# Self-Alignment in Large Language Models

NLP&AI 연구실
2024 여름세미나

**이정섭**

# Introduction

- **Self-Alignment with Instruction Backtranslation**
  (ICLR 2024 Oral, Meta)

- **Self-Rewarding Language Models**
  (arXiv 2024, Meta)

# Self-Alignment with Instruction Backtranslation

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer
Jason Weston & Mike Lewis
{xianl,jase,mikelewis}@meta.com

**ICLR 2024 Oral**

**이정섭**

# Introduction

## Background: Back-Translation

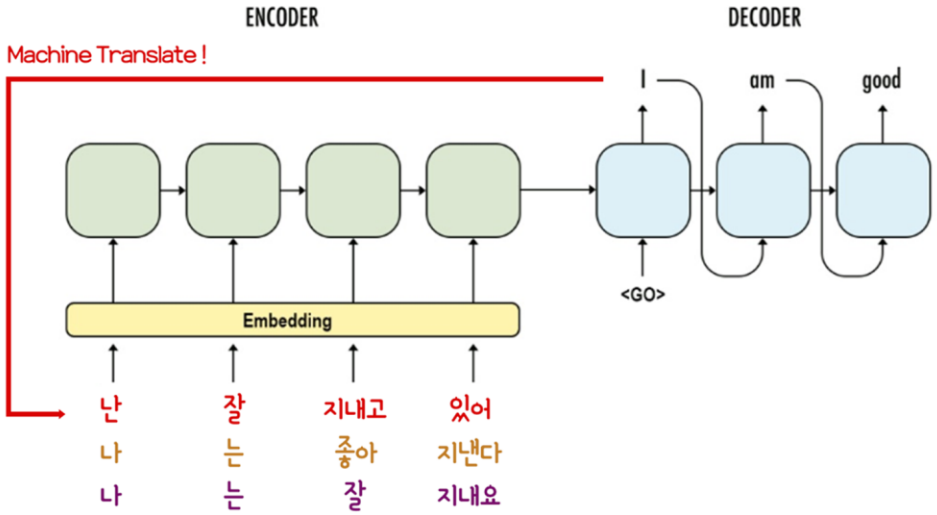
- 번역 모델을 지도학습 하기 위해서는 병렬 코퍼스가 필요하다.
- 예를 들어 아래 처럼 같은 의미를 문장이 여러가지 언어로 주어져야 한다.

| German | English |
| --- | --- |
| Ich befürworte die Feststellung des Berichterstatters, dass der Weltraum nicht b ˙·. | I support the rapporteur's stipulation that space should not become weaponised. |
| Obgleich sie wohlwollend gemeint sind, werden die vorgeschlagenen Änderungen auf ˙·. | Although meant well, the proposed amendments will, in my view, in fact, hamper t ˙·. |
| Ich komme zum Schluß, Herr Präsident. Ich möchte sagen, daß das Recht, daß das G ˙·. | To conclude, Mr President, I want to say that the law represents or should repre ˙·. |
| Ich denke jedoch wirklich, dass wir in den nächsten paar Jahren erleben, dass si ˙·. | I do think, however, that in the next few years, we will see them abandoning the ˙·. |
| Deshalb sehe ich keinen Zusammenhang zwischen der Stilllegung von Ignalina und d ˙·. | I therefore see no link between the closure of Ignalina and security of supply i ˙·. |
| 10 total › | |

# Introduction

## Background: Back-Translation

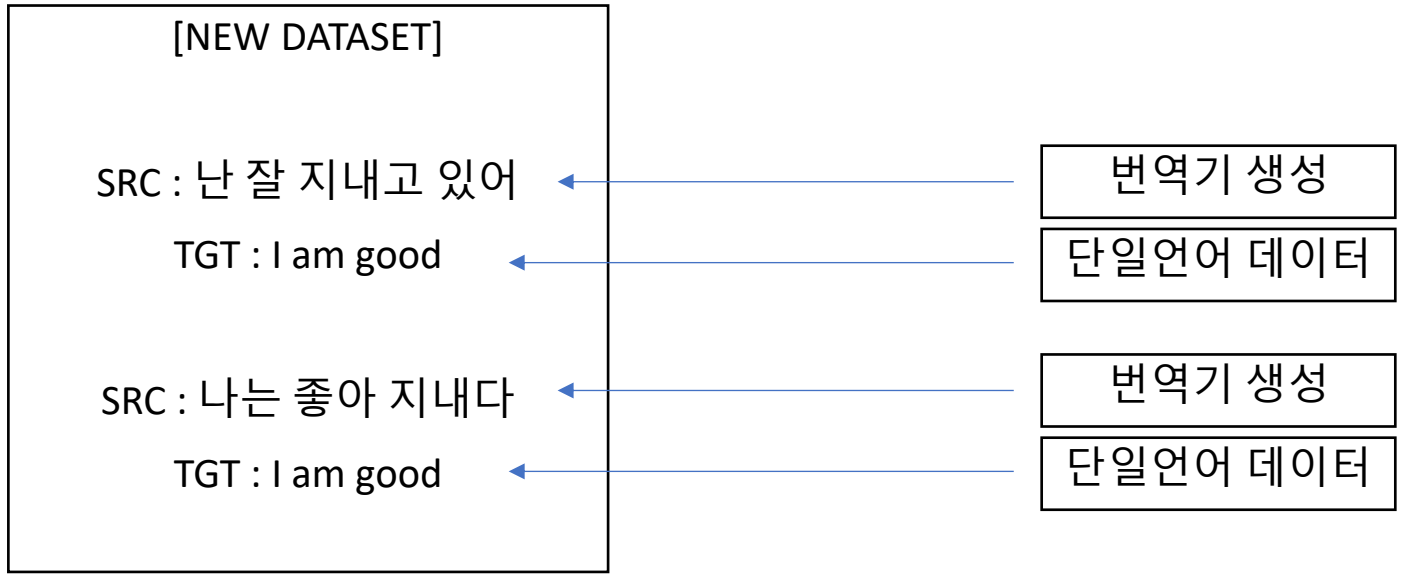
- 그런데 우리에게는 병렬 데이터가 많이 없다. (모으기도 힘들다)
- 그런데 단일언어 데이터는 정말 많다. (Wikipedia, Namuwiki, CNN News, …)
- 그런 데이터를 좀 이용해 볼 수 없을까? = Back-Translation



# Introduction

## Background : Back-Translation

- 1. 우리가 가진 적은 병렬데이터를 사용해서 반대방향 번역모델을 만든다.
- 2. 우리가 가진 **단일언어 데이터를 만들어진 반대방향 번역모델에 입력해서 결과**를 얻는다.
- 3. **번역기가 만든 데이터는 이제 입력이 되고 가지고 있던 단일언어 데이터는 출력**이 된다.
- **아주 간단한 아이디어이지만 성능을 굉장히 많이 향상 시켰다 !! (다소 충격적일 정도…)**




| | En–De | En–Fr |
|---|---|---|
| a. Gehring et al. (2017) | 25.2 | 40.5 |
| b. Vaswani et al. (2017) | 28.4 | 41.0 |
| c. Ahmed et al. (2017) | 28.9 | 41.4 |
| d. Shaw et al. (2018) | 29.2 | 41.5 |
| DeepL | 33.3 | **45.9** |
| Our result | **35.0** | 45.6 |
| *detok. sacreBLEU*[3] | *33.8* | *43.8* |

KOREA UNIVERSITY

# Introduction

- Human annotated dataset을 만드는 것은 굉장히 어려움
- 심지어 퀄리티가 매우 중요함

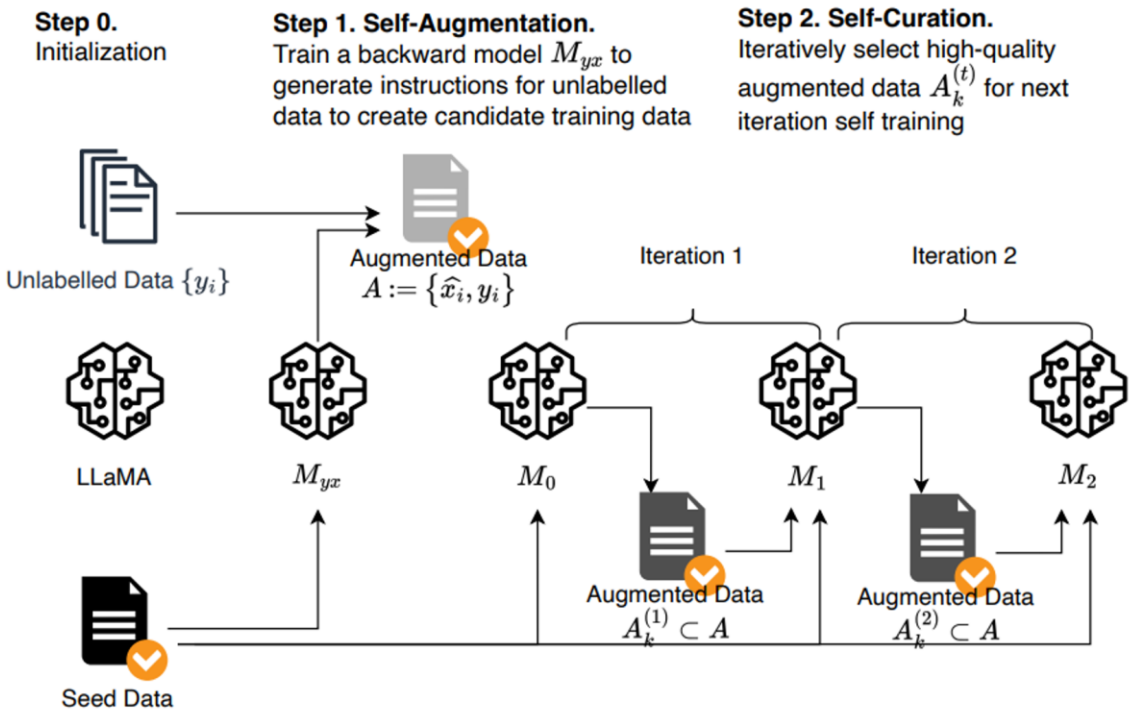
➔ unlabeled data를 사용해서 해결할 수 없을까?

Human Instruction Dataset도 Back-Translation 처럼
Unlabeled Data를 자동으로 annotation 하면 좋겠다 !

