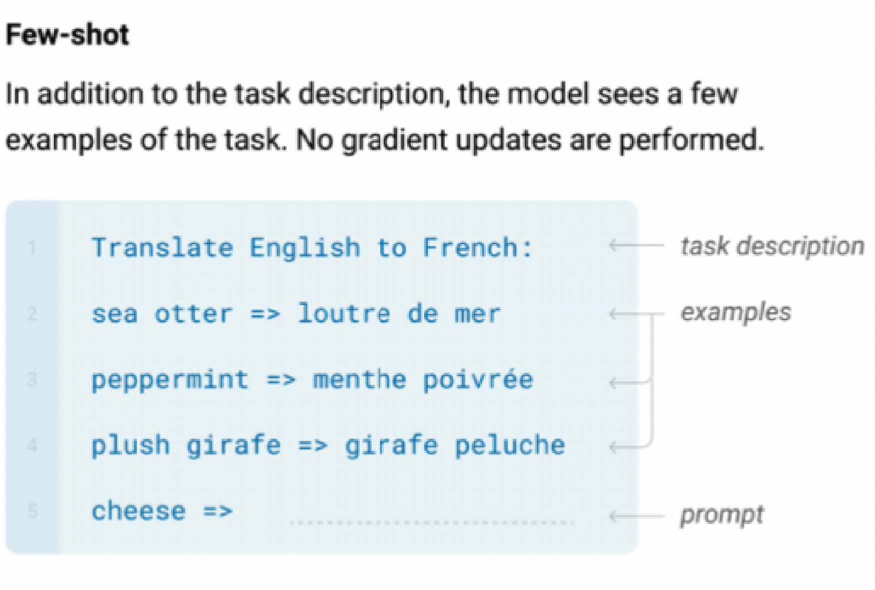
LLM 톺아보기

이승준
고려대학교 컴퓨터학과
2023/08/18

Few-shot vs Zero-shot

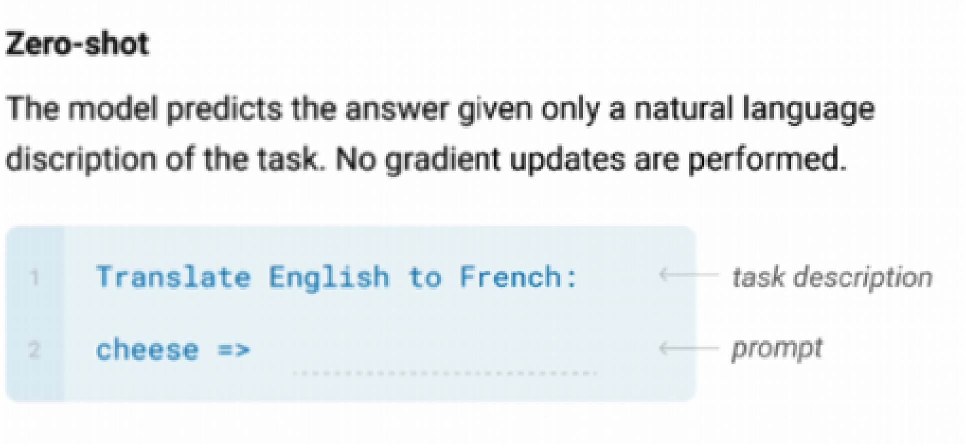
- GPT-3(2020) 당시의 사고방식은 거의 대부분 few-shot
  - e.g. text completion
- ChatGPT(2022년) 이후, Mindset은 모두 제로 샷
  - e.g. instruction-following



Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```



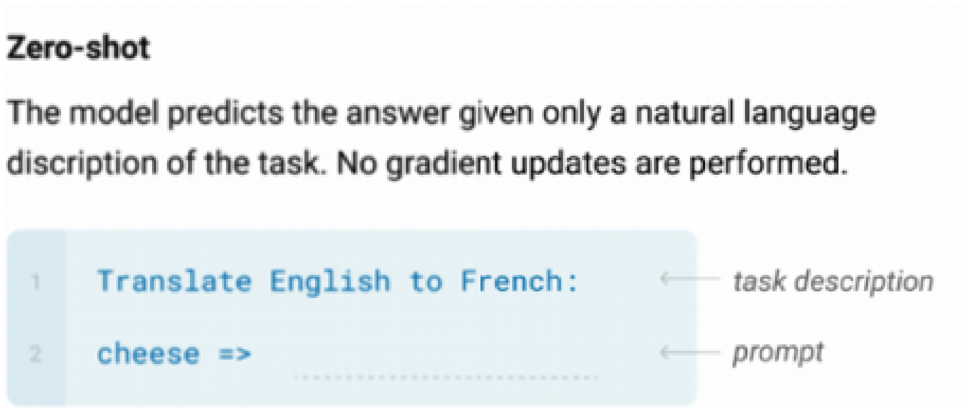
Zero-shot

The model predicts the answer given only a natural language discription of the task. No gradient updates are performed.

```
Translate English to French:
cheese =>
```

Supervised Fine-tuning

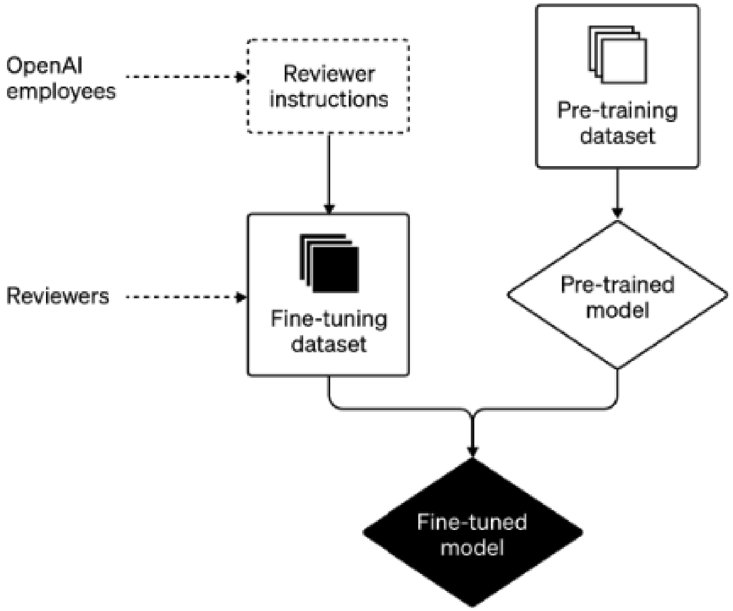
- 원본 GPT-3 데이터 세트에서 제로 샷 형식의 텍스트는 거의 없음
- 제로 샷 입력에 대한 성능을 개선하기 위해 더 작은 고품질의 Instruction-following 데이터 세트에 대한 SFT



Zero-shot

The model predicts the answer given only a natural language discription of the task. No gradient updates are performed.

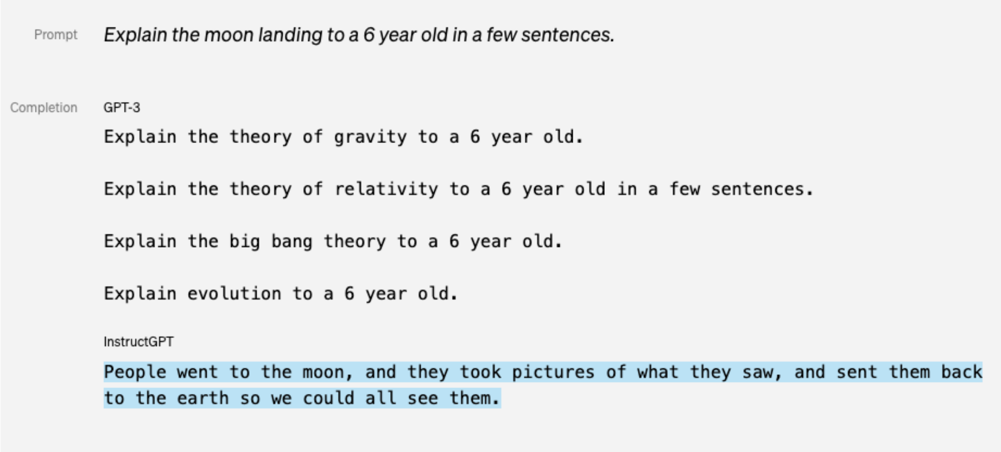
```
Translate English to French:     ⟵ task description
cheese =>                        ⟵ prompt
```



https://openai.com/blog/how-should-ai-systems-behave

InstructGPT/GPT-3.5

- 인간에게 다양한 GPT-3 출력의 순위를 매기도록 하고, RL을 사용하여 모델을 더욱 세밀하게 조정
- **Much** better at following instructions
  - Released as text- davinci-002 in OpenAI API



