동계 세미나 (2/19, Thu)

LLM Generalization

구선민

Natural Language Processing & Artificial Intelligence

고려대학교 KOREA UNIVERSITY

UNDERSTANDING THE EFFECTS OF RLHF ON LLM GENERALISATION AND DIVERSITY

Robert Kirk* [α] **Ishita Mediratta** [β] **Christoforos Nalmpantis** [β] **Jelena Luketina** [γ]
Eric Hambro [β] **Edward Grefenstette** [α] **Roberta Raileanu** [β]

[α] University College London, [β] Meta, [γ] University of Oxford

Motivation

*** RLHF의 효율성**

- LLM 성능이 향상됨에 따라 해결해야 할 태스크는 더욱 복잡해짐
  . 복잡한 태스크에서는 demonstration을 만드는 것이 어렵고 비용이 큼
  . 성능 평가 역시 명확한 정답 기반 평가가 어려움

- 대신 모델 출력을 사람이 평가하거나 순위를 매기는 방식이 더 효율적

Motivation

*** RLHF의 각 단계에 대한 영향 분석 부족**

- 현재 RLHF 파이프라인의 각 구성 요소가 최종 모델의 동작에 어떻게 기여하는지에 대한 이해가 부족

- 특히, 'Out-of-Distribution (OOD) generalization'와 'output diversity' 영역에서 영향 탐구 부족
  . OOD 일반화: 모델이 훈련 데이터의 분포를 넘어서는 다양한 실제 시나리오에서 성능 위해 중요
  . 출력 다양성: 모델이 다양한 출력을 생성하는 능력. creative or open-ended domains 에 필수적

- 기존 일부 연구에서는 RLHF로 인한 다양성 감소를 보여주었음
  . 그러나 체계적인 분석 X
  . BLEU 등 단순 token-level metric 사용 + 제한된 use case만 분석

Research Goal

*** RLHF 파이프라인 각 단계에 대한 영향 분석**

- RLHF pipeline의 각 단계가 다음에 미치는 영향을 체계적으로 분석
  . In-distribution performance
  . Out-of-distribution performance
  . Output diversity

- 분석 대상: SFT, RLHF, Best-of-N sampling

- RQ: RLHF pipeline의 각 단계는
  . Generalization에 어떤 영향을 미치는가?
  . Output diversity에 어떤 영향을 미치는가?
  . 성능 향상과 diversity 사이에 tradeoff가 존재하는가?

Dataset and Tasks

* Generalization

- Dataset: TL;DR dataset의 필터링된 버전 사용
  . 약 120,000개의 Reddit 게시물 및 요약으로 구성

- RM 학습 데이터: Stiennon et al. (2022)이 수집한 선호도 데이터 활용
  . 약 64,000개의 요약 비교로 이루어져 있으며, human annotator 가 정한 기준에 따라 선호되는 요약 라벨링

* Instruction Following task

- Dubois et al. (2023)의 AlpacaFarm에서 공개한 SFT, RLHF 및 RM 모델을 사용

Model Evaluation

* Generalization evaluation

- GPT-4를 human evaluator proxy로 사용

- Preference vs. Reference (PvR)
  . GPT-4에게 model output vs reference output (ground truth) 제시한 후,
  어떤 출력이 더 나은지 결정하도록 하여 측정
- Head-to-head Comparisons: 두 모델의 출력을 GPT-4에 제시하여 어느 쪽이 더 나은지 결정하도록 함

- OOD 일반화 평가
  . 요약 태스크: TL;DR dataset (train) → CNN/DailyMail (OOD test)
  . IF 태스크: AlpacaFarm (train) → AlpacaEval, Sequential Instructions (OOD test)

Model Evaluation

* Diversity evaluation

- 출력 다양성을 평가하기 위해, 이전 연구에서 많이 사용하는 여러 다양성 측정 방법을 사용
- distinct N-grams, Sentence-BERT embedding cosine similarity, NLI diversity

Experimental Results (GENERALISATION)

* Summarization

- BoN > RLHF > SFT 순서 성능
- Generalization Gap (ID – OOD) 는 큰 차이 없음
- Temperature 변화에도 동일한 경향 보임