**Back-Translation에서 영감을 받은 instruction backtranslation 제안**

# Method

## Instruction Backtranslation

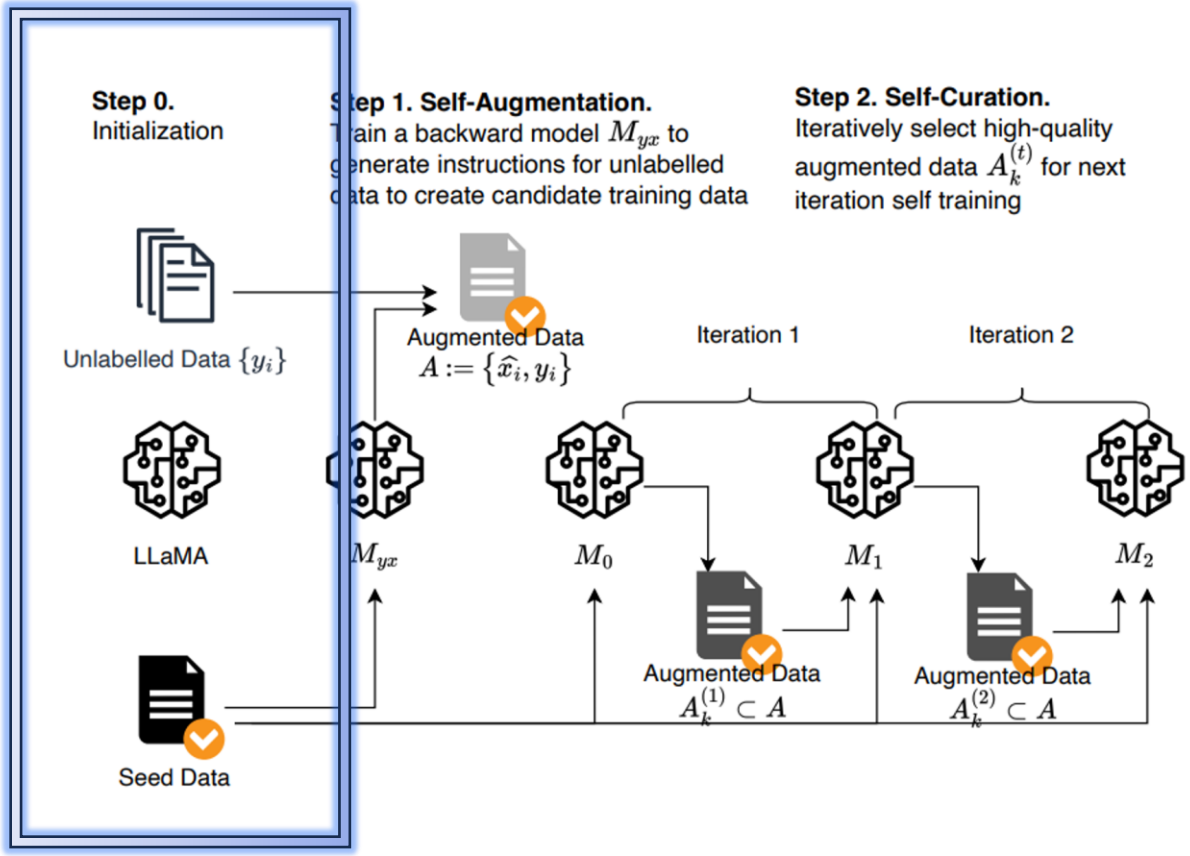



- **Self-augment**:
  unlabeled data에 대한 instruction을 생성해서 instruction tuning을 위한 후보 학습 데이터인 {Instruction, Output} 생산

- **Self-curate**:
  기본 모델을 튜닝해서 instruction을 따르도록 학습 데이터를 선택. 반복적으로 수행

# Method

## Instruction Backtranslation

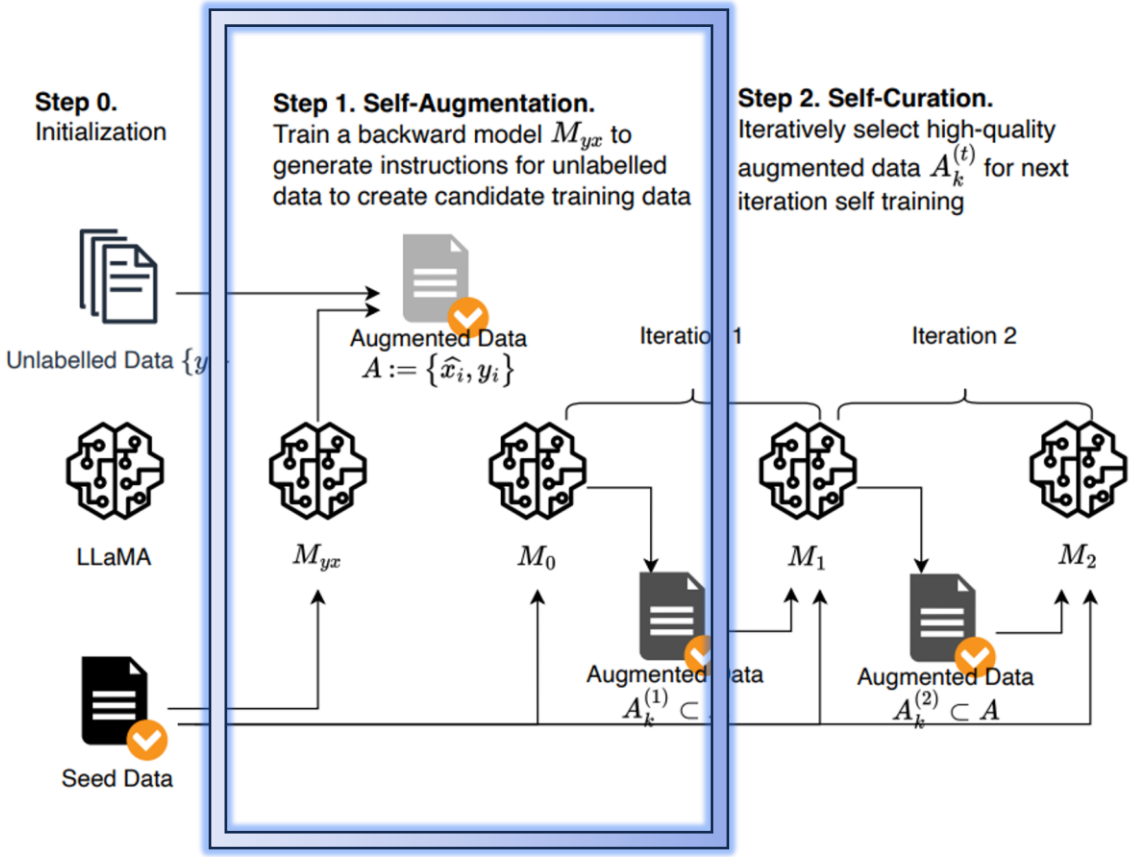



### 1) Initialization

- 초기 시드 모델을 만들기 위해, {Instruction, Output} 시드 데이터 준비
- unlabeled data는 중복제거, 필터링 등의 휴리스틱 방법으로 낮은 품질 데이터를 제거한 웹코퍼스 사용

# Method

## Instruction Backtranslation



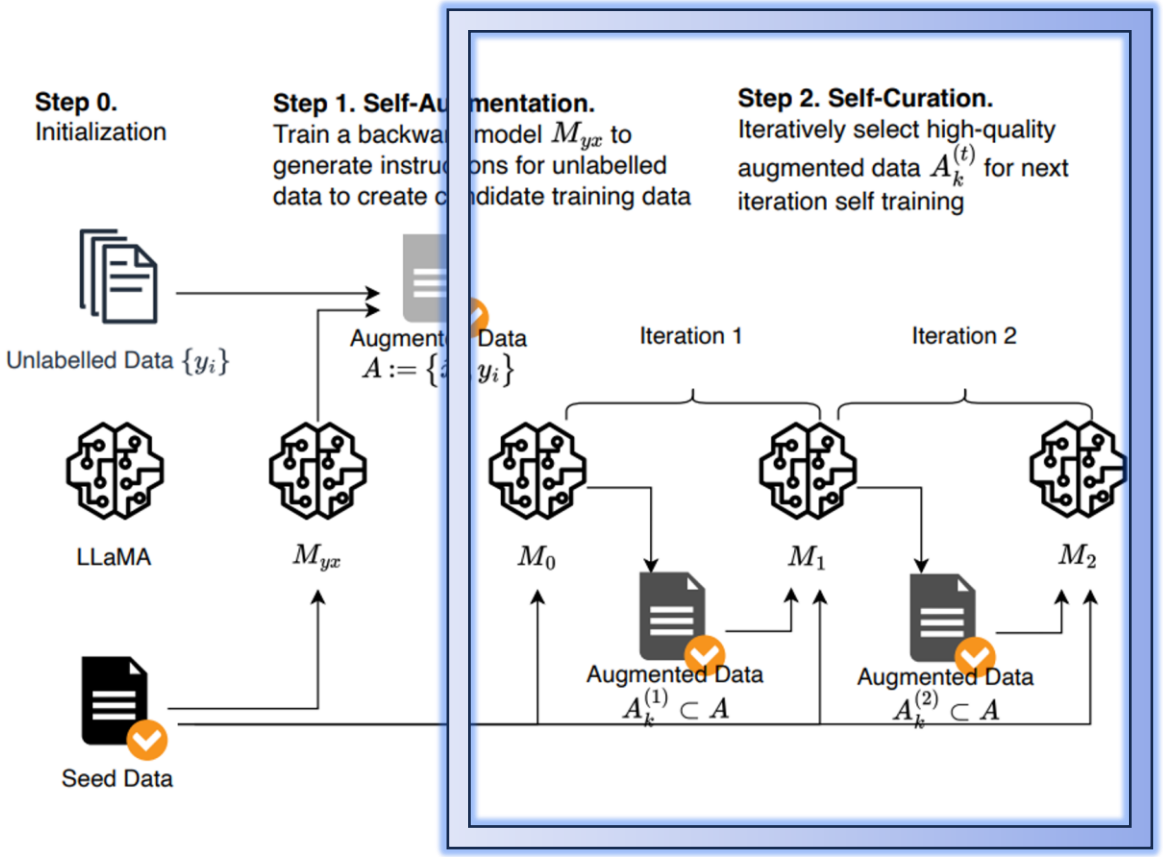

### 2) Self-Augmentation (Generating Instruction)

- 시드 데이터 {Output, Instruction} pair $\{(y_i, x_i)\}$ 으로 LLaMA를 튜닝해서 역방향 모델(backward model) $M_{yx} := p(x|y)$ 를 얻음
- Unlabeled Instruction data인 $y_i$ 를 역방향 모델에 넣어 $\mathcal{A} := \{(\hat{x}_i, y_i)\}$ 얻음

➔ 만들어진 데이터가 구질 수도 있음. 따라서 고품질 subset의 curation을 고려해야 함

# Method

## Instruction Backtranslation




### 3) Self-Curation (Selecting high quality examples)

- 만들어진 데이터 중 고품질 examples 만을 선택하는 단계
- 시드 Instruction Model인 M0를 시작으로, 튜닝하여 사용
- M0로 각 증강 샘플인 $\mathcal{A} := \{(\hat{x}_i, y_i)\}$에 대해 데이터 품질을 5점 척도로 평가
- 점수가 $a_i \geq k$ 인 샘플들을 선택해서 curated set $\mathcal{A}_k^{(1)}$생성
- Iterative self-curation : 위 과정을 M0에서 Mt 까지 반복해서 최종 증강 데이터 $\mathcal{A}_k^{(t)}$ 생성
- 논문에서는 t=2 까지 반복

# Method

## Instruction E

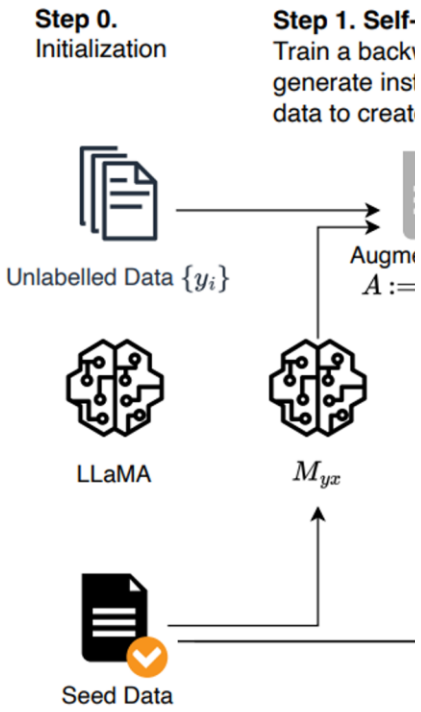



Table 19: Prompt used in the *self-curation* step to evaluate the quality of a candidate (instruction, output) pair in the dataset derived from self-augmentation.

```
Below is an instruction from an user and a candidate answer. Evaluate whether or
not the answer is a good example of how AI Assistant should respond to the user's
instruction. Please assign a score using the following 5-point scale:
1: It means the answer is incomplete, vague, off-topic, controversial, or not
exactly what the user asked for. For example, some content seems missing, numbered
list does not start from the beginning, the opening sentence repeats user's question.
Or the response is from another person's perspective with their personal experience
(e.g. taken from blog posts), or looks like an answer from a forum. Or it contains
promotional text, navigation text, or other irrelevant information.
2: It means the answer addresses most of the asks from the user. It does not
directly address the user's question. For example, it only provides a high-level
methodology instead of the exact solution to user's question.
3: It means the answer is helpful but not written by an AI Assistant. It addresses
all the basic asks from the user. It is complete and self contained with the
drawback that the response is not written from an AI assistant's perspective, but
from other people's perspective. The content looks like an excerpt from a blog post,
web page, or web search results. For example, it contains personal experience or
opinion, mentions comments section, or share on social media, etc.
4: It means the answer is written from an AI assistant's perspective with a
clear focus of addressing the instruction. It provide a complete, clear, and
comprehensive response to user's question or instruction without missing or
irrelevant information. It is well organized, self-contained, and written in a
helpful tone. It has minor room for improvement, e.g. more concise and focused.
5: It means it is a perfect answer from an AI Assistant. It has a clear focus on
being a helpful AI Assistant, where the response looks like intentionally written
to address the user's question or instruction without any irrelevant sentences. The
answer provides high quality content, demonstrating expert knowledge in the area, is
very well written, logical, easy-to-follow, engaging and insightful.

Please first provide a brief reasoning you used to derive the rating score, and
then write "Score: <rating>" in the last line.

<generated instruction>
<output>
```