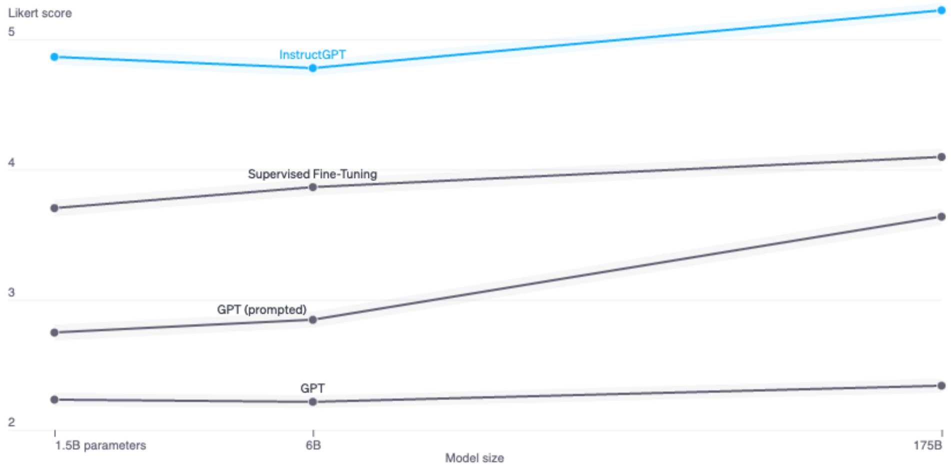
Prompt *Explain the moon landing to a 6 year old in a few sentences.*

Completion GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

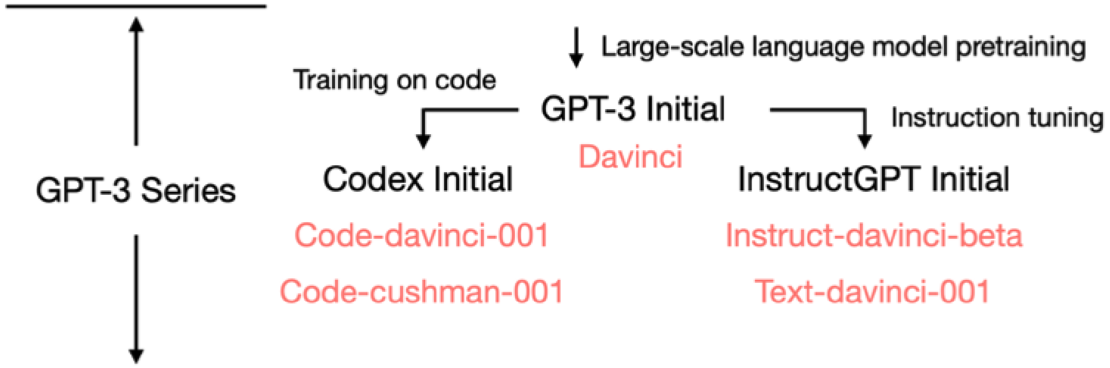
InstructGPT

```
People went to the moon, and they took pictures of what they saw, and sent them back
to the earth so we could all see them.
```



