Training Language Models with Language Feedback at Scale

자연어 심화세미나

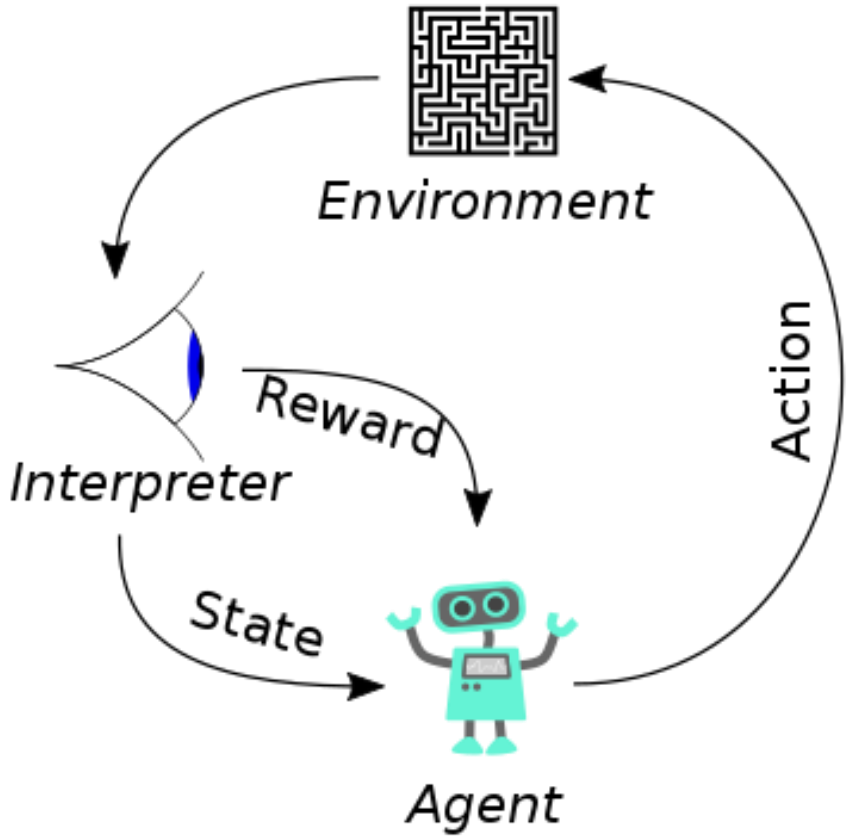
TOBIG'S 19기 최경주

Contents

Reinforcement Learning

What is RL?

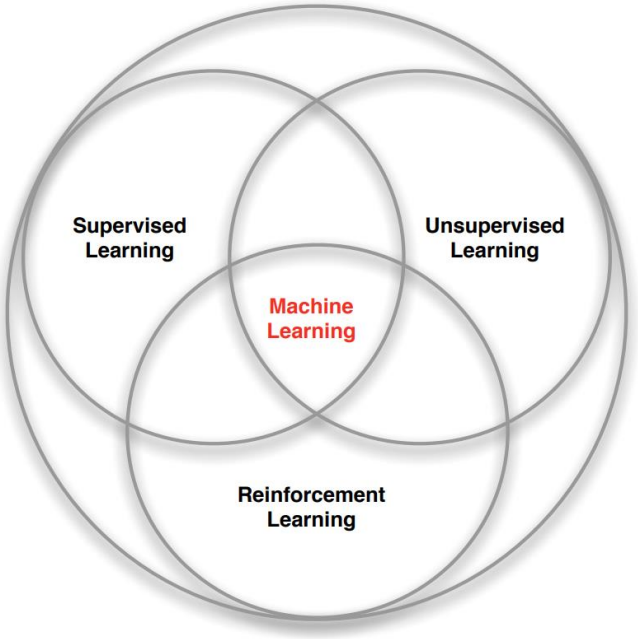
- 강화 학습이란?
  - 현재의 상태(State)에서 어떤 행동(Action)을 취하는 것이 최적인지를 학습하는 것이다.



https://upload.wikimedia.org/wikipedia/commons/thumb/1/1b/Reinforcement_learning_diagram.svg/375px-Reinforcement_learning_diagram.svg.png

Difference from other paradigms

- No supervisor, only a reward signal
- Feedback is delayed, not instantaneous
- Time really matters (sequential, non i.i.d data)
- Agent's actions affect the subsequent data it receives