**uality examples)**

을 선택하는 단계

., 튜닝하여 사용

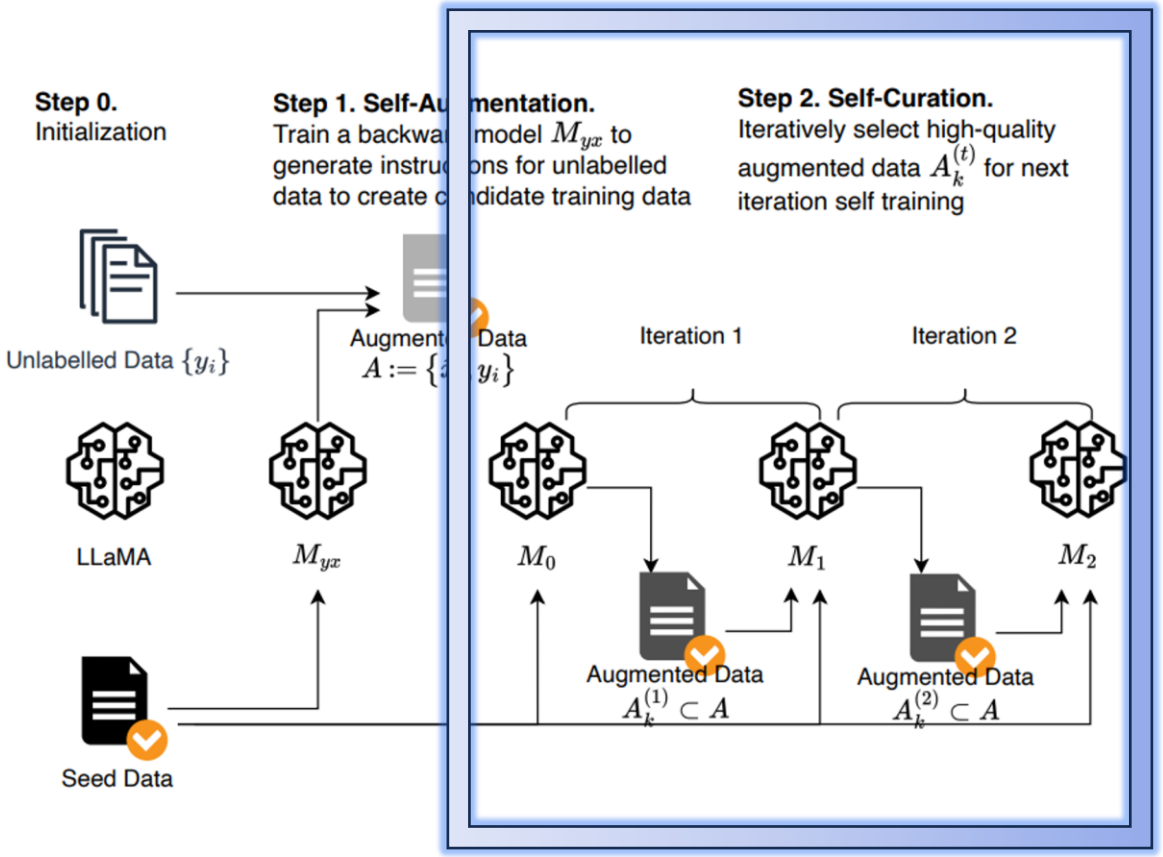
대해 데이터 품질을 5점

curated set $\mathcal{A}_k^{(1)}$생성

Mt 까지 반복해서 최종

# Method

## Instruction Backtranslation

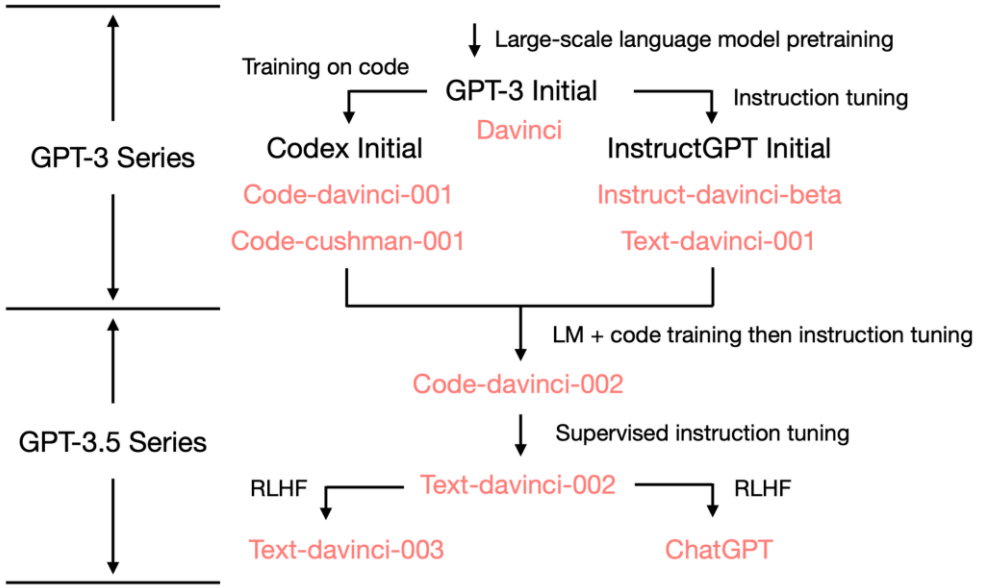



### 3) Self-Curation (Selecting high quality examples)

- 시드 데이터의 system prompt에는 “Answer in the style of an AI Assistant”를 입력하고, $\mathcal{A} := \{(\hat{x}_i, y_i)\}$에 대해서는 “Answer in the style of an AI Assistant” 사용하여 Iterative Curation 수행

# Experiments

## Setup




- **시드 데이터**: 3200개의 examples인 OpenAssistant 사용
  - 이 중, human annotated data인 0 등급을 기준으로 사용 (1턴만)
- **Unlabeled Data**: Clueweb Corpus에서 502k 데이터를 샘플링해서 사용
- **모델**: LLaMa [7B, 33B, 65B] 모델 사용. 입력 토큰이 아닌 출력 토큰에 대해서만 loss를 optimization
  - T: 0.7, p: 0.9
  - 이렇게 학습한 모델을 **Humback** 이라고 부름
- **Baselines**:
  - text-davinci-003: GPT-3을 기반의 Instruction Tuning 모델로, 인간이 작성한 지시사항, 출력으로 RLHF
  - LIMA: LLaMa 모델에 사람 전문가가 작성한 Instruction 1000개로 튜닝 (StackOverflow, WikiHow 등)
  - Guanaco: OpenAssistant 데이터셋에서 9000개의 멀티턴 튜닝된 LLaMa 모델 (Humback의 시드 데이터는1턴만 사용)

# Experiments

## Evaluation Setup

- **테스트 프롬프트 (평가 데이터셋)**:
  - Vicuna의 80개 프롬프트
  - Self-instruct의 252 프롬프트
  - OpenAssistant의 188프롬프트
  - Koala의 156 프롬프트
  - HH_RLHF의 129 프롬프트
  - LIMA의 300 프롬프트
  - 크라우드 소싱된 64개 프롬프트

총 1130개의 unique prompt이고, 작문, 코딩, 수학, 정보검색, 역할놀이, 안전 등 다양한 카테고리 포함

- **Validation 프롬프트 (검증 데이터셋)**:
  - 테스트셋에 포함되지 않은 AlpacaEval 의 256개 프롬프트

- **평가방법:**
  - AlpacaEcal의 자동평가: – GPT-4의 모델 win rate
  - Human Evaluation

# Experiments

tatsu-lab / alpaca_eval

Type / to search

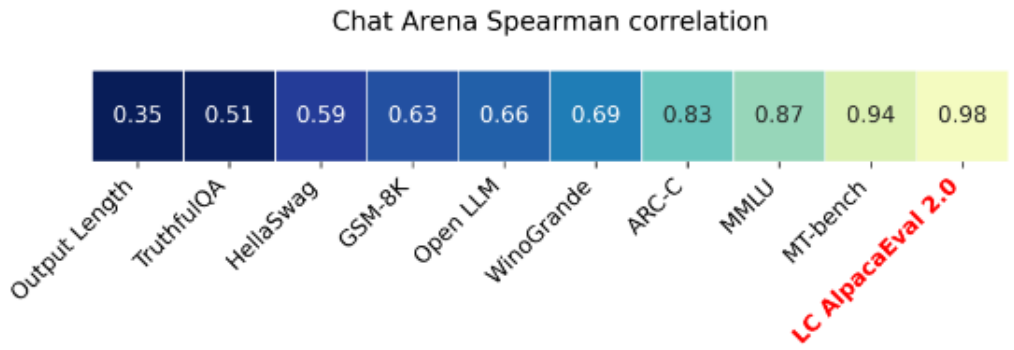
Code | Issues 1 | Pull requests | Actions | Projects | Security | Insights

Star 1.1k

## AlpacaEval : An Automatic Evaluator for Instruction-following Language Models

Code License Apache 2.0 | Data License CC By NC 4.0 | python 3.10+ | discord server

**AlpacaEval 2.0 with length-controlled win-rates** (paper) has a spearman correlation of **0.98** with ChatBot Arena while costing less than **$10** of OpenAI credits run and running in less than 3 minutes. Our goal is to have a benchmark for chat LLMs that is: fast (< 5min), cheap (< $10), and highly correlated with humans (0.98). Here's a comparison with other benchmarks:



# Results

## 데이터 통계

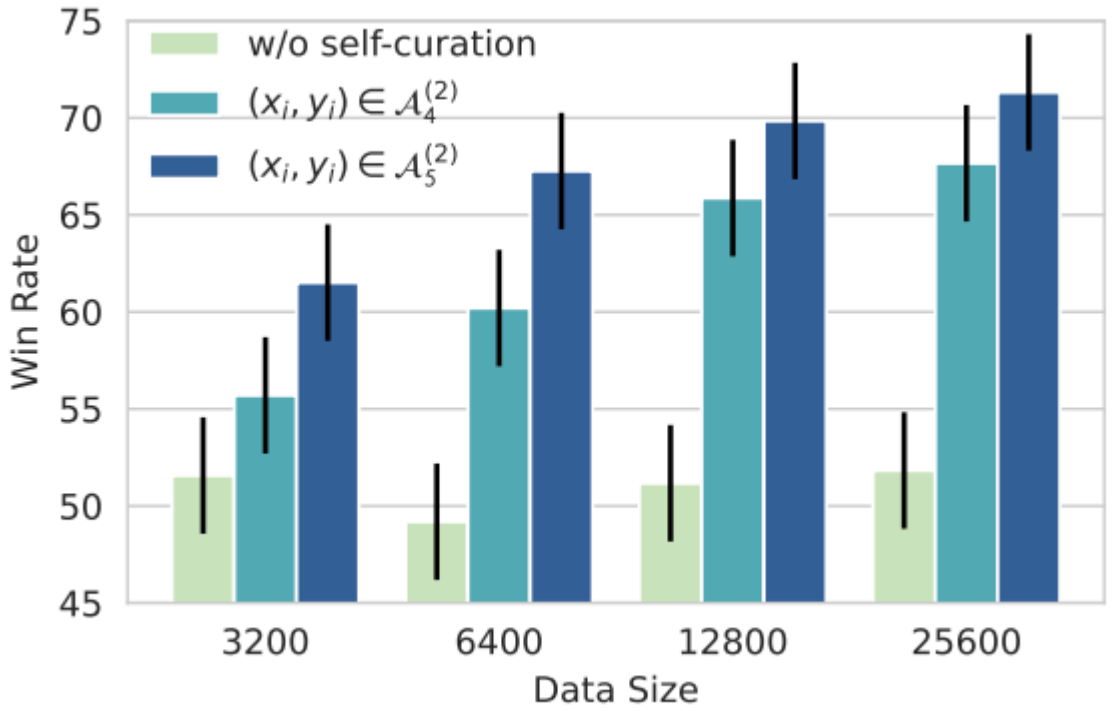
| | # examples | Instruction Length | Output Length |
|---|---|---|---|
| Seed data | 3200 | $148 \pm 322$ | $1072 \pm 818$ |
| Augmented data, $\mathcal{A}_5^{(2)}$ | 41821 | $115 \pm 175$ | $1663 \pm 616$ |
| Augmented data, $\mathcal{A}_4^{(2)}$ | 195043 | $206 \pm 298$ | $1985 \pm 649$ |
| Augmented data, all | 502133 | $352 \pm 134$ | $1722 \pm 653$ |


(a) Seed data.