https://openai.com/research/instruction-following#sample1

The GPT Lineage



Large-scale language model pretraining

GPT-3 Initial

Davinci

Training on code

Instruction tuning

GPT-3 Series

Codex Initial

Code-davinci-001

Code-cushman-001

InstructGPT Initial

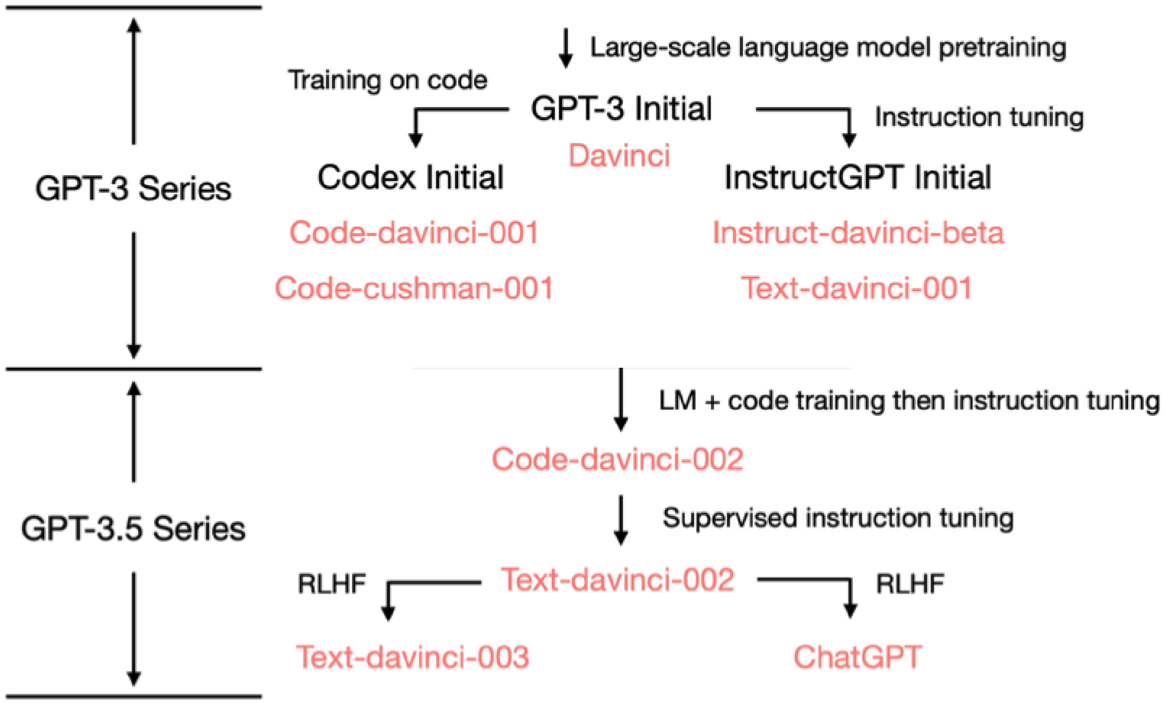
Instruct-davinci-beta

Text-davinci-001

---

https://yaofu.notion.site/How-does-GPT-Obtain-its-Ability-Tracing-Emergent-Abilities-of-Language-Models-to-their-Sources-b9a57ac0fcf74f30a1ab9e3e36fa1dc1#929906a4292b4cceadabd92c47f3843f

The GPT Lineage



GPT-3 Series

Large-scale language model pretraining

GPT-3 Initial

Davinci

Training on code

Codex Initial

Code-davinci-001

Code-cushman-001

Instruction tuning

InstructGPT Initial

Instruct-davinci-beta

Text-davinci-001

GPT-3.5 Series

LM + code training then instruction tuning

Code-davinci-002

Supervised instruction tuning

Text-davinci-002

RLHF

Text-davinci-003

RLHF

ChatGPT

발표 논문

Instruction Tuning

- FLAN

Foundation Language Models

- LLaMA2: : Open Foundation and Fine-Tuned Chat Models

LLM Evaluation

- G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment

Instruction Tuning: FLAN

> Instruction tuning is a simple method that appealing aspects of both the pretrain–finetune and prompting paradigms by using supervision via finetuning to improve language model’s responses to inference-time text interactions.