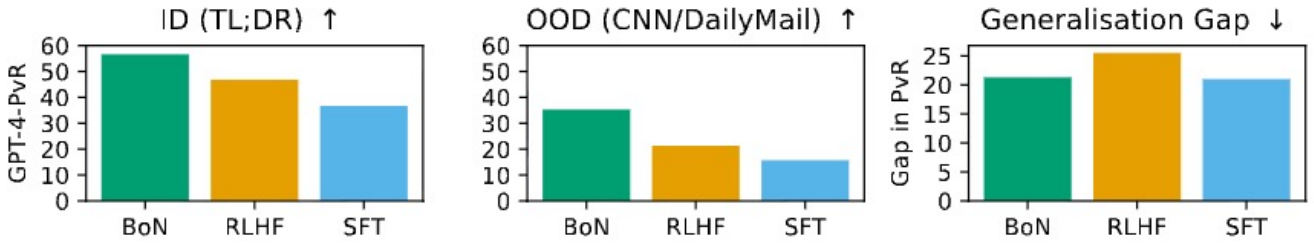


Figure 1: **Summarisation Generalisation Results.** GPT-4-PvR for SFT, BoN and RL policies, based on LLaMa 7B, trained on the summarisation task. In-distribution is performance on TL;DR, and out-of-distribution is on CNN/DailyMail, and generalisation gap is ID – OOD performance.

Experimental Results (GENERALISATION)

* Instruction Following

- AlpacaEval(easy OOD)에서는 모든 모델이 비슷하게 잘 일반화되지만, Sequential Instructions (hard OOD) 태스크에는 RLHF가 훨씬 더 잘 일반화
  . RLHF가 더 큰 분포 이동에 대해 SFT에 비해 더 잘 일반화될 수 있음을 시사

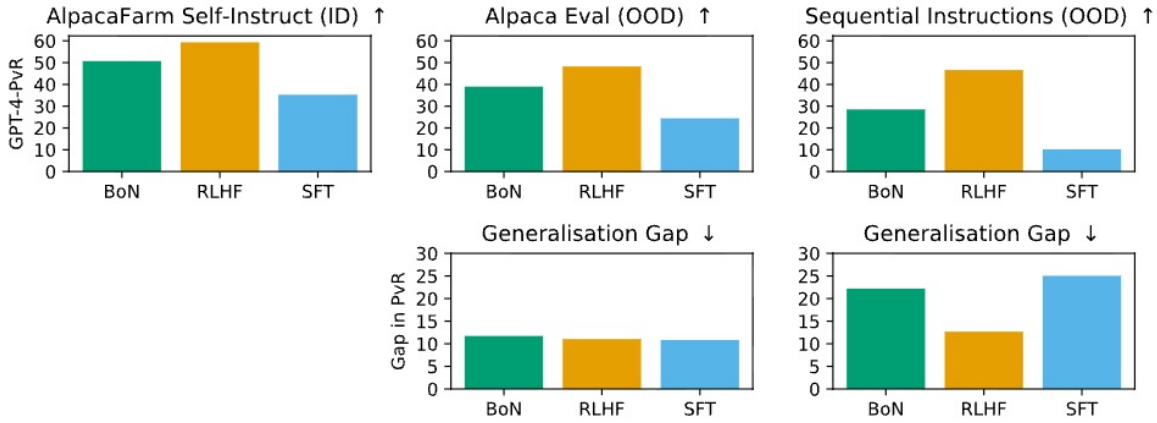


Figure 2: **Instruction Following Generalisation Results.** GPT-4 PvR for SFT, BoN and RL policies, based on LLaMa 7B, trained on the AlpacaFarm Self-Instruct instruction following task. ID is on AlpacaFarm Self-Instruct, OOD is on the AlpacaEval and Sequential Instructions datasets respectively, and generalisation gap is ID – OOD performance.