(b) Augmented data in $\mathcal{A}_5$

# Results

## 데이터 퀄리티 vs 데이터 양




**w/o self-curation**

- 데이터 양을 늘린다고 성능이 좋아지지 않음

**A_4, A_5**

- 퀄리티가 좋은 데이터는 많을 수록 좋다.

# Results

## 데이터 스케일링 효율성

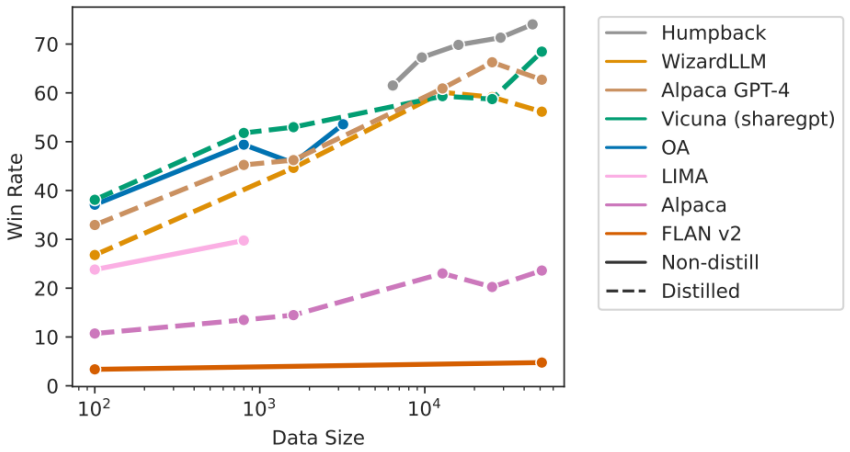



Figure 3: Comparing data efficiency of different instruction tuning datasets. The y-axis is the win rate against text-davinci-003 when finetuning 7B LLaMa with the given instruction tuning dataset. Dashed lines depict models that use distillation from more powerful models to construct data, and methods with solid lines do not.

Table 2: Scaling coefficient $\alpha$ of representive instruction datasets created using differnet methods and data sources.

| | Source | $\alpha \uparrow$ |
|---|---|---|
| Humpback (this work) | OA, self-augmented and self-curated | 6.95 |
| WizardLLM[2] (Xu et al., 2023) | Distilled from ChatGPT, GPT-4 (June 2023) | 5.69 |
| Alpaca-GPT4 (Peng et al., 2023) | Distilled from GPT-4 (April 2023) | 5.40 |
| Vicuna (Chiang et al., 2023) | Distilled from ChatGPT, GPT-4 (June 2023) | 4.53 |
| Open Assistant (OA) (Köpf et al., 2023) | Human Annotation | 4.43 |
| LIMA (Zhou et al., 2023) | Human Annotation, Community QA | 2.86 |
| Alpaca (Taori et al., 2023) | Distilled from ChatGPT (March 2023) | 1.99 |
| FLAN v2 (Chung et al., 2022) | Instruction data for NLP tasks | 0.22 |

## text-davinci-003과 비교한 Win rate

각 모델이 주어진 튜닝데이터로 7B LLaMa를 튜닝한 성능 비교.

- 점선은 powerful 모델을 사용해서 만든 증류 데이터, 실선은 다른 출처에서 가져온 데이터

Humpback의 경우 k = 5, t = 2

➔ 데이터 양이 늘어날 수록, 성능 증가

➔ 증류된 데이터가 대부분 성능이 좋음

# Results

## Model Quality – AlpacaEval

Table 3: Results on the Alpaca leaderboard (win rate over text-davinci-003 evaluated by GPT-4). Humpback outperforms other non-distilled models by a wide margin with efficient data scaling beyond human annotated data.

| | | Annotated Examples | Total Examples | Win Rate % |
|---|---|---|---|---|
| Non-distilled | Humpback 33B | 3k | 45k | **79.84** |
| | OASST RLHF 33B | 161k | 161k | 66.52 |
| | Guanaco 33B | 9k | 9k | 65.96 |
| | OASST SFT 33B | 161k | 161k | 54.97 |
| Non-distilled | Humpback 65B | 3k | 45k | **83.71** |
| | Guanaco 65B | 9k | 9k | 71.80 |
| | LIMA 65B | 1k | 1k | 62.70 |
| Non-distilled | Humpback 70B | 3k | 45k | 87.94 |
| | LLaMa2 Chat 70B | 1.4m | 5.7m | **92.66** |
| Distilled | Vicuna 33B | 140k | 140k | **88.99** |
| | WizardLLM 13B | 190k | 190k | 86.32 |
| | airoboros 65B | 17k | 17k | 73.91 |
| | Falcon Instruct 40B | 100k | 100k | 45.71 |
| Proprietary | GPT-4 | | | **95.28** |
| | Claude 2 | | | 91.36 |
| | ChatGPT | | | 89.37 |
| | Claude | | | 88.39 |

## text-davinci-003과 비교한 Win rate

- **Non-distilled**: 어떠한 외부 모델(예: ChatGPT, GPT-4 등)에 의존하지 않고 훈련된 LLaMa 모델. 대부분의 모델은 인간 주석 데이터에 크게 의존
- **Distilled**: 더 강력한 외부 모델을 루프에 포함하여 훈련된 모델 (외부 모델에서 데이터를 증류)
- **Proprietary**:독점 데이터 및 기술을 사용하여 훈련된 모델 (ChatGPT, Claude)

➔ Humpback은 non-distilled에서 적은 데이터로 최고 성능

➔ distilled 데이터도 좋음

➔ Humpback의 self-curation이 없을 경우, non-distilled는 개구짐..

# Results

## Model Quality – Human Evaluation

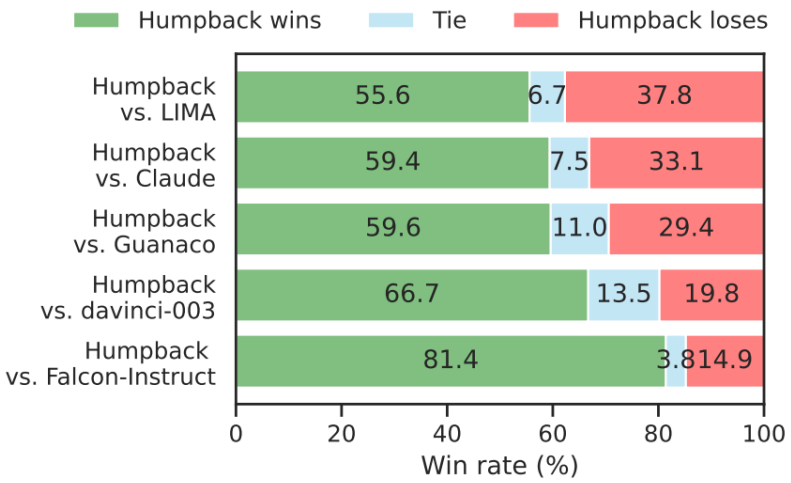



Figure 4: Humpback is preferred to both open source (e.g. LIMA(Zhou et al., 2023) (65B), Guanaco (Dettmers et al., 2023) (65B),Falcon-Instruct(Almazrouei et al., 2023)) (40B) and proprietary (e.g. davinci-003(Ouyang et al., 2022) and Claude(Bai et al., 2022a)) instruction-tuned models in pairwise human preference judgements.

## 1130개 프롬프트로 테스트 (Win Rate)

각 프롬프트에 대해, 두 모델의 출력 제시.

사람에게 세 가지 옵션 중에서 선택하도록 요청 (순서 랜덤)

1) 첫 번째 모델의 출력이 두 번째 모델보다 훨씬 나음

2) 두 번째 모델의 출력이 첫 번째 모델보다 훨씬 나음

3) 두 출력 간에 유의미한 차이가 없음

➔ AlpacaEval과 분포 유사

# Resu

## Model Q

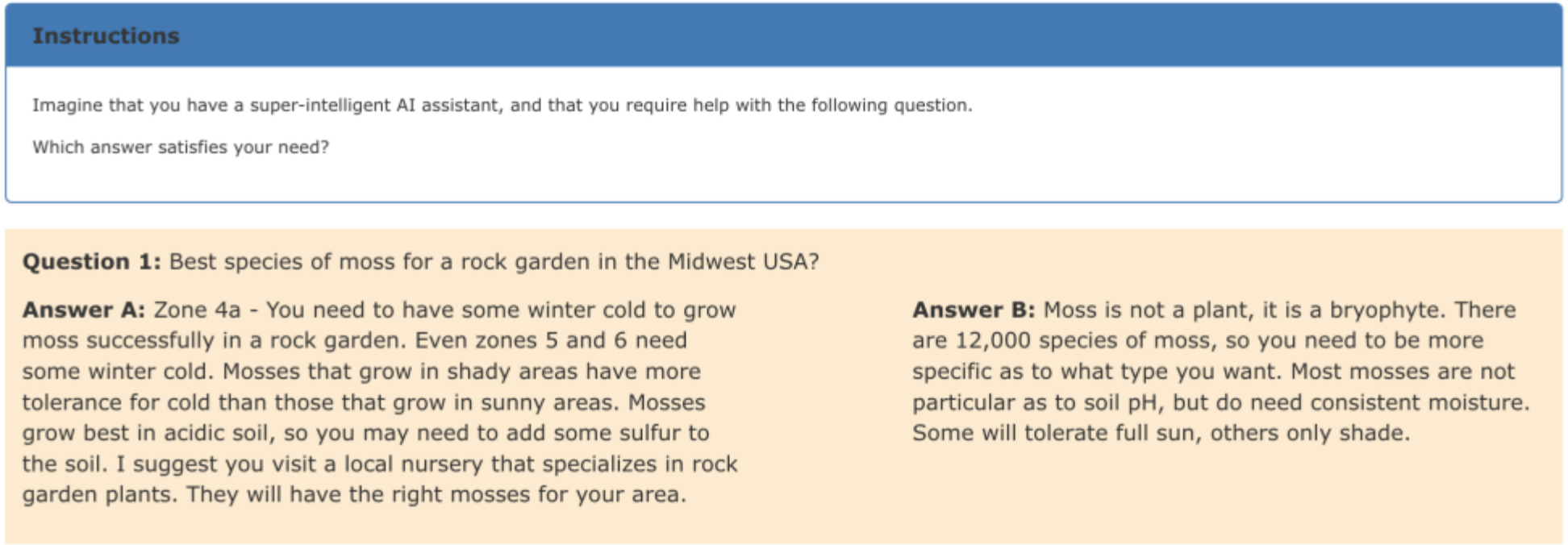



Figure 4: Humpback
(Dettmers et al., 202
davinci-003(Ouyang
human preference ju

# Results

## Model Quality – Commonsense Reasoning & MMLU

Table 4: Comparison on zero-shot commonsense reasoning and MMLU.

| | SIQA | PIQA | Arc-E | Arc-C | OBQA | MMLU |
|---|---|---|---|---|---|---|
| LLaMA 33B | 50.2 | 82.2 | 80.0 | 54.8 | 58.6 | 49.5 |
| Humpback 33B | 53.4 | 74.5 | 84.4 | 68.5 | 46.4 | 55.4 |
| LLaMA 65B | 52.3 | 82.8 | 78.9 | 56.0 | 60.2 | 54.8 |
| Humpback 65B | 60.4 | 78.9 | 88.7 | 73.0 | 64.0 | 59.0 |

Table 7: Results on MMLU by domains.

| | Humanities | STEM | Social Sciences | Other | Average |
|---|---|---|---|---|---|
| LLaMA 65B, 5-shot | 61.8 | 51.7 | 72.9 | 67.4 | 63.4 |
| LLaMA 65B, 0-shot | 63.0 | 42.5 | 62.3 | 57.5 | 54.8 |
| Humpback 65B, 0-shot | 65.6 | 47.6 | 68.1 | 60.8 | 59.0 |

## 5 Commonsense Reasoning Benchmarks & MLLU (Massive Multitask Language Understanding)

마찬가지로 개선됨 !