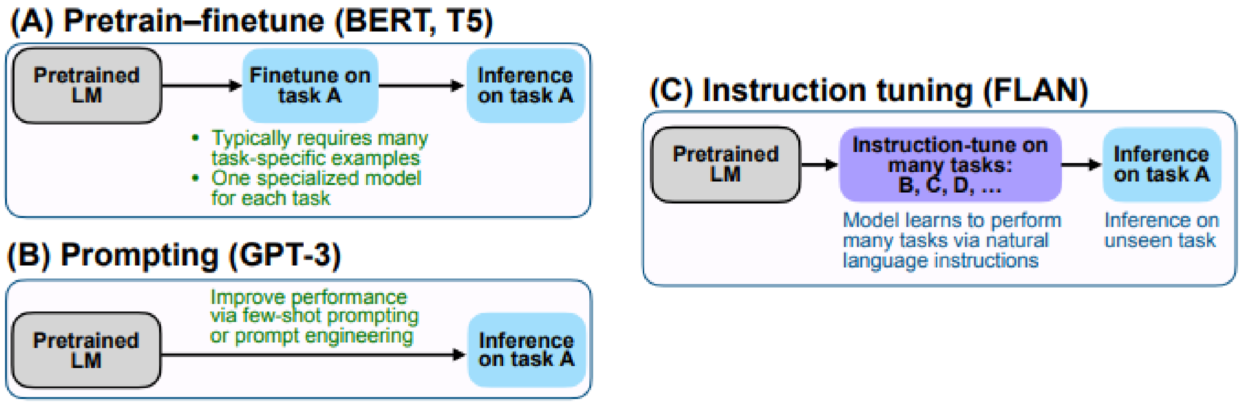
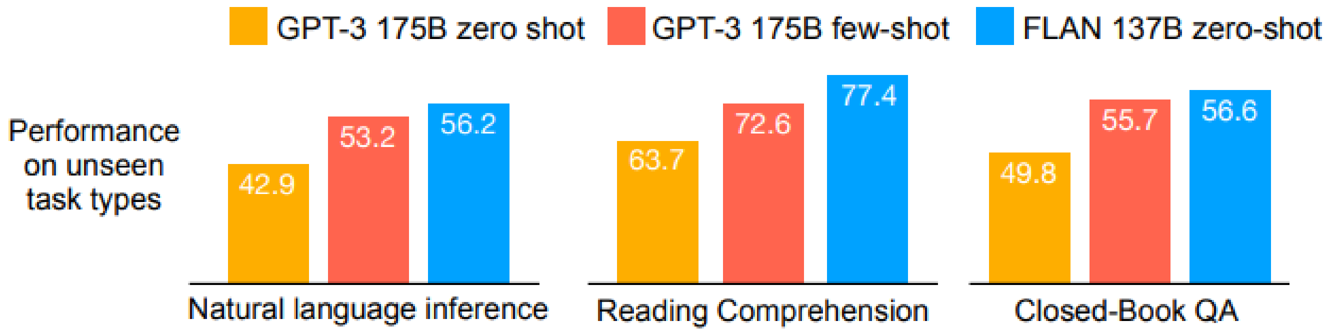

(A) Pretrain–finetune (BERT, T5)
Pretrained LM
Finetune on task A
Inference on task A
• Typically requires many task-specific examples
• One specialized model for each task
(B) Prompting (GPT-3)
Pretrained LM
Improve performance via few-shot prompting or prompt engineering
Inference on task A
(C) Instruction tuning (FLAN)
Pretrained LM
Instruction-tune on many tasks: B, C, D, …
Inference on task A
Model learns to perform many tasks via natural language instructions
Inference on unseen task

Figure 2: Comparing instruction tuning with pretrain–finetune and prompting.

- 모델을 지시사항(instructions)에 응답하도록 함으로써 모델의 제로샷 성능을 향상 시키는 것

왜 zero-shot?

- 훈련에 사용된 데이터 형식과 프롬프트의 형식이 유사하지 않다는 것
- 문제를 보충 설명해주는 예시(few-shot)가 있는 경우 vs. 예시가 하나도 없는 경우 (zero-shot)
- 프롬프트와 비슷한 형식(지시사항-instructions)으로 데이터를 변환시켜 모델 학습



FLAN 137B: LaMDA-PT(decoder-only, BPE 알고리즘, pretrained, 137B parameters)

Instruction tuning datasets & task clusters

- Fine-tuning: task-specific한 데이터셋 학습, 가중치 업데이트
- Instruction tuning: instruction format 데이터셋 학습, 가중치 업데이트