Experimental Results (DIVERSITY)

* Per-input diversity

- 동일한 입력에 대해 여러 번 샘플링했을 때의 다양성
- RLHF는 SFT에 비해 출력 다양성을 크게 감소

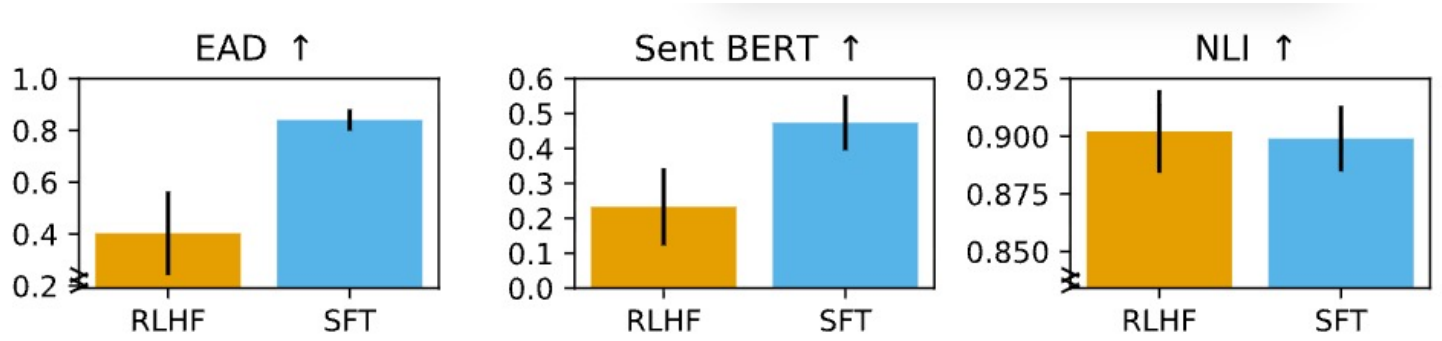


Figure 4: **Per-input diversity metrics for RLHF and SFT models**. For these scores the outputs used to calculate the diversity are a sample of outputs from the model for single input. These per-input scores are then averaged, as in Eq. (2). Error bars are standard deviation of the per-input diversity score across different inputs. Note that some plots have broken y-axis for better visualisation.

Experimental Results (DIVERSITY)

* Across-input diversity

- 다양한 입력에 대해 얼마나 다양한 스타일, 내용, 어휘를 가진 출력을 생성하는지
- RLHF 가 특정 스타일의 텍스트를 출력하도록 편향되는 "mode collapse" 현상 보임
  → 일반화 가능성과 다양성 사이의 trade-off

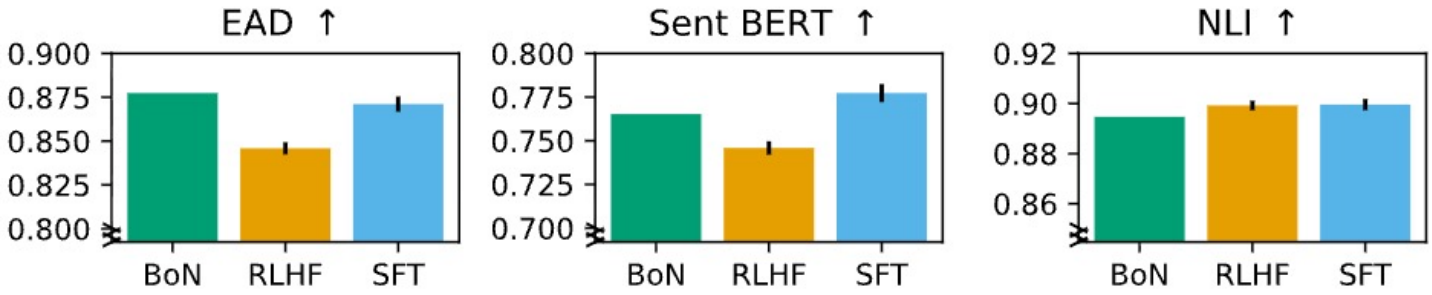


Figure 5: **Across-input diversity metrics for RLHF, BoN and SFT models**. For these scores the outputs used to calculate the diversity are a set of single outputs from a range of inputs, as in Eq. (3). Note that all plots have broken y-axis for better visualisation; the differences between SFT and RLHF are much smaller in this case than in the per-input diversity metrics in Fig. 4. Error bars (where present) are standard deviation of the across-input scores over different samples from the set of outputs for each input.