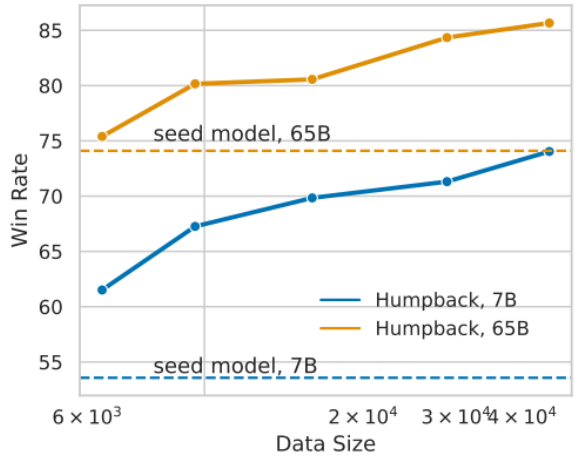
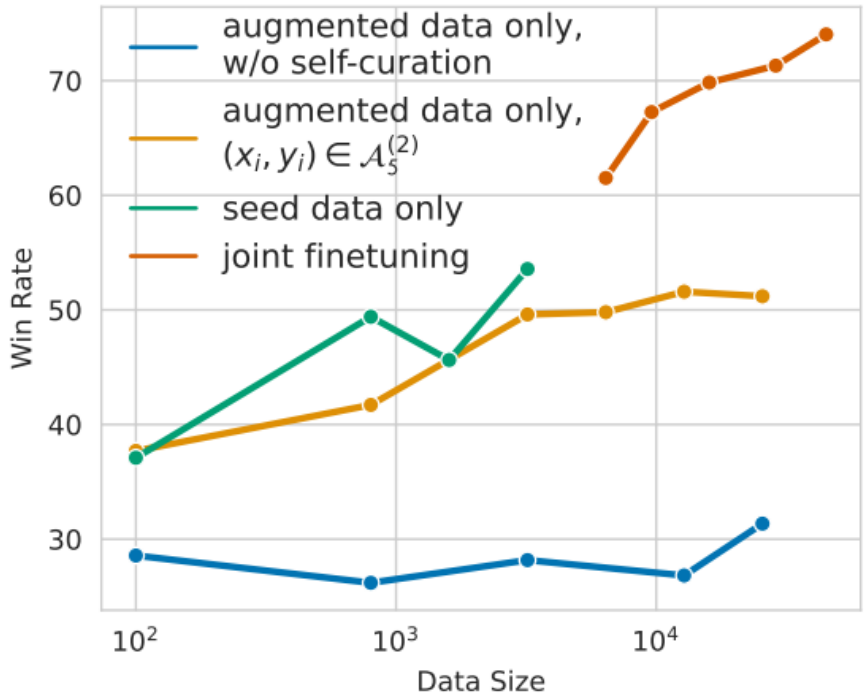



Figure 7: Scaling up self-curated instruction data $\mathcal{A}_5$ brings improvement in both small (7B) and large (65B) LLaMa finetuned models, and neither model is saturated with 40,000 instructions.

# Ablations

## Training on self-augmented data only




1) 증강된 데이터만 사용 (Self-augmentation). Self-curation 안함

2) 증강된 데이터만 사용 + Self-curation

3) 시드 데이터만 사용

4) Humpback (제안하는 방법)

**3) Self-Curation (Selecting high quality examples)**

- 시드 데이터의 system prompt에는 "Answer in the style of an AI Assistant"를 입력하고, $\mathcal{A} := \{(\hat{x}_i, y_i)\}$에 대해서는 "Answer in the style of an AI Assistant" 사용하여 Iterative Curation 수행

➔ 시드데이터 사용 안하면, 성능이 더 나빠질 수 있음

➔ self-curation하면 데이터 많아질 수록 성능이 좋아지긴 한다

➔ self-curation 안하면 성능이 안좋아짐

# Ablations

## System Prompts

Table 5: Effect of system prompt. We report mean win rate and its standard error.

| Train | Inference | Win Rate (%) |
|---|---|---|
| $S_a$ for seed data, $S_w$ for augmented data | $\{S_a, S_w\}$ | 66.47 ±3.04 |
| no system prompt | no system prompt | 59.96 ±3.09 |
| $S_a$ for seed data, $S_w$ for augmented data | $S_a$ | 62.69 ±3.06 |
| $S_a$ for seed data, $S_w$ for augmented data | no system prompt | 62.70 ±3.07 |

**3) Self-Curation (Selecting high quality examples)**

- 시드 데이터의 system prompt에는 "Answer in the style of an AI Assistant"를 입력하고, $\mathcal{A} := \{(\hat{x}_i, y_i)\}$에 대해서는 "Answer in the style of an AI Assistant" 사용하여 Iterative Curation 수행

➔ 프롬프트 구분해서 넣는 것이 성능 더 좋음

# Appendix

| | |
|---|---|
| **Output:** | Inclusive Sports Coaching provides 1:1 Programs for individuals looking to develop their sporting skills, as well as improve their self confidence and opportunities for social and community inclusion.<br>We recommend an 8 or 12 Session program to identify areas for improvement and sporting skills, conduct drills and physical activities to work towards specific outcomes, while engaging with the client in areas such as listening, memory retention, cognitive processing, social interaction, encouraging conversations, accepting and giving constructive feedback, and other areas as needed.<br>At the halfway point we produce a status report on progress, and have found parents/carers often share this with OT's, Physios and Teachers as a way to share information on the individual and provide a strong network of support.<br>At the end of the program we produce a final report, with recommendations for ongoing improvement, potential for progress along the person's chosen sport pathway where applicable, etc. |
| **Generated instruction:** | I have a business called Inclusive Sports Coaching. We provide 1:1 sport coaching for people with disabilities. I want to have some materials on hand to give to parents when they enquire about our services. What do you recommend I include in these materials? |

Table 10: Examples of instructions generated by the backward model $M_b$.

Table 11: Improvement of our method over the seed model in mathematical reasoning.

| | |
|---|---|
| **Prompt:** | Jeff takes 20 minutes to jog around the race course one time, and 25 minutes to jog around a second time. What is his average speed in miles per hour for the whole jog if one course is 3 miles long? |
| $\mathcal{M}_0$ | We can start by converting the minutes into hours:<br>20 minutes = 0.33 hours 25 minutes = 0.42 hours<br>Total time = 0.75 hours<br>Next, we can calculate his average speed using the distance and total time:<br>Average speed = Total distance / Total time Average speed = 3 miles / 0.75 hours Average speed = 4 miles per hour<br>Therefore, Jeff's average speed for the whole jog was 4 miles per hour. |
| $+\mathcal{A}_5$ | We can calculate Jeff's average speed in miles per hour for the whole jog as follows:<br>Total time taken to complete two jogs = 20 minutes + 25 minutes = 45 minutes<br>Distance covered in two jogs = 3 miles * 2 = 6 miles<br>Average speed = Total distance / Total time<br>Average speed = 6 miles / 45 minutes<br>Average speed = 13.33 miles per hour<br>Therefore, Jeff's average speed in miles per hour for the whole jog is 13.33 miles per hour. |

**Training.** For experiment on data scaling efficiency, models were trained with increasing number of examples $N$ for each dataset. For fair comparison, for each $N \in \{100, 800, 1600, 3200, 6400, 12800, 25600, 51200\}$, all datasets were trained for the same number of steps with the same batch size as is shown in Table 18.

# Self-Rewarding Language Models

Weizhe Yuan[1,2] Richard Yuanzhe Pang[1,2] Kyunghyun Cho[2]
Xian Li[1] Sainbayar Sukhbaatar[1] Jing Xu[1] Jason Weston[1,2]

[1] Meta [2] NYU

**arXiv 2024**

**이정섭**

# Introduction

**초인적인 LLM 모델 학습을 위해 super human feedback이 필요**

➔ 현재의 접근 방법은 주로 **인간의 선호도(human preference data)로 보상 모델을 학습**

- 인간의 성능 수준에 의해 병목이 될 수 있음
- 이러한 고정된 보상 모델은 LLM 훈련 중에 개선될 수 없음

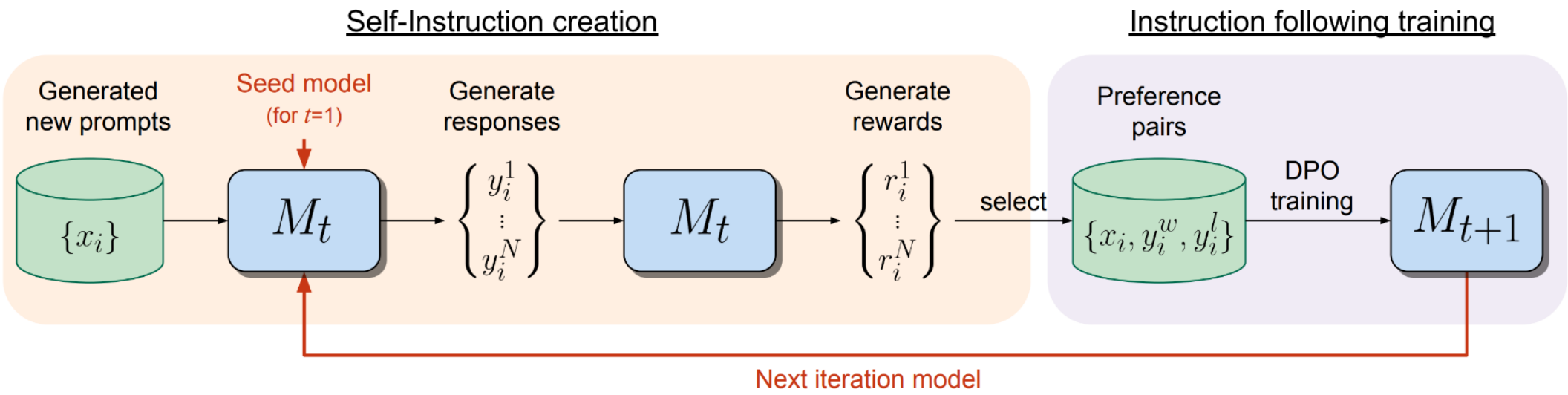
해당 연구에서는

**훈련 중에 자체 보상을 제공하기 위해 LLM-as-a-Judge 프롬프팅을 통해 언어 모델 자체를 사용하는 'Self-Rewarding Language Models'를 연구**

➔ **LLM alignment 중에 지속적으로 업데이트되는 자체 보상 모델을 훈련시키는 방법을 제안**

# Method

## Self-Rewarding Language Models




### 1) Self-Instruction creation

**Generated new prompts**는 모델 Mt에서 후보 응답을 생성하는 데 사용되며,
이는 LLM-as-a-Judge 프롬프트를 통해 자체 reward 예측

### 2) Instruction following training

**Preference pairs**은 **Generated responses**에서 선택되고, DPO 학습에 사용되어 모델 Mt+1이 됨

# Method

## Self-Rewarding Language Models

### 1) Initialization
- 시드 데이터 준비
  - Instruction Fined-tuning (Instruction Fine-Tuning, IFT) 데이터
  - LLM-as-a-Judge Instruction Following 데이터 (Evaluation Fine-Tuning, EFT)