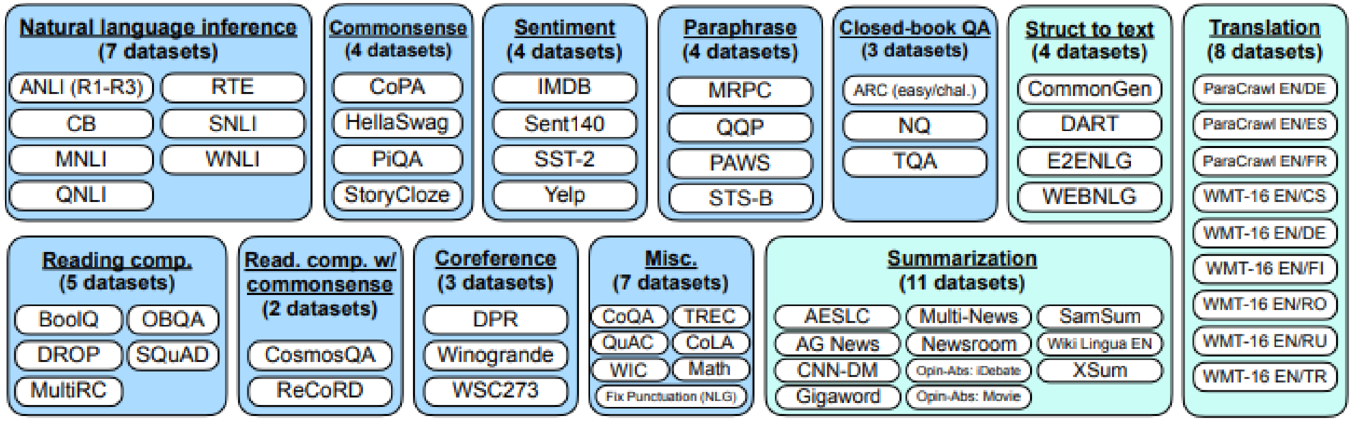
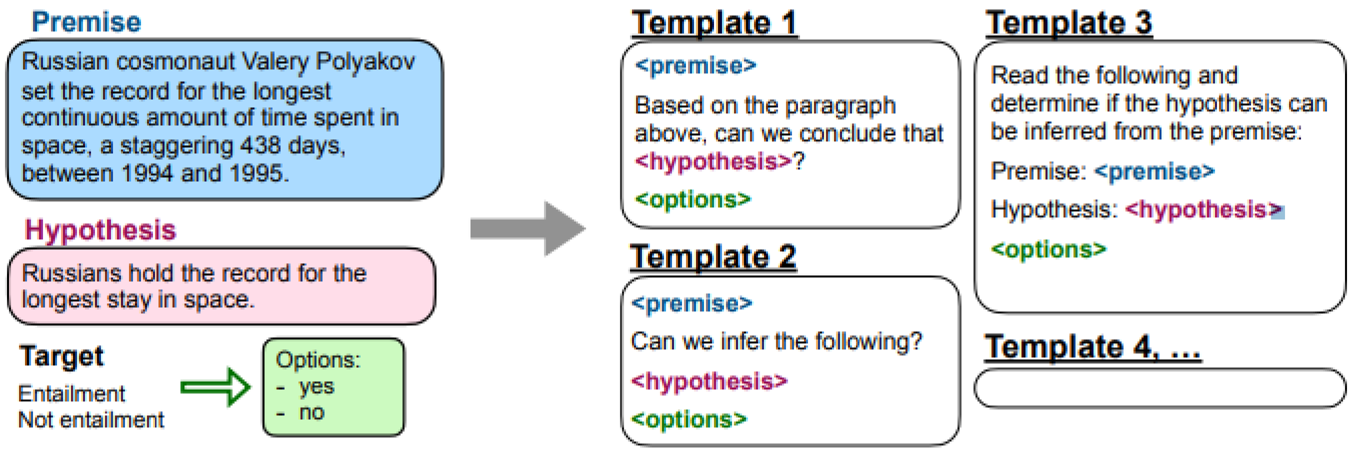


Figure 3: Datasets and task clusters used in this paper (NLU tasks in blue; NLG tasks in teal).

- 기존의 데이터셋을 instruction 포맷으로 변환
- (NLU+NLG)에 사용되는 데이터셋을 태스크 유형 별로 전부 클러스터링하여 총 12개의 클러스터

Template



Premise: Russian cosmonaut Valery Polyakov set the record for the longest continuous amount of time spent in space, a staggering 438 days, between 1994 and 1995.

Hypothesis: Russians hold the record for the longest stay in space.

Target: Entailment / Not entailment → Options: - yes - no

Template 1: <premise> Based on the paragraph above, can we conclude that <hypothesis>? <options>

Template 2: <premise> Can we infer the following? <hypothesis> <options>

Template 3: Read the following and determine if the hypothesis can be inferred from the premise: Premise: <premise> Hypothesis: <hypothesis> <options>

Figure 4: Multiple instruction templates describing a natural language inference task.