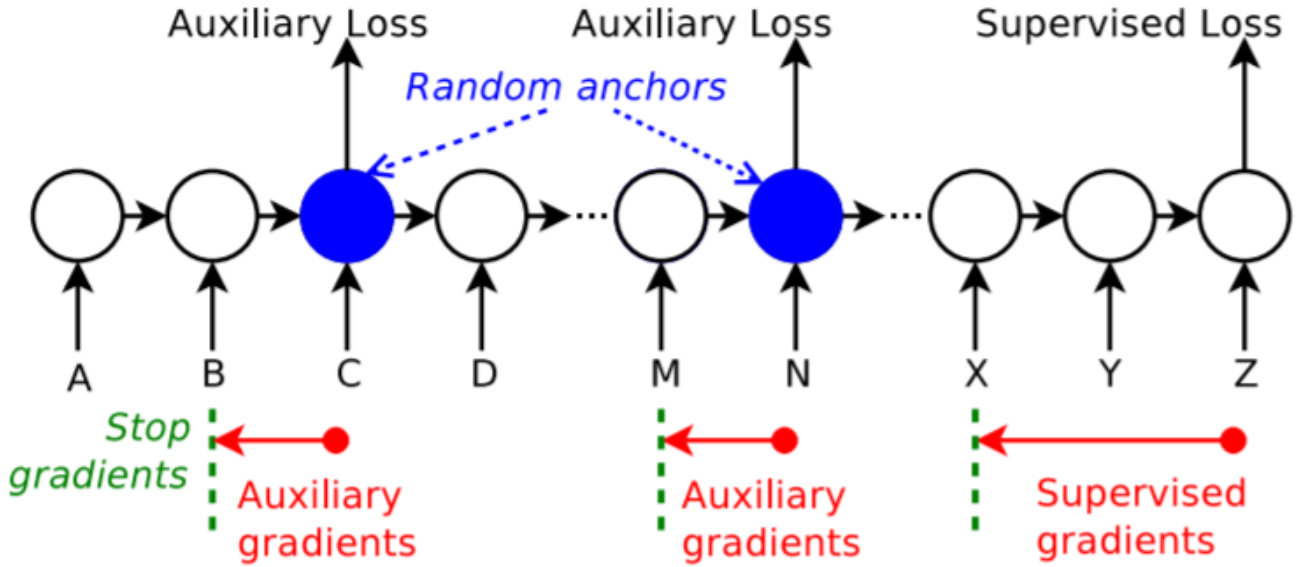
Language Models

Key limitation

- Generate violates human preferences
  - misinformation
  - offensive language
  - factually incorrect summaries
- To alleviate, generate text that scores highly according to human preference
  - Learn from human feedback -> problem: limited information

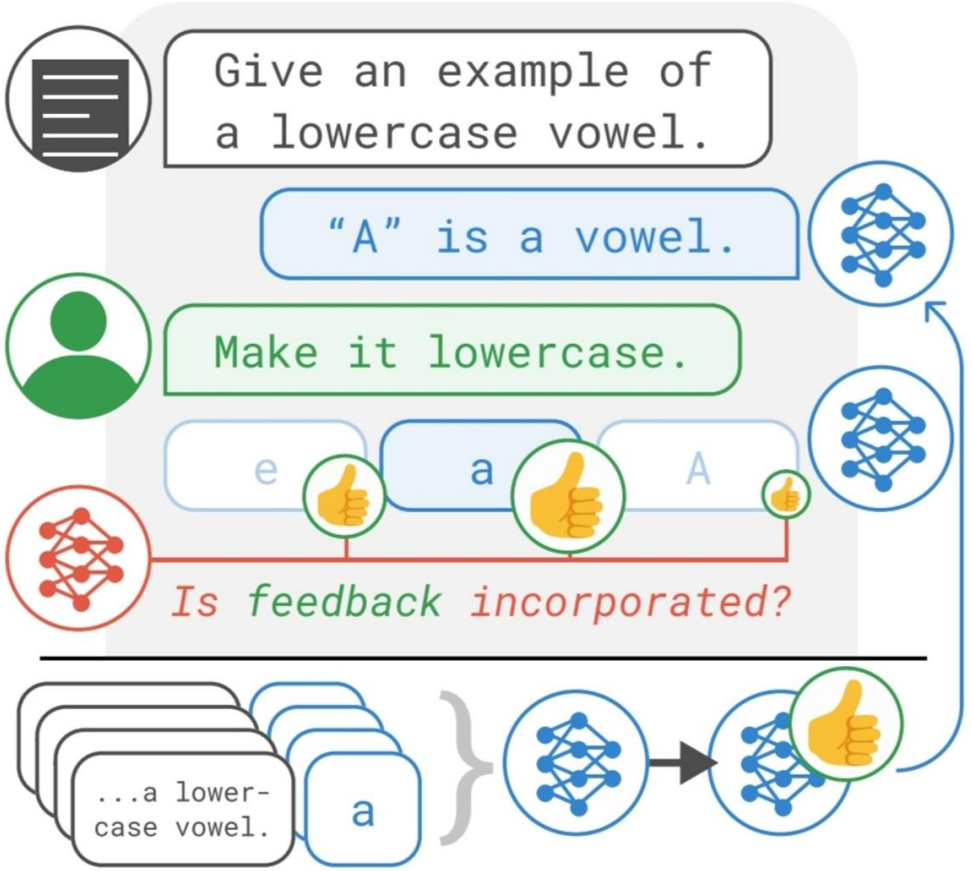
Auxiliary loss

- 학습을 잘 하기 위한 보조적인 loss
- Hidden state 중간에 loss를 적용하는 방식
- Not straightforwardly generalized



ILF(Imitation learning from Language Feedback)

- 1. Generate multiple refinements
- 2. Use instruction-finetuned LM to choose the refinement
- 3. finetune the LM that generated the initial output on the chosen refinement



Language feedback as Bayesian Inference

- Goal : **produce high-quality output for context**
- Use LM to generate output by conditioning on the context

- Data-generating process : $p_\theta(c, x_1, \mathcal{I}) = p(c)\pi_\theta(x_1|c)p(\mathcal{I}|c, x_1)$
- Output is high quality according to human preference when $\mathcal{I} = 1$

Language feedback as Bayesian Inference

- Goal : maximizing $\mathbb{E}_{c\sim p(c)} \log p(\mathcal{I}=1|c)$
- Approximate by $q(x_1|c)$ using ELBo
-

$$\log p(\mathcal{I}=1|c) = \log \sum_{x_1} p_\theta(x_1, \mathcal{I}=1|c)$$

$$\geq \sum_{x_1} q(x_1|c) \log \frac{p_\theta(x_1, \mathcal{I}=1|c)}{q(x_1|c)}$$