### 2) Self-Instruction 생성
학습된 LLM으로 학습 셋 수정 진행.
프롬프트 생성 → 후보응답 생성 → 후보응답 평가
세 과정으로 학습에 사용할 self-instruction 데이터 생성 및 선별

### 3) Instruction Following Training & Overall Self-Alignment Algorithm
사전학습 모델 M0을 Mt까지 학습하는 과정

# Method

## 1) Initialization

### 1-1) Seed instruction following data (IFT 데이터)

사전학습된 LLM을 학습하기 위해 인간이 작성한 일반 지침 따르기 예제 시드 셋 Instruction Fine-Tuning (IFT) 사용

데이터는 (instruction prompt, response) pairs로 구성 (OpenAssistant/oasst1 데이터셋에서 3,200개의 첫 대화 턴만 샘플링하여 사용)

### 1-2) Seed LLM-as-a-Judge instruction following data (EFT 데이터)

IFT 데이터만 사용해도 LLM-as-a-Judge를 학습 가능하지만, 이러한 학습 데이터는 높은 성능을 향상시킬 수 없음.

데이터는 (evaluation instruction prompt, evaluation result response) pairs로 구성

해당 데이터에서 입력 프롬프트는 모델에게 특정 instruction에 대한 주어진 response의 quality를 평가하도록 요청하는 것.

- 제공된 evaluation result response는 CoT 추론과 5점 만점 최종 점수로 구성
- LLM이 여러 측면의 품질을 포괄하는 5 points (relevance, coverage, usefulness, clarity and expertise)을 사용하여 응답을 평가.

Review the user's question and the corresponding response using the additive 5-point scoring system described below. Points are accumulated based on the satisfaction of each criterion:

- Add 1 point if the response is relevant and provides some information related to the user's inquiry, even if it is incomplete or contains some irrelevant content.
- Add another point if the response addresses a substantial portion of the user's question, but does not completely resolve the query or provide a direct answer.
- Award a third point if the response answers the basic elements of the user's question in a useful way, regardless of whether it seems to have been written by an AI Assistant or if it has elements typically found in blogs or search results.
- Grant a fourth point if the response is clearly written from an AI Assistant's perspective, addressing the user's question directly and comprehensively, and is well-organized and helpful, even if there is slight room for improvement in clarity, conciseness or focus.
- Bestow a fifth point for a response that is impeccably tailored to the user's question by an AI Assistant, without extraneous information, reflecting expert knowledge, and demonstrating a high-quality, engaging, and insightful answer.

User: <INSTRUCTION_HERE>

<response><RESPONSE_HERE></response>

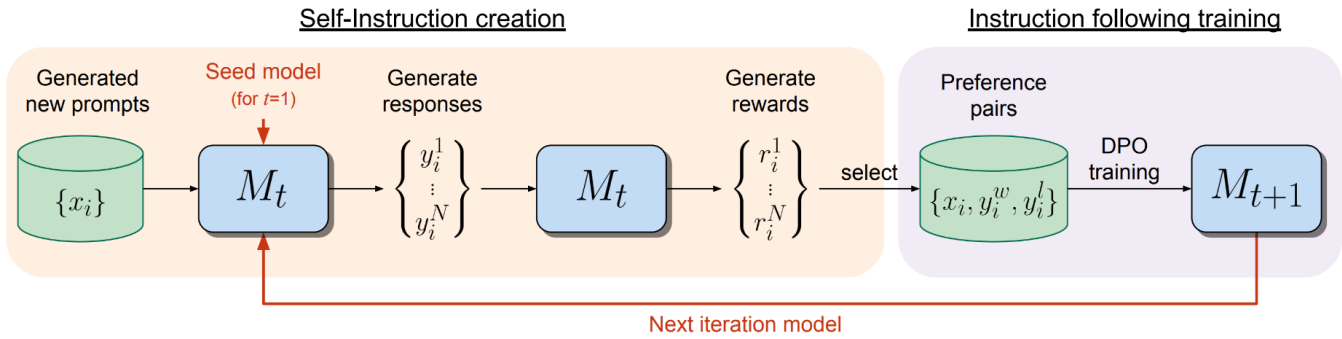
After examining the user's instruction and the response:

- Briefly justify your total score, up to 100 words.
- Conclude with the score using the format: "Score: <total points>"

Remember to assess from the AI Assistant perspective, utilizing web search knowledge as necessary. To evaluate the response in alignment with this additive scoring model, we'll systematically attribute points based on the outlined criteria.

# Method

## 2) Self-Instruction creation




학습된 모델을 사용하여 self training set을 수정하도록 만드는 과정

### 반복 iteration을 위한 추가 학습 데이터를 생성

**1. Generate a new prompt**

소수의 샘플 프롬프트를 사용하여 새로운 프롬프트 x_i를 생성하고, 기존 seed IFT 데이터에서 프롬프트를 샘플링

(생성된 프롬프트와 기존 프롬프트와의 ROUGE-L 유사도가 0.7 미만일 때만 풀에 추가, 너무 길거나 짧은 프롬프트 필터링 등)

**2. Generate candidate responses**

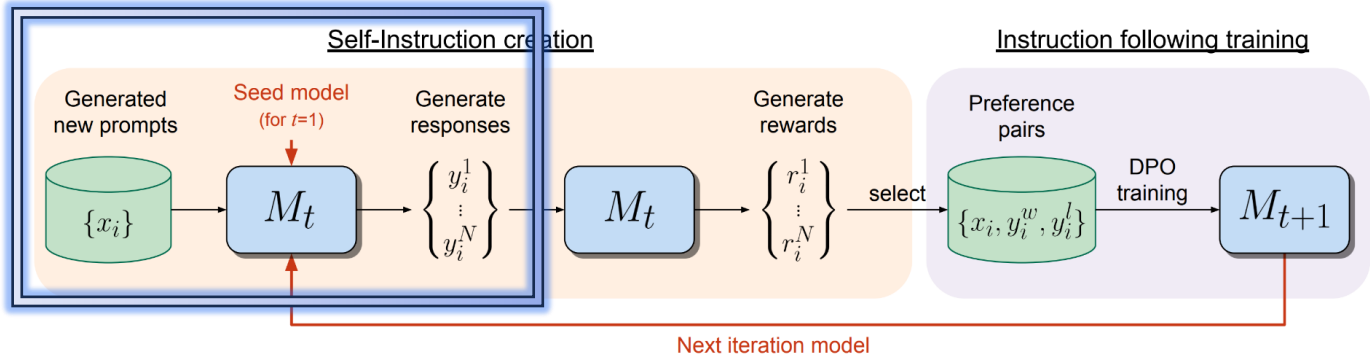
주어진 프롬프트 x_i에 대해 모델에서 샘플링하여 N개의 다양한 후보 응답 {y_1, . . . , y_N} 생성

**3. Evaluate candidate responses**

M_t 모델의 LLM-as-a-Judge 능력을 사용하여 자체 후보 응답을 평가

# Method

## 2) Self-Instruction creation




학습된 모델을 사용하여 self training set을 수정하도록 만드는 과정

### 반복 iteration을 위한 추가 학습 데이터를 생성

**1. Generate a new prompt**

소수의 샘플 프롬프트를 사용하여 새로운 프롬프트 x_i를 생성하고, 기존 seed IFT 데이터에서 프롬프트를 샘플링

(생성된 프롬프트와 기존 프롬프트와의 ROUGE-L 유사도가 0.7 미만일 때만 풀에 추가, 너무 길거나 짧은 프롬프트 필터링 등)

**2. Generate candidate responses**

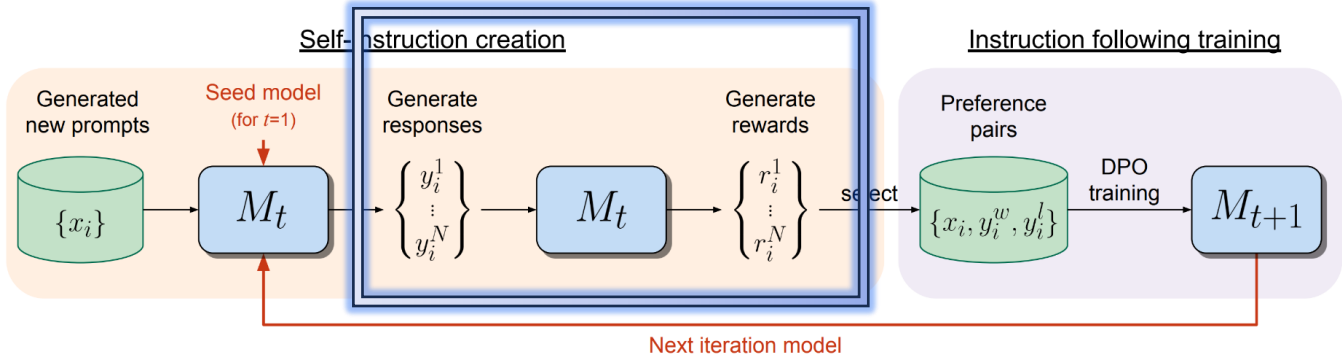
주어진 프롬프트 x_i에 대해 모델에서 샘플링하여 N개의 다양한 후보 응답 {y_1, . . . , y_N} 생성

**3. Evaluate candidate responses**

M_t 모델의 LLM-as-a-Judge 능력을 사용하여 자체 후보 응답을 평가

# Method

## 2) Self-Instruction creation




학습된 모델을 사용하여 self training set을 수정하도록 만드는 과정

### 반복 iteration을 위한 추가 학습 데이터를 생성

**1. Generate a new prompt**

소수의 샘플 프롬프트를 사용하여 새로운 프롬프트 x_i를 생성하고, 기존 seed IFT 데이터에서 프롬프트를 샘플링

(생성된 프롬프트와 기존 프롬프트와의 ROUGE-L 유사도가 0.7 미만일 때만 풀에 추가, 너무 길거나 짧은 프롬프트 필터링 등)

**2. Generate candidate responses**

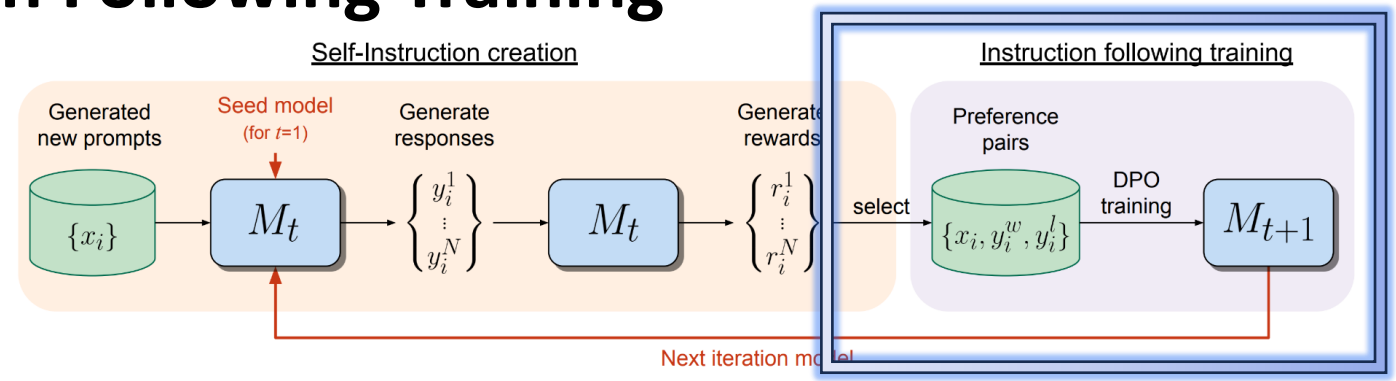
주어진 프롬프트 x_i에 대해 모델에서 샘플링하여 N개의 다양한 후보 응답 {y_1, . . . , y_N} 생성