PAFT: Prompt-Agnostic Fine-Tuning

Chenxing Wei[†§], **Mingwen Ou**[°], **Ying He**[#†], **Yao Shu**[#≀], **Fei Yu**[‡]
[†]College of Computer Science and Software Engineering, Shenzhen University, China
[°]Tsinghua Shenzhen International Graduate School, Tsinghua University, China
[§]Guangdong Lab of AI and Digital Economy (SZ), China
[≀]Hong Kong University of Science and Technology (Guangzhou), China
[‡]School of Information Technology, Carleton University, Canada
weichenxing2023@email.szu.edu.cn, yaoshu@hkust-gz.edu.cn

Motivation

*** 프롬프트 강건성 부족**

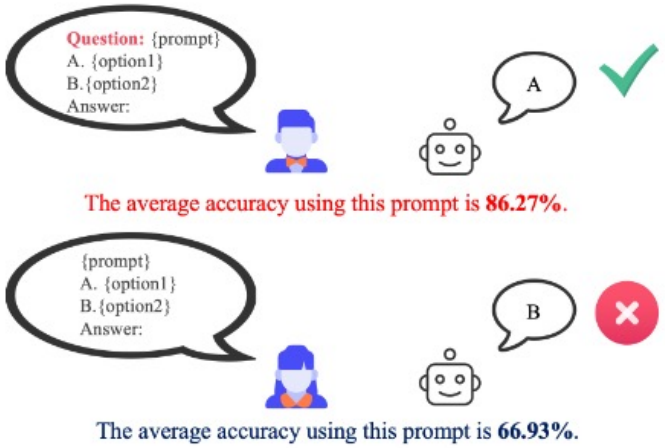
- 기존 방법은 fixed instruction prompt 기반 학습을 사용
- 이로 인해 모델이 task 자체가 아니라 특정 prompt 표현에 과적합

→ 결과적으로,
- 동일한 의미라도 프롬프트 표현이 조금만 바뀌어도 성능 급격히 저하

- QA task에서 prompt wording 변화만으로 정확도 크게 감소

- 챗봇 및 agent가 학습 시 보지 못한 instruction에 대해 취약한 성능 보임

- 심한 경우 random guessing 수준의 성능까지 하락 가능



Motivation

* Research Gap

- 기존 fine-tuning task semantics 이 아닌 prompt surface pattern 학습
  → 모델이 task의 본질이 아니라 특정 instruction phrasing에 의존하게 됨

- 기존 연구는 주로 prompt engineering, prompt tuning, in-context learning 에 집중
  → Fine-tuning 단계에서 prompt robustness를 직접 개선하는 연구는 부족