Language feedback as Bayesian Inference

- Using Expectation-Maximization procedure: alternating between maximizing E-step and M-step
- E-step : q에 관해 F를 최대화하는 것
- M-step : π_θ 에 관해 F를 최대화하는 것

⟶ Imitation learning from Language Feedback

Language feedback as Bayesian Inference

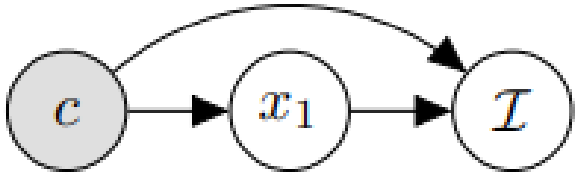
- E-step : Maximizing $F(\theta, q)$ w.r.t q
- High quality text에 더 높은 우도를 할당하기 위해 q를 개선하는 것과 동일

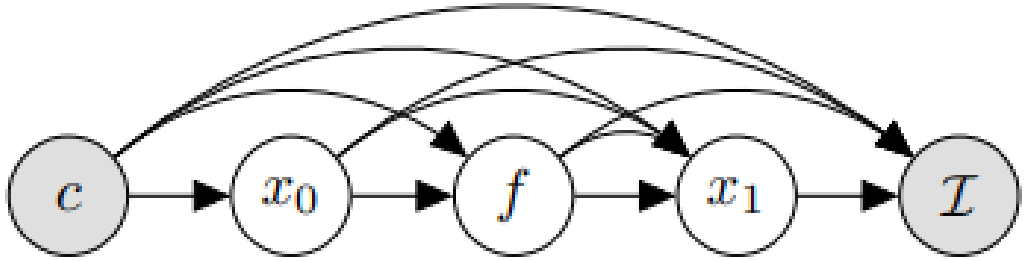
$$
\begin{aligned}
q(x_1|c) &= \sum_{x_0,f} p_\theta(x_0, f, x_1|\mathcal{I}=1, c) \\
&\propto \sum_{x_0,f} p_\theta(x_0, f, x_1|c) p_\theta(\mathcal{I}=1|c, x_0, f, x_1) \\
&= \sum_{x_0,f} p_\theta(x_0|c) p(f|c, x_0) p_\theta(x_1|c, x_0, f) \\
&\qquad p_\theta(\mathcal{I}=1|c, x_0, f, x_1).
\end{aligned}
$$

Language feedback as Bayesian Inference

- Model $p_\theta(\mathcal{I} = 1|c, x_0, f, x_1)$ as a Boltzmann distribution:

$$p_\theta(\mathcal{I} = 1|c, x_0, f, x_1) \propto \exp(R(c, x_0, f, x_1)/\beta)$$





Language feedback as Bayesian Inference

- M-step : Maximizing $F(\theta, q)$ w.r.t πθ
- Q에 의해 정의된 분포에서 cross-entropy loss를 최소화하는 하는 것과 동일

$$\begin{aligned}\operatorname*{argmax}_{\theta} F(\theta, q) &= \operatorname*{argmax}_{\theta} \mathbb{E}_{x_1 \sim q(x_1|c)} \log p_\theta(x_1, \mathcal{I} = 1|c) \\ &= \operatorname*{argmin}_{\theta} \mathbb{E}_{x_1 \sim q(x_1|c)} - \log \pi_\theta(x_1|c).\end{aligned}$$

Language feedback as Bayesian Inference

- Implement R by condition an instruction-finetuned LM on a binary question
- Prompt가 주어졌을 때 positive answer의 probability를 reward로 사용

$$p(y_{\text{good}}|\text{prompt}) = \frac{p(y_{\text{good}}|\text{prompt})}{p(y_{\text{good}}|\text{prompt}) + p(y_{\text{bad}}|\text{prompt})}$$

$$q(x_1|c) \propto \mathbb{E}_{x_0 \sim \pi_{\theta_{\text{old}}}(x_0|c)} \mathbb{E}_{f \sim p(f|c,x_0)} \pi_{\theta_{\text{old}}}(x_1|c, x_0, f) \exp(R(c, x_0, f, x_1)/\beta)$$

Language feedback as Bayesian Inference

- To obtain a sample $x_1 \sim q$
- sample N refinements $\{x_1^1, \ldots, x_1^N\} \sim \pi_{\theta_{\text{old}}}(x_1|c, x_0, f)$
- Compute $x_1 = \textbf{argmax}_{x_1^i} \exp R(c, x_0, f, x_1^i)$

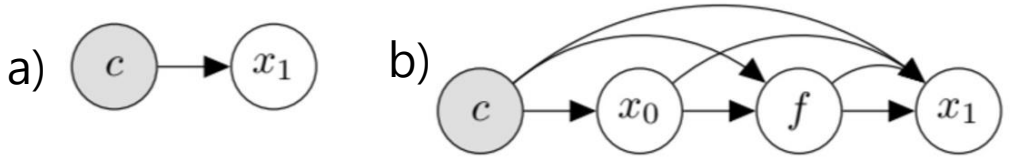
Difference other works

- Previous works use language feedback at test time to correct mistakes

- ILF uses feedback to train models
- This approach does not require human intervention at test time

ILF Algorithm

- a) graphical model of the target distribution : p(c|x1)
- b) graphical model of the proposal distribution : q(c|x1)



Algorithm 1 Imitation Learning from Language Feedback

Input: number of iterations K, a sequence of sets of source documents $\mathcal{C} = [\mathcal{C}_1, ..., \mathcal{C}_K]$, language model π_θ, refinement language model π_ψ, reward model R

for k **in** $1...K$ **do**

  Initialize finetuning dataset $\mathcal{D}_k = \{\}$

  for document c **in** $\mathcal{C}_k$ **do**

    $x_0 \sim \pi_\theta(x_0|c)$

    Human provides feedback f on (c, x_0)