**3. Evaluate candidate responses**

M_t 모델의 LLM-as-a-Judge 능력을 사용하여 자체 후보 응답을 평가

# Method

## 3) Instruction Following Training




학습 초기 (t=0)에는 seed IFT & seed EFT 데이터로 수행
이후 자체 피드백을 통해 추가 데이터 보강 (t-1에 사용했던 데이터 계속 누적)

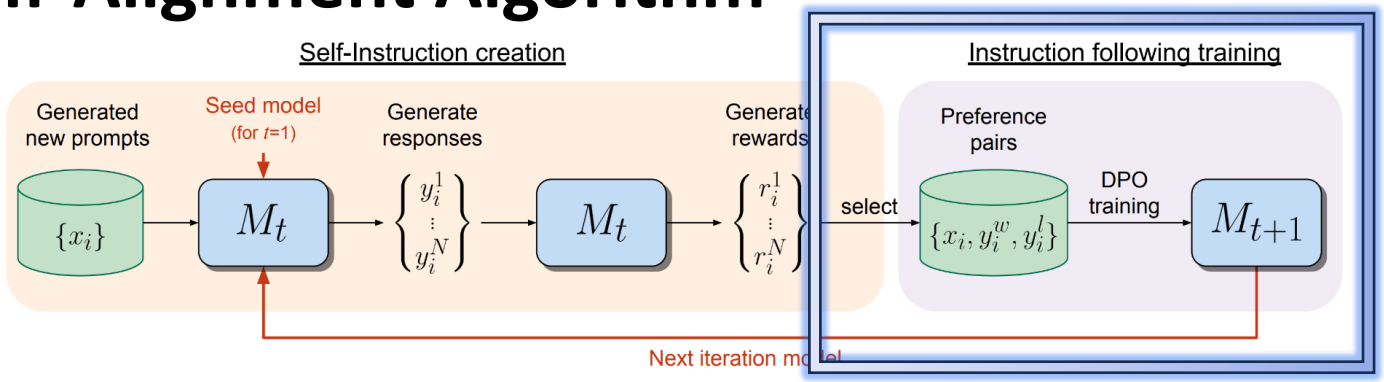
### AI Feedback Training

- Self-instruction 생성 절차를 수행한 후, 학습을 위해 seed 데이터를 추가 예제로 보강

  ➔ 보강된 데이터를 "AI Feedback Training (AIFT)" 데이터로 지칭
- 데이터는 preference pair를 구성 (instruction prompt x_i, winning response y^w_i, losing response y^l_i) 형태
- winning / losing pair를 구성하기 위해 N개의 응답 중에서 최고 점수 받은 응답과 최저 점수 응답을 선택하고, 점수가 동일한 경우 pair를 버림
- DPO로 학습 진행

# Method

## 4) Overall Self-Alignment Algorithm




- **Iterative Training**

M_1, . . . , M_T를 학습 진행. 각 t번째 모델 은t – 1번째 모델이 생성한 보강된 훈련 데이터를 사용

- **각 모델별 학습 데이터**
  - M_0: 미세 조정 없는 사전학습된 LLM (실험에서는 사전학습모델로 Llama 2 70B를 사용)
  - M_1: M_0를 초기화한 후, IFT+EFT 시드 데이터로 SFT를 사용하여 fine-tuning
  - M_2: M_1을 초기화한 후, DPO를 사용하여 AIFT(M_1) 데이터로 훈련
  - M_3: M_2를 초기화한 후, DPO를 사용하여 AIFT(M_2) 데이터로 훈련

# Experiments

## Model & Seed Data

### Model

- Pretrained Llama 2 70B

### Seed Data

#### IFT data

- (instruction prompt, response) pairs로 구성
- OpenAssistant/oasst1 데이터셋에서 3,200개의 첫 대화 턴만 샘플링하여 사용

#### EFT data

- Open Assistant 데이터를 LLM-as-a-Judge 데이터로 생성
- 1,630개 train / 531개 eval set으로 구성

# Experiments

## 평가 지표

- **Instruction Following Ability**
  - GPT-4를 사용 & AlpacaEval 평가 프롬프트
  - Win rate 사용
  - MT-Bench (수학, 코딩, 롤플레이, 작문 등)
- **NLP Benchmark**
  - ARC-Easy, ARC-Challenge, Hellaswag, SIQA, PIQA, GSM8K, MMLU, OBQA, NQ
  - Win rate 사용
  - MT-Bench (수학, 코딩, 롤플레이, 작문 등)

## Training Details

- **SFT**
  - global batch = 16, drop out = 0.1
  - lr = 5.5e-6 ~ 1.1e-6
- **Self-instruction Creation**
  - 새로운 프롬프트 생성 → Llama 2-chat 70B로 8-shot 프롬프팅하여 Self-Instruct 방식으로생성
    - IFT 데이터에서 6개 사용 & 생성된 프롬프트에서 2개 사용
    - Temperatur = 0.6, top-p = 0.9

# Results

## Instruction Following Ability

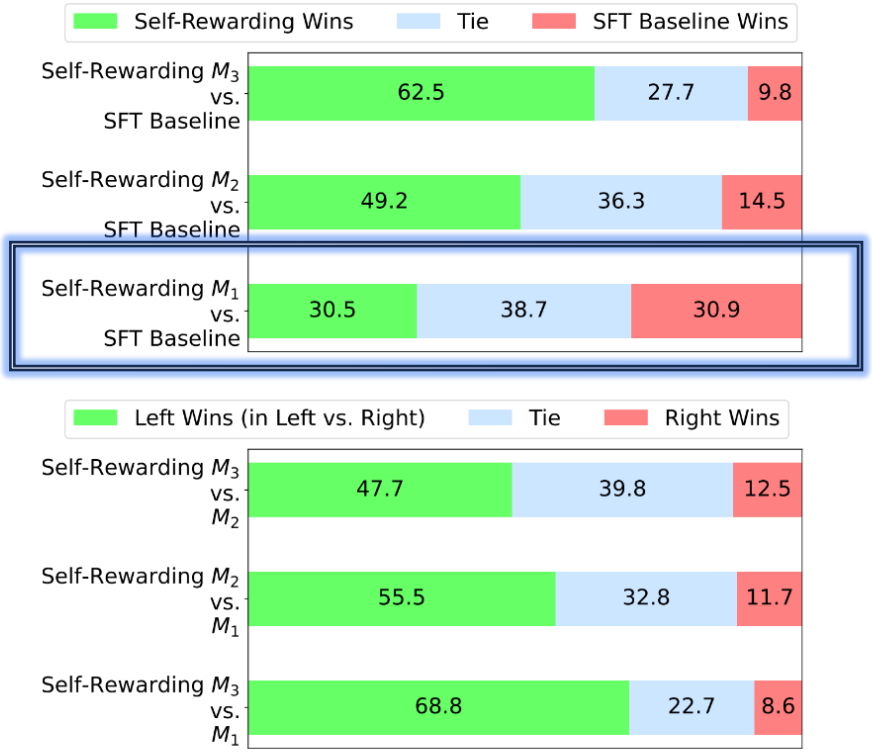



Figure 3: **Instruction following ability improves with Self-Training:** We evaluate our models using head-to-head win rates on diverse prompts using GPT-4. The SFT Baseline is on par with Self-Rewarding Iteration 1 ($M_1$). However, Iteration 2 ($M_2$) outperforms both Iteration 1 ($M_1$) and the SFT Baseline. Iteration 3 ($M_3$) gives further gains over Iteration 2 ($M_2$), outperforming $M_1$, $M_2$ and the SFT Baseline by a large margin.

**EFT+IFT 시드 학습은 IFT 단독 학습과 유사한 성능**

LLM-as-a-Judge Instruction Following (EFT)를 추가해도, IFT 데이터만 사용하는 경우와 비교하면 Instruction Following 능력에 영향을 미치지 않음.

➔ 긍정적인 결과로, 모델의 자체 보상 능력이 다른 기술에 영향을 미치지 않음을 의미. 따라서 IFT+EFT 훈련을 Self-Rewarding 모델의 1단계(M_1)로 사용할 수 있으며, 이후 반복을 진행할 수 있음

# Results

## Instruction Following Ability

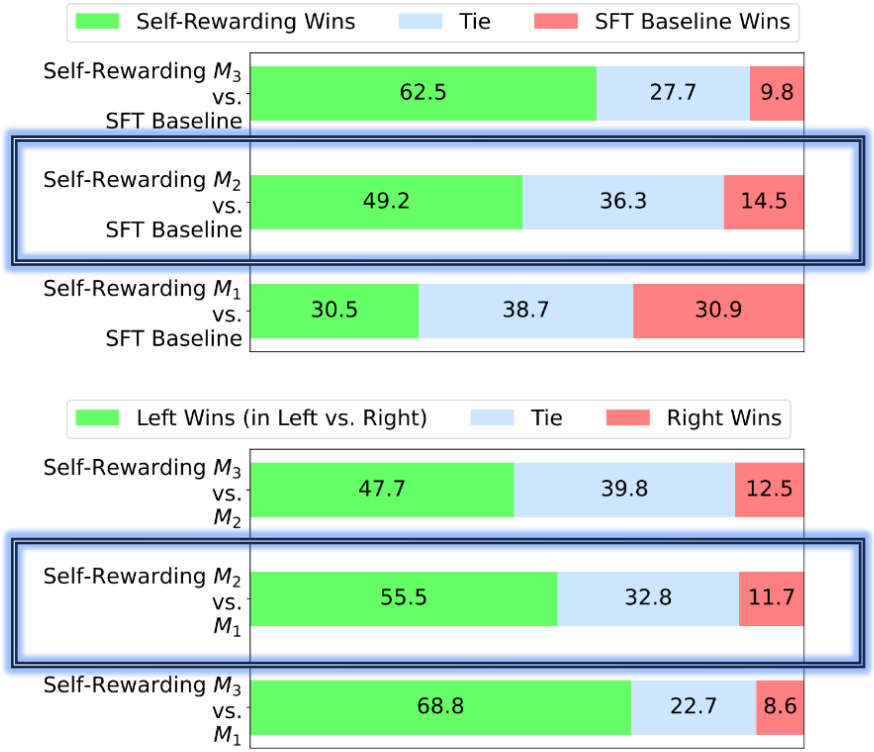



Figure 3: **Instruction following ability improves with Self-Training:** We evaluate our models using head-to-head win rates on diverse prompts using GPT-4. The SFT Baseline is on par with Self-Rewarding Iteration 1 ($M_1$). However, Iteration 2 ($M_2$) outperforms both Iteration 1 ($M_1$) and the SFT Baseline. Iteration 3 ($M_3$) gives further gains over Iteration 2 ($M_2$), outperforming $M_1$, $M_2$ and the SFT Baseline by a large margin.

**EFT+IFT 시드 학습은 IFT 단독 학습과 유사한 성능**

LLM-as-a-Judge Instruction Following (EFT)를 추가해도, IFT 데이터만 사용하는 경우와 비교하면 Instruction Following 능력에 영향을 미치지 않음.