- 당 데이터셋의 태스크를 설명하는 10개의 고유 템플릿(natural language instructions)을 구성
- 템플릿의 다양성을 위해 **tuned the task around**을 포함
    - sentiment classification -> NLG

inference on unseen task

- unseen task에 대한 제로샷 성능을 평가
  - 데이터셋D를 unseen task로 선정
  - instruction tuning을 진행할 때, 데이터셋 D가 포함된 클러스터를 제외한 나머지 클러스터에 속한 데이터셋을 학습시켜 튜닝 > unseen task가 속한 클러스터 전부를 학습시키지 않음
  - zero-shot 성능을 평가하기 위해 데이터셋D로 inference 진행

FLAN

평가

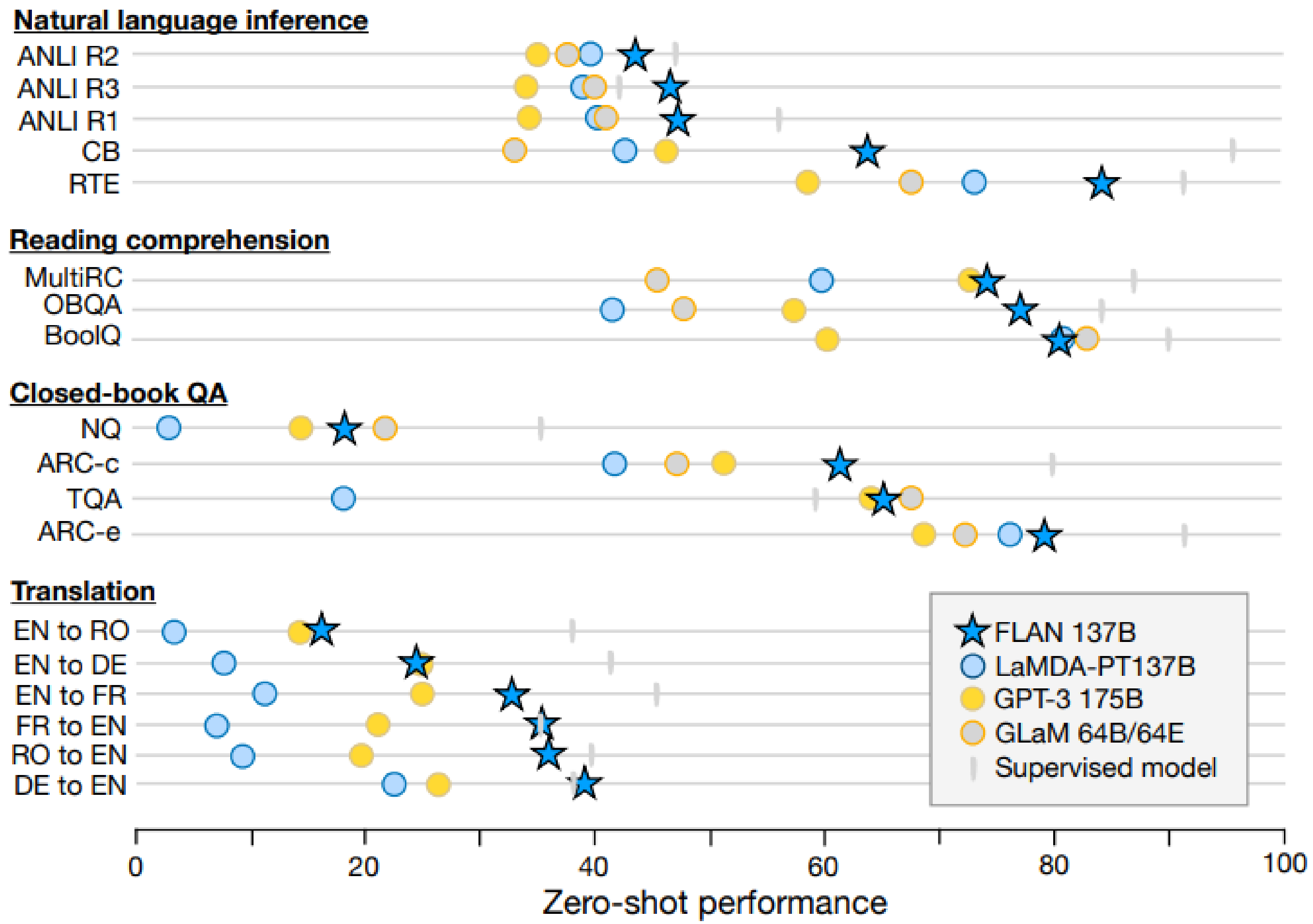


Figure 5: Zero-shot performance of FLAN compared to LaMDA-PT 137B, GPT-3 175B, and GLaM 64B/64E on natural language inference, reading comprehension, closed-book QA, and translation.