    $\{x_1^1, \ldots, x_1^N\} \sim \pi_\psi(x_1|c, x_0, f)$

    $x_1 = \text{argmax}_{x_1^n} R(x_1^i|x_0, f, c)$

    Add (c, x_1) to $\mathcal{D}_k$

  end for

  Update π_θ by supervised finetuning on $\mathcal{D}_k$ (as in Eq. 4)

end for

Method

Method

- To generate improved output x1: high-quality summaries

- f : given language feedback
- x0 : initial model-generated output x0
- c : context(e.g., source document)
- $\pi\theta$: language model
- R : reward function

Method

- c is drawn from context distribution p(c)

- R로 측정된 output의 quality에 비례하는 ground truth distribution $p_c^*(x_1)$ 에 근접하도록 LM을 finetuning

$$\min_\theta \mathbb{E}_{c\sim p(c)}\mathrm{KL}(p_c^*, \pi_\theta),$$

$$\text{where } p_c^*(x_1) \propto \exp(\beta R(x_1|c))$$

$$\mathcal{L}(\theta) = -\mathbb{E}_{c\sim p(c)}\mathcal{L}_\theta(c),$$

$$\text{where } \mathcal{L}_\theta(c) = \sum_{x_1} p_c^*(x_1)\log \pi_\theta(x_1|c)$$

Method

- x1의 규모가 지수적으로 커지고 $p_c^*(x_1)$의 정규화 상수를 정확하게 계산하는 것이 불가능
- 같은 이유로 Loss function를 완벽하게 계산하지 못함
- 직접 샘플링을 할 수 있는 $q_c(x_1)$ 에서 중요도 샘플링을 사용

$$\mathcal{L}_\theta(c) = \sum_{x_1} q_c(x_1) \frac{p_c^*(x_1)}{q_c(x_1)} \log \pi_\theta(x_1|c)$$

Method

- 분산을 최소화하기 위해 $q_c(x_1)$을 $p_c^*(x_1)$에 가깝게 설계해야 함
- 1. c에서 LM으로부터 x0를 선택
- 2. 사람이 x0을 평가하고 (c, x0) pair에 대한 language feedback f를 제공
- 3. refinement LM이 (c, x0, f) pair에 대해 refine된 output x1을 생성

$$q_c(x_1) = \sum_{f,x_0} \pi_\psi(x_1|x_0, f)p(f|x_0)\pi_\theta(x_0|c)$$

Method

- N개의 summaries $x_1^i, \ldots, x_1^N$ 을 $q_c(x_1)$ 에서 샘플링
- ω i : importance weight (not computable) -> assume that $q_c(x_1^i)$ is constant

$$\mathcal{L}_\theta(c) \approx \sum_{i=1}^{N} \underbrace{\frac{p_c^*(x_1^i)}{q_c(x_1^i)}}_{=\omega^i} \log \pi_\theta(x_1^i|c)$$

Experiment

Experiment

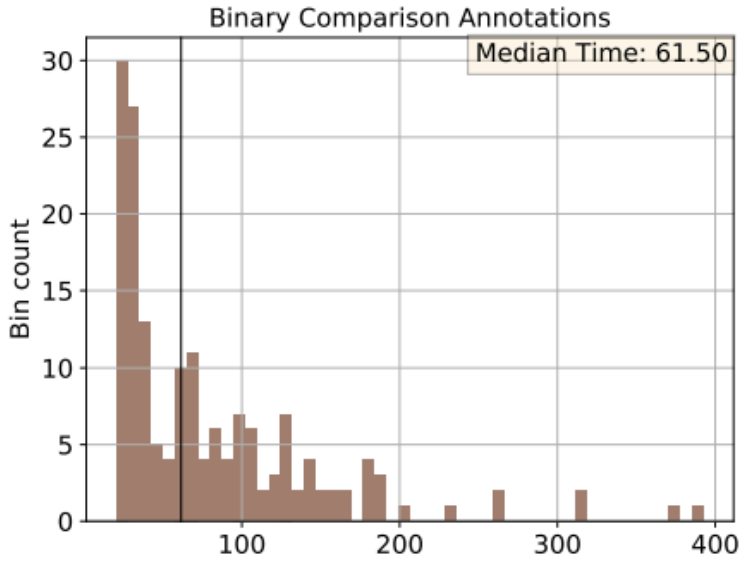
- Set up – choose model & size
  - Generate one output per sample with greedy decoding
  - Use differently-sized GPT-3 models

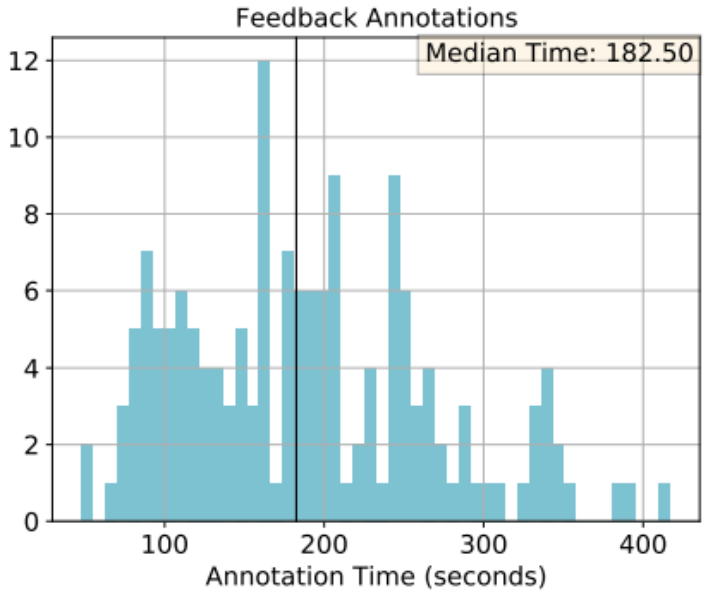
Models	Ada (-)	Babbage (1B)	Curie (6.7B)	Davinci (175B)
GPT-3	1.2 ± 0.3	1.7 ± 0.4	8.2 ± 0.7	38.5 ± 1.3
FeedME	1.6 ± 0.3	2.2 ± 0.4	6.0 ± 0.6	35.8 ± 1.3

