{q_\theta(\mathsf{y}_t|\mathbf{y}_{<t}, \mathsf{x})} \right]$$

Exposure Analysis



- KL/RKL divergence in KD causes high exposure bias.
- Their asymmetry leads to weak student-teacher alignment.
- Symmetric JS divergence requires both teacher and student sampling.

Conclusion

Our LLMR:

- Is an unsupervised RL-KD approach outperforms other KD methods in text generation.
- Uses LLM for direct reward function in RL training.
- Demonstrates exposure bias mitigation with proof.

Thank you!

We thank all reviewers and chairs for their valuable comments. The research is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No. RGPIN2020-04465, an Alberta Innovates Project, the Amii Fellow Program, the Canada CIFAR AI Chair Program, a UAHJIC project, a donation from DeepMind, and the Digital Research Alliance of Canada (alliancecan.ca).