→ 학습 과정에서 다양한 prompt를 동적으로 사용하는 PAFT 제안

Contribution

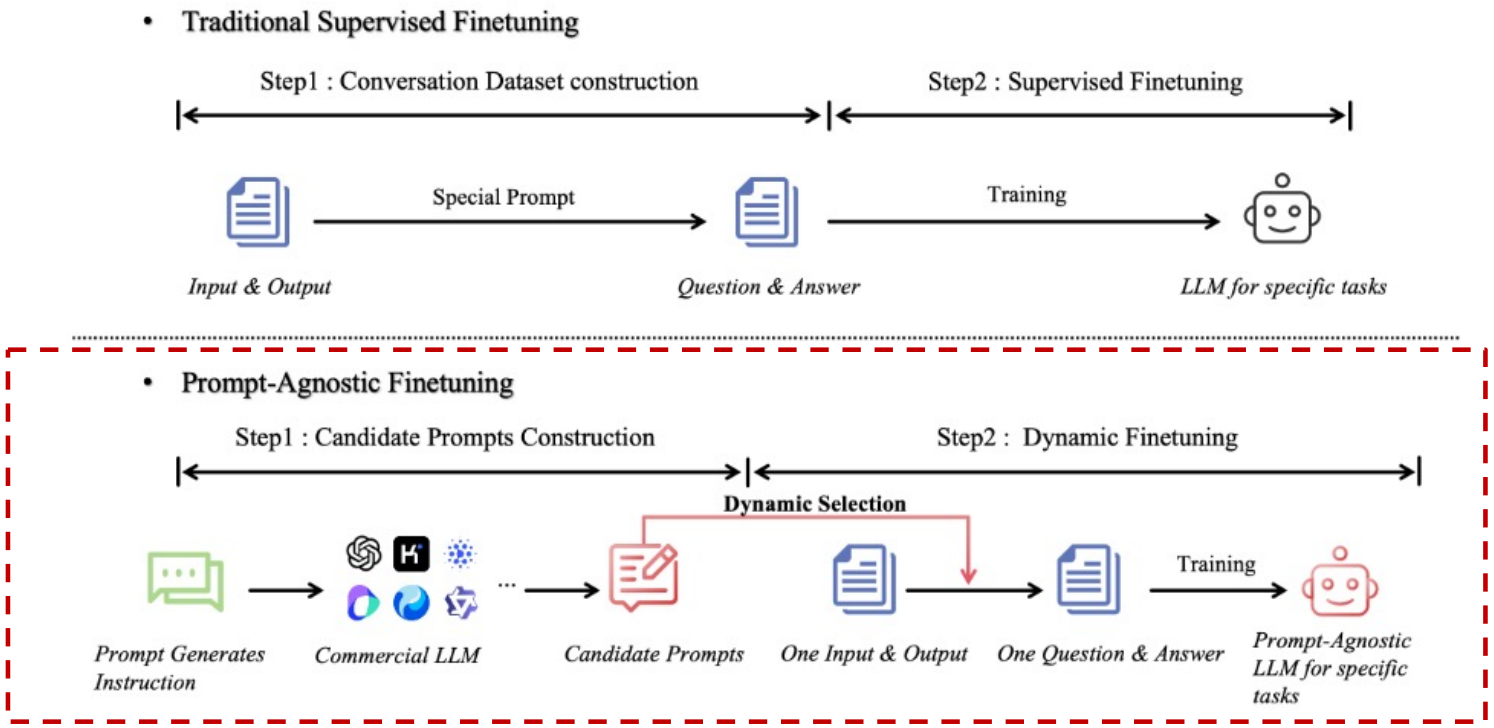
1. fine-tuning with fixed prompts 는 unseen prompts 에 대한 일반화 성능 하락 및 성능 저하된다는 점 강조

2. fine-tuned model 프롬프트 강건성을 향상시키기 위해 후보 프롬프트 구축과 다이나믹 파인튜닝을 통합한 새로운 프레임워크인 PAFT 제안

3. 다양한 다운스트림 태스크, 파인튜닝 알고리즘, unseen 프롬프트를 포함한 다양한 테스트 프롬프트 전반에 걸쳐 PAFT의 일관되고 강건한 성능을 경험적으로 입증