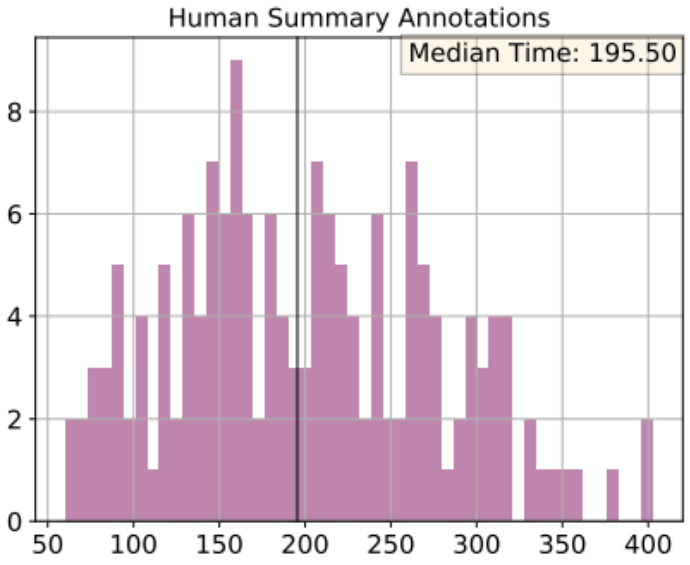
  - Take 175B parameter models & FeedME(instruction-finetuned model)

Experiment

- Text summarization on ILF
  - Using TL;DR dataset
  - language feedback by Surge AI







Experiment

- Comparing Refinement Ranking Methods
  - Condition FeedMe on the initial summaries of train dataset
  - Scoring Refinements with InstructRM
  - Five different prompts & select refinement with the highest average p(y|prompt)

	Scoring Function	Win Rate in % vs. Random Selection
Task Specific Heuristic	Max Length	65.0 ± 2.7
Zero-Shot	Embedding Similarity	48.3 ± 3.0
	InstructRM Prompt 1	55.0 ± 3.0
	InstructRM Prompt 2	58.0 ± 2.9
	InstructRM Prompt 3	56.5 ± 2.9
	InstructRM Prompt 4	55.8 ± 2.8
	InstructRM Prompt 5	50.0 ± 3.0
	InstructRM Ensemble	$\mathbf{56.0 \pm 3.0}$

Reward Model

- Standard RM
- LM의 마지막 embedding layer를 제거하고 scalar value를 출력하도록 학습

- Reward Model with Language Output
- Scalar 대신 언어 토큰을 출력하도록 학습
- 선택된 데이터셋을 사용하여 LM을 finetune, $\lambda \log p(x_0) + \log p(y|x_0)$를 최대화

$$\mathcal{L}(p_\theta, x, y) = -\lambda \cdot \sum_{t=1}^{|x|} \log p_\theta(x_t|x_{<t}) - \sum_{t=1}^{|y|} \log p_\theta(y_t|x, y_{<t})$$

Reward Model

- Evaluate various RMs

	Models	# Params	Train Data Size	Development Accuracy (in %)	Validation Accuracy (in %)
LM Loss / Our dataset	OPT Comparison	13B	5K	66.5 ± 3.3	72.6 ± 1.9
	OPT RM	1.3B	5K	70.0 ± 3.2	69.6 ± 2.0
	OPT RM	13B	100	54.5 ± 3.5	53.4 ± 2.2
	OPT RM	13B	1K	68.5 ± 3.2	67.2 ± 2.1
	OPT RM	**13B**	**5K**	$\mathbf{69.5 \pm 3.2}$	$\mathbf{73.4 \pm 1.9}$
	GPT-3 Comparison	-	5K	68.0	71.2 ± 2.0
	GPT-3 Binary	-	**5K**	-	$\mathbf{74.2 \pm 2.0}$
RM Loss / Our dataset	OPT	13B	5K	68.5 ± 3.2	71.8 ± 2.0
RM Loss / Stiennon et al. (2020) train dataset	Stiennon et al. (2020) RM	1.3B	64K	58.0 ± 3.4	63.8 ± 2.1
LM Loss / Stiennon et al. (2020) train dataset	OPT Binary	13B	90K	69.0 ± 3.2	68.6 ± 2.0