➔ 긍정적인 결과로, 모델의 자체 보상 능력이 다른 기술에 영향을 미치지 않음을 의미. 따라서 IFT+EFT 훈련을 Self-Rewarding 모델의 1단계(M_1)로 사용할 수 있으며, 이후 반복을 진행할 수 있음

**Iteration 2 (M_2)는 Iteration 1 (M_1) 및 SFT 베이스라인보다 개선됨**

Self-Rewarding 학습의 2단계(M_2)는 1단계(M_1)와의 헤드투헤드 평가에서 우수한 Instruction Following 능력. SFT 베이스와 붙어도 개선된 성능

➔ 1단계에서 제공된 AIFT(M_1) 보상 데이터를 사용하여 성능이 크게 향상된다는 것을 의미

# Results

## Instruction Following Ability

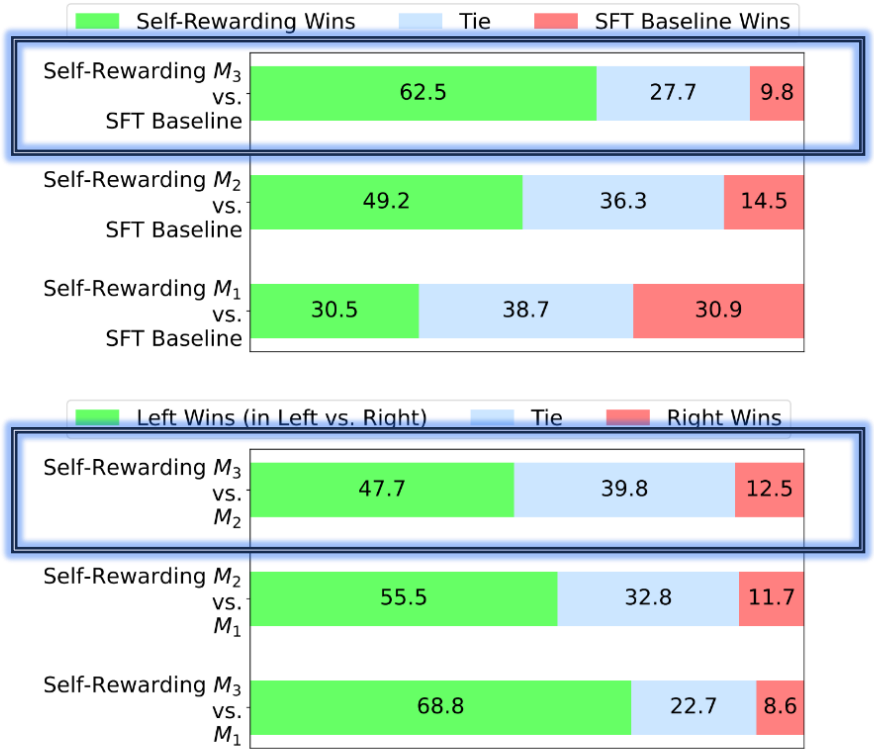



Figure 3: **Instruction following ability improves with Self-Training:** We evaluate our models using head-to-head win rates on diverse prompts using GPT-4. The SFT Baseline is on par with Self-Rewarding Iteration 1 ($M_1$). However, Iteration 2 ($M_2$) outperforms both Iteration 1 ($M_1$) and the SFT Baseline. Iteration 3 ($M_3$) gives further gains over Iteration 2 ($M_2$), outperforming $M_1$, $M_2$ and the SFT Baseline by a large margin.

**EFT+IFT 시드 학습은 IFT 단독 학습과 유사한 성능**

LLM-as-a-Judge Instruction Following (EFT)를 추가해도, IFT 데이터만 사용하는 경우와 비교하면 Instruction Following 능력에 영향을 미치지 않음.

➔ 긍정적인 결과로, 모델의 자체 보상 능력이 다른 기술에 영향을 미치지 않음을 의미. 따라서 IFT+EFT 훈련을 Self-Rewarding 모델의 1단계(M_1)로 사용할 수 있으며, 이후 반복을 진행할 수 있음

**Iteration 2 (M_2)는 Iteration 1 (M_1) 및 SFT 베이스라인보다 개선됨**

Self-Rewarding 학습의 2단계(M_2)는 1단계(M_1)와의 헤드투헤드 평가에서 우수한 Instruction Following 능력. SFT 베이스와 붙어도 개선된 성능

➔ 1단계에서 제공된 AIFT(M_1) 보상 데이터를 사용하여 성능이 크게 향상된다는 것을 의미

**Iteration 3 (M_3)는 Iteration 2 (M_2)보다 개선됨**

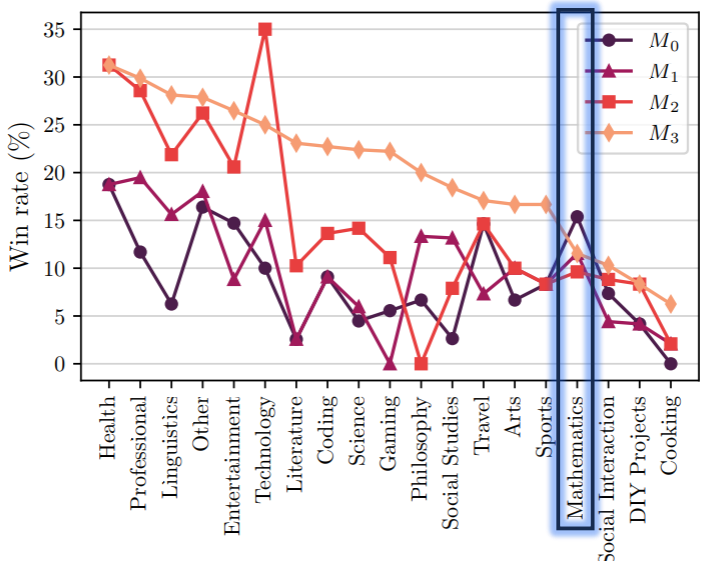
Iteration 3 (M_3)은 Iteration 2 (M_2)와의 평가에서 47.7% 승리, M2는 12.5% 승리로 추가적인 성능 향상이 나타남. SFT 베이스라인 대비 M3의 승률은 62.5% 승리, 9.8% 패배로 증가하여 M2 모델보다 더 자주 승리함

➔ 전반적으로, Iteration 2의 보상 모델에서 제공된 AIFT(M2) 데이터를 사용하여 Iteration 3로의 학습을 통해 큰 성능 향상이 관찰됨

# Results

## Instruction Following Ability

### 잘하는 / 잘못하는 카테고리




➔ Math에서는 성능 손실, Reasoning에서는 아주 약간의 개선

### AlpacaEval 2.0에서도 성능 향상 관측

Table 1: **AlpacaEval 2.0 results** (win rate over GPT-4 Turbo evaluated by GPT-4). Self-Rewarding iterations yield improving win rates. Iteration 3 ($M_3$) outperforms many existing models that use proprietary training data or targets distilled from stronger models.

| Model | Win Rate | Alignment Targets: Distilled | Alignment Targets: Proprietary |
|---|---|---|---|
| Self-Rewarding 70B | | | |
| *Iteration 1* ($M_1$) | 9.94% | | |
| *Iteration 2* ($M_2$) | 15.38% | | |
| *Iteration 3* ($M_3$) | 20.44% | | |
| *Selected models from the leaderboard* | | | |
| GPT-4 0314 | 22.07% | | ✓ |
| Mistral Medium | 21.86% | | ✓ |
| Claude 2 | 17.19% | | ✓ |
| Gemini Pro | 16.85% | | ✓ |
| GPT-4 0613 | 15.76% | | ✓ |
| GPT 3.5 Turbo 0613 | 14.13% | | ✓ |
| LLaMA2 Chat 70B | 13.87% | | ✓ |
| Vicuna 33B v1.3 | 12.71% | ✓ | |
| Humpback LLaMa2 70B | 10.12% | | |
| Guanaco 65B | 6.86% | | |
| Davinci001 | 2.76% | | ✓ |
| Alpaca 7B | 2.59% | ✓ | |

# Results

## Instruction Following Ability

### NLP Benchmarks

Table 3: **NLP Benchmarks.** Self-Rewarding models mostly tend to maintain performance compared to the Llama 2 70B base model and the SFT Baseline, despite being fine-tuned on very different instruction-following prompts.

| | ARC (↑) challenge | HellaSwag (↑) | GSM8K (↑) | MMLU (↑) | NQ (↑) |
|---|---|---|---|---|---|
| Llama 2 | 57.40 | 85.30 | 56.80 | 68.90 | 25.30 |
| SFT Baseline | 55.97 | 85.17 | 50.72 | 69.76 | 34.35 |
| $M_1$ | 57.51 | 84.99 | 60.27 | 69.34 | 35.48 |
| $M_2$ | 54.51 | 84.27 | 59.29 | 69.31 | 33.07 |
| $M_3$ | 53.13 | 83.29 | 57.70 | 69.37 | 31.86 |

➔ **Open Assistant 프롬프트를 기반으로 하고 있어, NLP benchmark 성능이 떨어져야 하지만, 대체로 유지함**
(RLHF 이후에 NLP benchamrk 성능 저하 된다는 이전 연구 인용, **alignment tax**)

### MT-Bench (9 Tasks)

Table 2: **MT-Bench Results** (on a scale of 10). Self-Rewarding iterations yield improving scores across various categories. Math, code & reasoning performance and iteration gains are smaller than for other categories, likely due to the makeup of the Open Assistant seed data we use.

| | Overall Score | Math, Code & Reasoning | Humanities, Extraction, STEM, Roleplay & Writing |
|---|---|---|---|
| SFT Baseline | 6.85 | 3.93 | 8.60 |
| $M_1$ | 6.78 | 3.83 | 8.55 |
| $M_2$ | 7.01 | 4.05 | 8.79 |
| $M_3$ | 7.25 | 4.17 | 9.10 |

➔ **Math & Reasoning에서는 아주 약간의 개선**
**& 싱글턴 데이터를 늘렸는데, 멀티턴에서도 개선**

### Human Evaluation

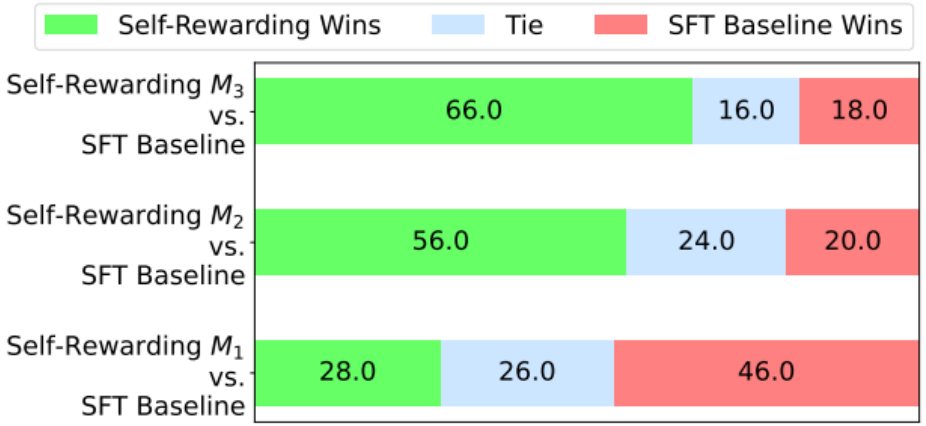



Figure 5: **Human evaluation results.** Iterations of Self-Rewarding ($M_1$, $M_2$ and $M_3$) provide progressively better head-to-head win rates compared to the SFT baseline, in agreement with the automatic evaluation results.

# Conclusion & Limitations

## Self-Rewarding Language Models

➔ 이전 Self-Alignment with Instruction Backtranslation 연구는 SFT 반복적 학습에 대해 연구함.
해당 연구는, DPO 반복적 학습에 대해 연구함

Reasoning, Math에 성능 감소 및 특정 task에서 과도한 길이의 응답을 생성하는 경향이 있음

**+ 멀티턴 성능을 싱글턴 증강으로 개선하기 때문에, 큰 성능 향상이 없는 것으로 보임**

# Thank you

# Q&A