Overview

* PAFT



Preliminaries

*** 프롬프트 견고성 평가**

- 프롬프트의 형식은 태스크 유형에 관계없이 모델 성능에 많은 영향을 미치며, 프롬프트의 10%만이 거의 최적의 결과를 생성

- 사소한 프롬프트 수정(예: 재구성, 구두점, 순서 변경)은 상당한 변동을 유발

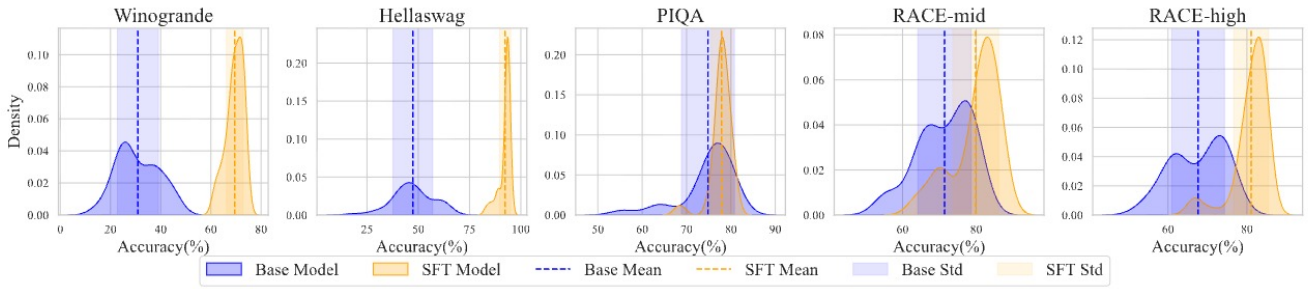


Figure 3: This figure presents experimental results across four datasets comparing base and SFT model performance on 450 diverse prompts (both human-written and LLM-generated). Probability distribution plots reveal that despite SFT's overall accuracy improvements, substantial performance variability persists—certain prompts yield markedly lower accuracy, with high standard deviations indicating significant prompt-dependent fluctuations. These findings underscore crucial impact of prompt and demonstrate the necessity for prompt-agnostic fine-tuning approaches.

Candidate Prompt Construction

* Diverse LLM Ensemble

- pre-training data, architectures, and optimization objectives 의 차이에서 는 task interpretation 의 고유한 변동성을 포착하기 위해 다양한 생성 능력을 가진 10 개의 LLM 사용

- 언어 스타일 및 인스트럭션 접근 방식 전반에 걸쳐 프롬프트 형식의 포괄적인 커버리지를 보장하여 단일 모델 생성 편향을 효과적으로 완화

Candidate Prompt Construction

* Dual Prompting Strategy

- 품질과 다양성의 균형을 맞추기 위해 few-shot 및 zero-shot 기법 결합

- 각각 20개의 프롬프트를 생성함으로써, 고품질의 다양한 형식으로 구성된 포괄적인 세트를 만들고, 모델을 현실적인 프롬프트 품질 분포에 노출시키며, 실제 시나리오에 대한 견고성을 향상

Candidate Prompt Construction

* Rigorous Evaluation Design

- 생성된 프롬프트를 랜덤으로 학습 및 테스트셋(8:1 비율)로 분할하여 각 세트에서 완전히 구별되는 프롬프트 보장

- 훈련 중에 모델을 다양한 프롬프트 스타일에 노출시키는 동시에 새로운 형식에 대한 일반화 능력 평가의 강건성

- unseen 프롬프트를 사용하여 평가함으로써, 성능 향상이 특정 패턴에 과적합 여부 평가

Dynamic Fine-Tuning

* Dynamic Fine-Tuning Algorithm

- P: Candidate Prompt Training Set
- D: Task-Specific Dataset
- T: Number of Training Epochs
- K: Number of Same Prompt Training
- θ_0^0: Initialized Trainable Parameters
- η_θ: Learning Rate

Algorithm 1 The PAFT Framework

Input: Generate a good candidate prompt training set $\mathbb{P}$; A task-specific dataset $\mathbb{D}$; The number of training epochs T; The number of same prompt training K; Initialized trainable parameters θ_0^0; Learning rate η_θ
Output: Fine-tuned model parameters θ^*.
for each epoch $t = 0$ **to** $T - 1$ **do**
  $p \leftarrow$ RandomlySample($\mathbb{P}$) // Randomly select a prompt from the candidate set
  $k \leftarrow 0$ // Initialize the step counter
  for each data point $(x, y) \in \mathbb{D}$ **do**
    $\mathbf{I} \leftarrow$ InputConstruction(x, p) // Construct input using prompt p and data x
    $\theta_t^{k+1} \leftarrow \theta_t^k - \eta_\theta \nabla_\theta \ell(\theta, \mathbf{I})|_{\theta=\theta_t^k}$
    $k \leftarrow k + 1$ // Increment the step counter
    if $k \mod K == 0$ **then**
      $p \leftarrow$ RandomlySample($\mathbb{P}$)
    end if
  end for
  $\theta_{t+1}^0 \leftarrow \theta_t^k$
end for
return $\theta^* = \theta_T^0$