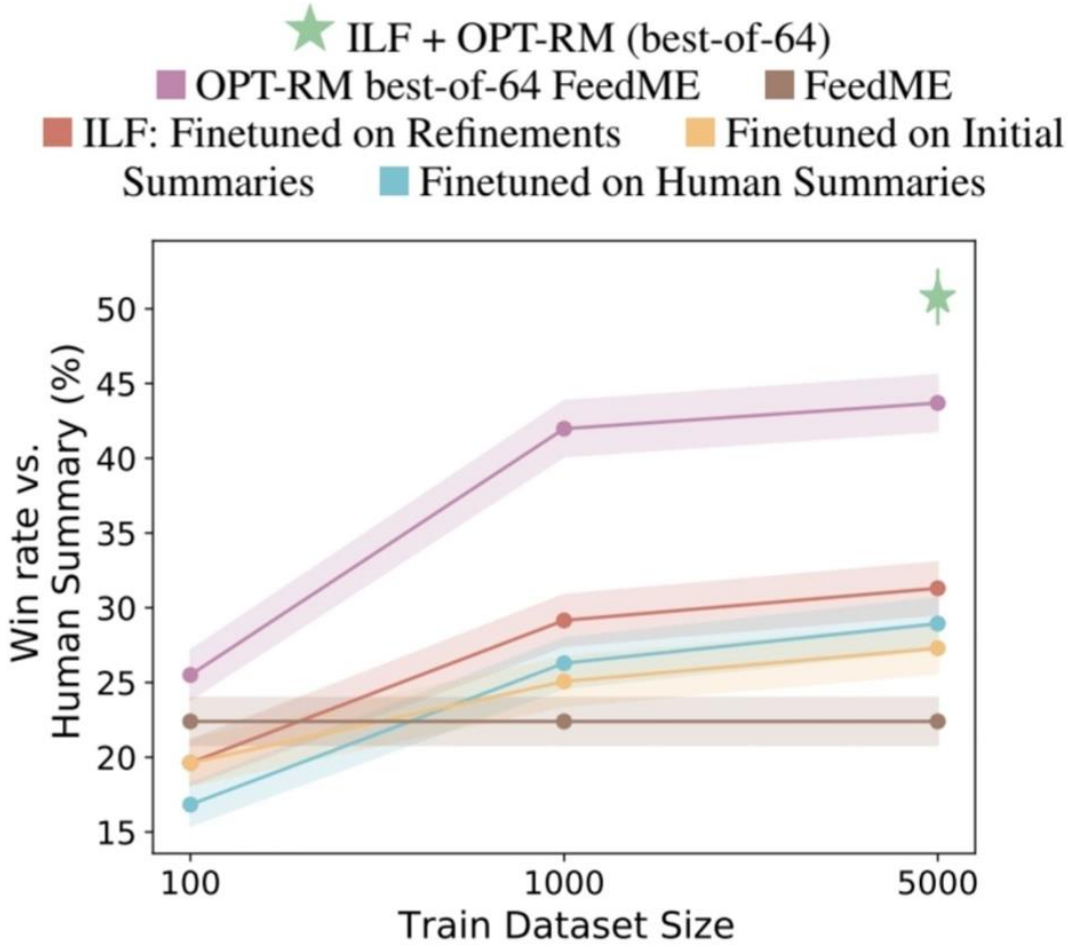
Reward Model

- Compare reward models and ranking methods

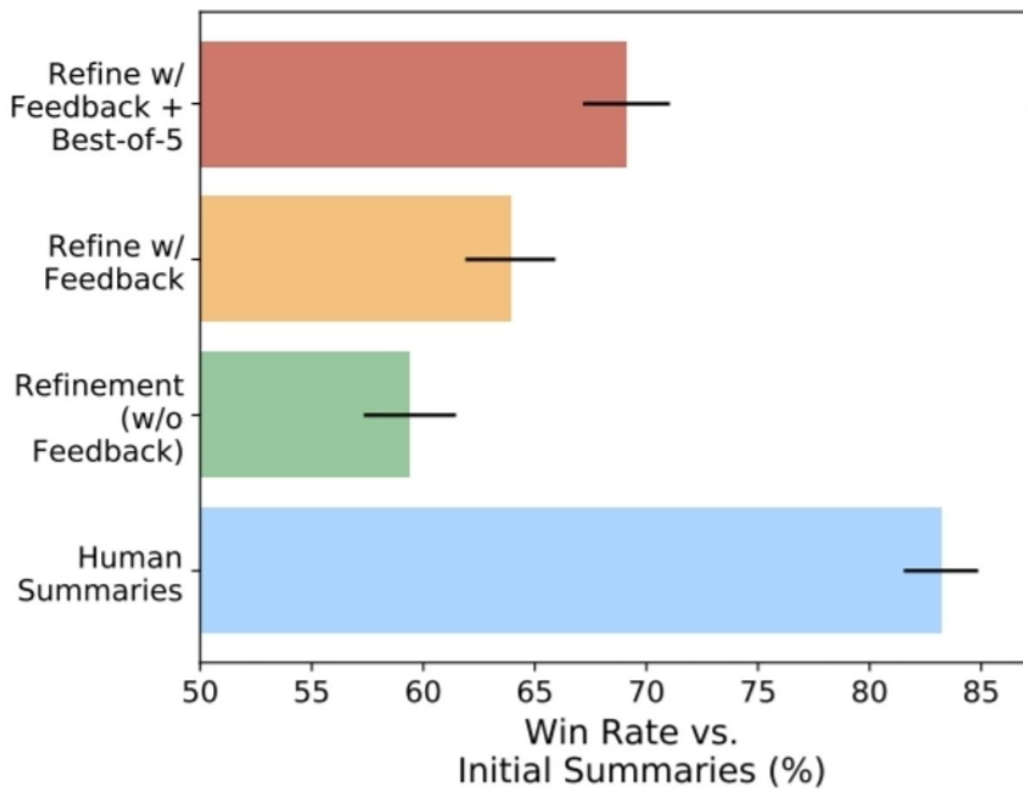
	Scoring Function	Win Rate vs Random Selection (in %)
Task Specific Heuristic	Max Length	65.0 ± 2.7
Zero-Shot	Embedding Similarity	48.3 ± 3.0
	InstructRM Ensemble	$\mathbf{56.0 \pm 3.0}$
Finetuning on 5K samples	**OPT Binary**	$\mathbf{63.3 \pm 2.7}$
	GPT-3 Binary	61.8 ± 2.9

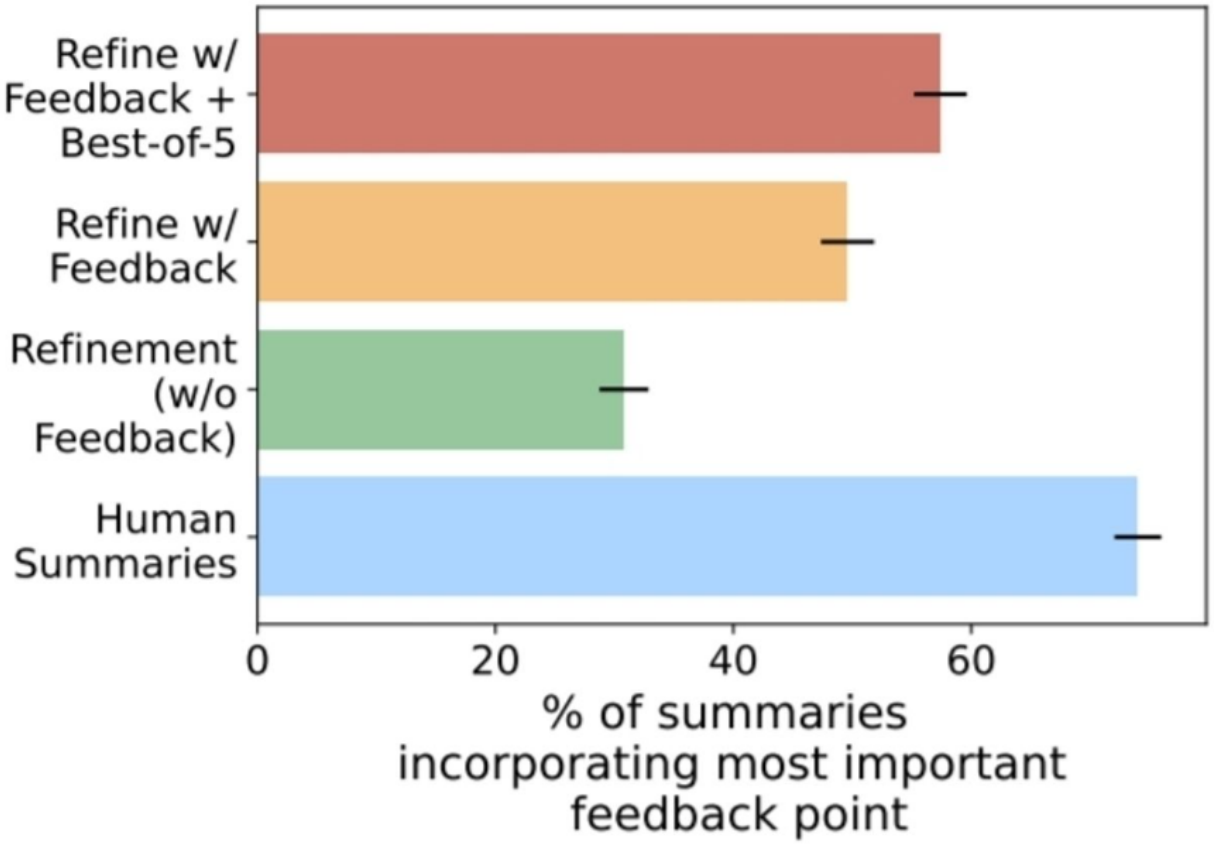
Experiment

- Comparing Feedback Learning Algorithms
  - Single iteration of ILF
  - Human-written summaries
  - Initial summaries
  - Best-of-N
  - ILF + Best-of-N