Experimental Setup

*** Datasets**

- Reasoning 및 Language Understanding
  . HellaSwag: 상식 기반 추론 능력 평가
  . Winogrande: 문맥 기반 언어 이해 능력 평가
  . RACE: 독해 및 논리적 추론 능력 평가
  . PIQA: 물리적 상식 기반 문제 해결 능력 평가

- Specialized Capability Evaluation
  . HumanEval: 코드 생성 능력 평가
  . T-Eval: Tool 사용 능력 평가
  . XStory-Cloze: 대화 및 multilingual reasoning 평가

- Mathematical Reasoning
  . GSM8K: 수학 문제 해결 능력 평가
  . Geometry3k: 복합적 수학 추론 능력 평가

Experimental Results

* Prompt Robustness

Methods	**Hellaswag**			**PIQA**			**Winogrande**			**RACE-mid**			**RACE-high**			**Average**		
Metric	Mean	Std	Top	Mean	Std	Top	Mean	Std	Top	Mean	Std	Top	Mean	Std	Top	Mean	Std	Top
Base Model	47.36	±9.78	0%	74.68	±6.24	0%	45.15	±11.78	0%	71.39	±7.33	0%	67.62	±6.78	0%	61.24	±8.38	0%
User	92.35	±2.78	0%	77.87	±2.36	0%	78.16	±7.97	0%	79.88	±6.32	22%	81.05	±4.45	4%	81.86	±4.78	5%
TopAccuracy	91.27	±2.79	86%	75.96	±3.89	0%	66.77	±3.94	0%	84.81	±4.06	59%	82.45	±3.26	14%	80.25	±3.63	32%
BATprompt	90.30	±1.79	78%	83.41	±1.74	16%	69.01	±4.45	0%	83.92	±5.38	65%	81.33	±4.21	12%	81.56	±3.51	34%
ZOPO	92.46	±2.43	86%	83.52	±2.23	27%	74.75	±3.81	0%	83.50	±5.05	51%	82.36	±4.53	35%	83.32	±3.61	40%
PAFT	**93.83**	**±0.70**	**100%**	**89.33**	**±0.63**	**100%**	**82.09**	**±0.81**	**100%**	**87.26**	**±2.23**	**94%**	**85.17**	**±1.71**	**73%**	**87.57**	**±1.57**	**94%**
↪ Improv.	+1.37	-1.09	14%	+5.81	-1.11	73%	+3.93	-3.00	100%	+2.45	-1.83	29%	+2.72	-1.55	38%	+4.25	-1.94	54%

Method	**HumanEval**	**Xstory_cloze**	**Geometry3k**	**T-Eval**	**GSM8K**
Base	41.31 (± 10.36)	48.23 (± 8.36)	32.17 (± 15.36)	58.97 (± 14.03)	74.36 (± 21.37)
SFT	49.63 (± 4.31)	54.77 (± 4.79)	37.94 (± 6.17)	70.37 (± 8.14)	81.47 (± 13.24)
PAFT	**54.24 (± 1.36)**	**60.27 (± 0.73)**	**40.19 (± 1.27)**	**73.17 (± 3.27)**	**85.71 (± 5.93)**
↪ Improv.	+4.61 (-2.95)	+5.50 (-4.06)	+2.25 (-4.90)	+2.80 (-4.87)	+4.24 (-7.31)

Experimental Results

* Inference Efficiency

Table 3: Comparison of inference time (in hours) for different fine-tuning methods. PAFT shows better inference efficiency than other methods. The last line shows the multiple of PAFT improvement.

Inference time/h	Hellaswag	PIQA	Winogrande	RACE	Average
Base Model	3.97	1.35	1.72	6.24	3.32
User	6.52	0.98	3.27	8.23	4.75
TopAccuracy	5.75	1.13	2.76	7.56	4.30
BATprompt	4.57	1.57	3.14	7.98	4.32
ZOPO	5.12	0.87	3.23	8.28	4.38
PAFT	**1.19**	**0.39**	**0.45**	**2.08**	**1.02**
↪ Improv.	×3.3	×2.23	×3.82	×3.00	×3.25

Experimental Results

* Hyperparameter Robustness.