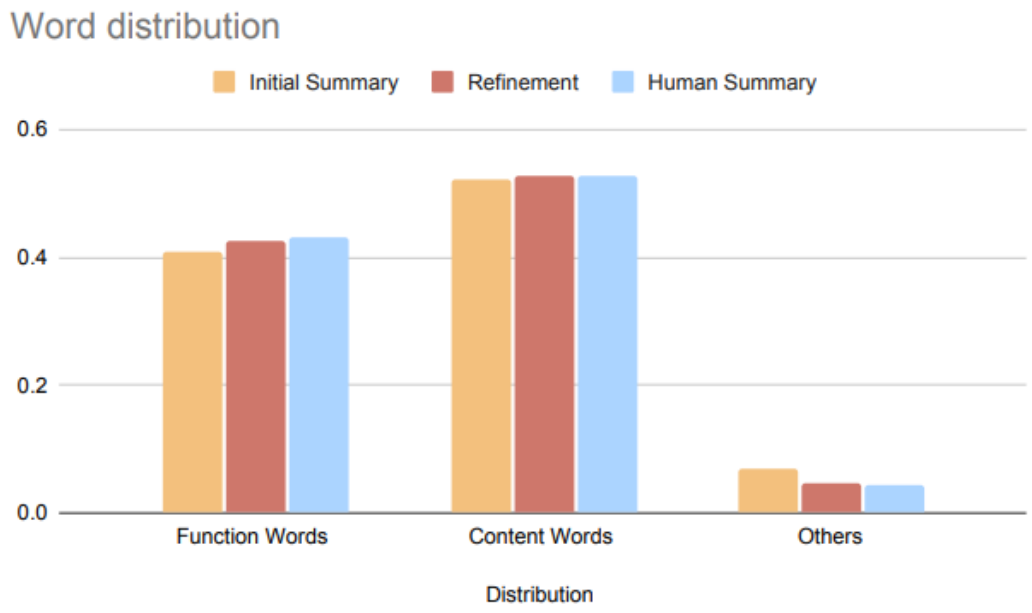


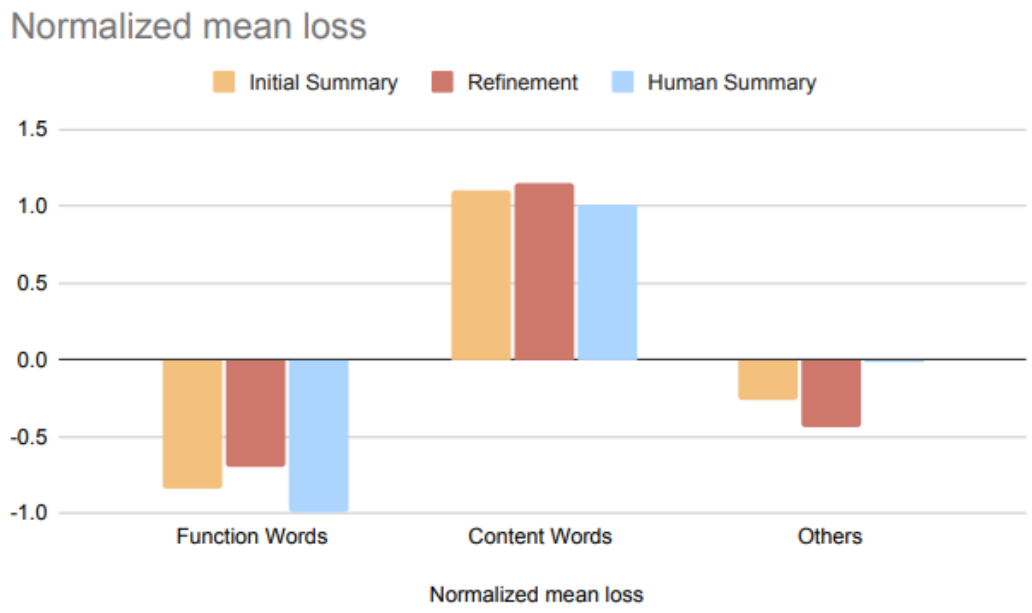
Ablation Study





Distribution of tokens of various finetuning datasets





Prompts

- Summarization Prompts

Methods	Format
INITIAL SUMMARY	Write an excellent summary of the given text. Title: {title} Text: {text} TL;DR:
REFINEMENT WITH FEEDBACK	Write an excellent summary that incorporates the feedback on the given summary and is better than the given summary. Title: {title} Text: {text} Summary: {summary} Feedback on Summary: {feedback} Improved TL;DR:
REFINEMENT WITHOUT FEEDBACK	Write an excellent summary that is better than the given summary. Title: {title} Text: {text} Summary: {summary} Improved TL;DR:

Prompts

- InstructRM Prompts
- Example: Prompt2

PROMPT 2	Post title: {title} Original post: {text} Original summary: {summary} Feedback: {feedback} New summary: {refinement} Question: Does the new summary incorporate the feedback provided? Answer Yes or No. Answer:

Prompts

- InstructRM Prompts
- Example: Prompt4

PROMPT 4	
	Here's a summary of a Reddit post, feedback on the summary, and a new summary. You will be asked to determine whether the new summary incorporates the feedback provided. A good summary is a short piece of text that has the essence of the original text. A good summary tries to accomplish the same purpose and conveys the same information as the original text. Remember, you will be asked to determine whether the new summary incorporates the feedback provided. Post title: {title} Below, there's the content of the post that was summarized. Original Post: {text} Remember, you will be asked to determine whether the new summary incorporates the feedback provided. Here's the original summary. Original summary: {summary} Remember, you will be asked to determine whether the new summary incorporates the feedback provided. A human then provided feedback on the above summary. Feedback: {feedback} Based on this feedback, a new summary was written. New summary: {refinement} Does this new summary incorporate the feedback provided? Answer Yes or No. Answer:

Prompts

- Finetuning Prompts

FINETUNING ON FEEDBACK + REFINEMENTS	Write an excellent summary that incorporates the feedback on the given summary and is better than the given summary. Title: {title} Text: {post} Summary: {summary} Feedback on summary:	{feedback} Improved TL;DR: {refinement} ###

Prompts

- Reward Model Prompts

Reward Model Type	Prompt	Completion
BINARY RM	Title: {title} Text: {post} TL;DR: {summary_A/summary_B} Question: Is the above an excellent summary of the given text? An excellent summary is coherent, accurate, concise, and detailed. Answer with Yes or No. Answer:	{" Yes"/" No"}
COMPARISON RM	Title: {title} Text: {post} Summary A: {summary_A} Summary B: {summary_B} Question: Which summary is the better one? An excellent summary is coherent, accurate, concise, and detailed. Answer with A or B. Answer:	{" A"/" B"}

Discussion

Reference

- '강화 학습이란', https://ko.wikipedia.org/wiki/%EA%B0%95%ED%99%94_%ED%95%99%EC%8A%B5
- Training Language Models with Language Feedback at Scale. 2023, https://arxiv.org/abs/2303.16755
- Auxiliary Loss, https://seqml.github.io/a2ls/