- 기본 세팅 K = 4, T = 3)으로 거의 최적의 성능을 달성, 모든 태스크에서 평균 정확도 87.46%(±1.34)
- PAFT는 하이퍼파라미터 튜닝의 필요성을 줄여, 실제 애플리케이션을 위한 실용적 및 효율적 솔루션

Table 5: Performance comparison of PAFT with varying hyperparameters K (number of iterations per prompt) and T (number of epochs) across multiple reasoning and reading comprehension tasks. Results are reported as mean accuracy (± standard deviation) on the Hellaswag, PIQA, Winogrande, RACE-mid, and RACE-high datasets. The best results for each metric are highlighted in bold.

# K and T	Hellaswag	PIQA	Winogrande	RACE-mid	RACE-high	Average
$K = 1, T = 3$	93.58 (± 1.47)	**89.33** (± 0.63)	81.78 (± 1.11)	86.30 (± 2.73)	84.35 (± 2.24)	87.07 (± 1.64)
$K = 2, T = 3$	93.59 (± 1.24)	88.37 (± **0.49**)	**82.09** (± **0.81**)	86.30 (± 2.64)	84.02 (± 2.24)	86.87 (± 1.48)
$K = 4, T = 3$	**93.83** (± 1.10)	89.07 (± 0.53)	81.96 (± 1.15)	**87.26** (± **2.23**)	**85.17** (± 1.71)	**87.46** (± **1.34**)
$K = 8, T = 3$	**93.83** (± **0.70**)	88.99 (± 0.59)	82.69 (± 0.97)	86.25 (± 2.75)	84.36 (± 2.06)	87.22 (± 1.41)
$K = 1, T = 6$	93.37 (± 1.47)	88.32 (± 0.68)	81.05 (± 3.44)	84.40 (± 2.30)	83.34 (± **1.66**)	86.10 (± 1.91)

Experimental Results

* Impact of Training Prompt Quantity

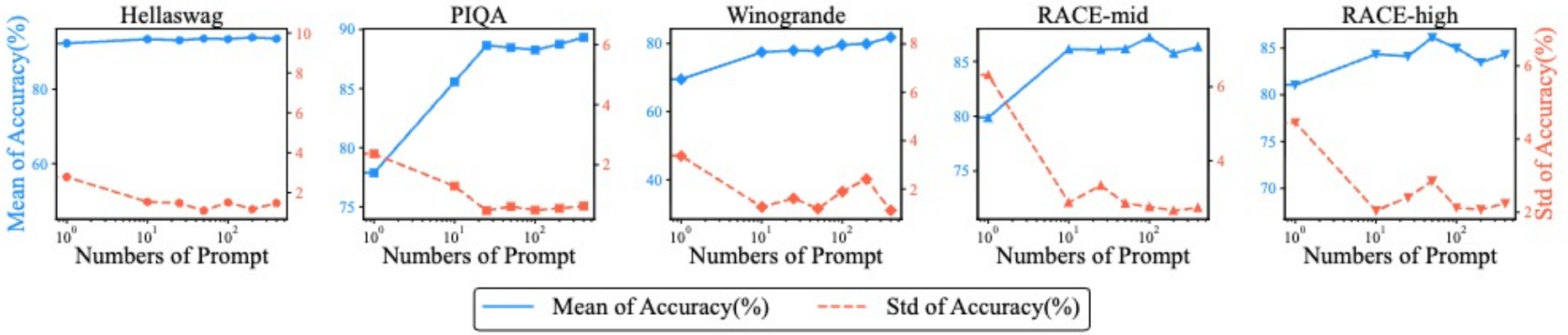


Figure 8: Scaling Law of Training Prompt Numbers: Mean and Standard Deviation of Accuracy Across Different Datasets. The x-axis represents the number of prompts on a logarithmic scale, while the y-axis shows the mean accuracy (left) and standard deviation of accuracy (right) for each dataset.

감사합니다