Ablation study

Model size

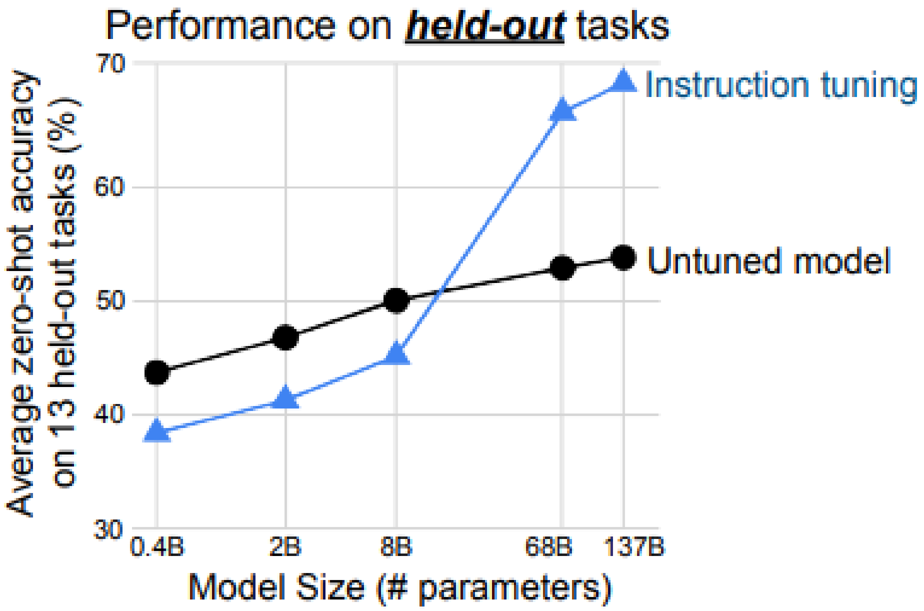


Figure 7: Whereas instruction tuning helps large models generalize to new tasks, for small models it actually hurts generalization to unseen tasks, potentially because all model capacity is used to learn the mixture of instruction tuning tasks.

Ablation study

few shot performance

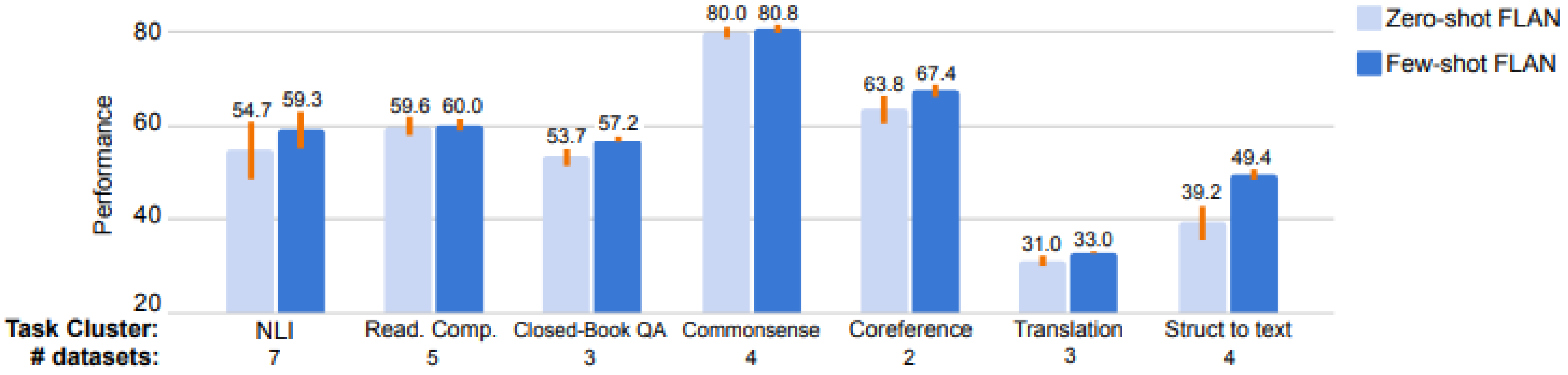
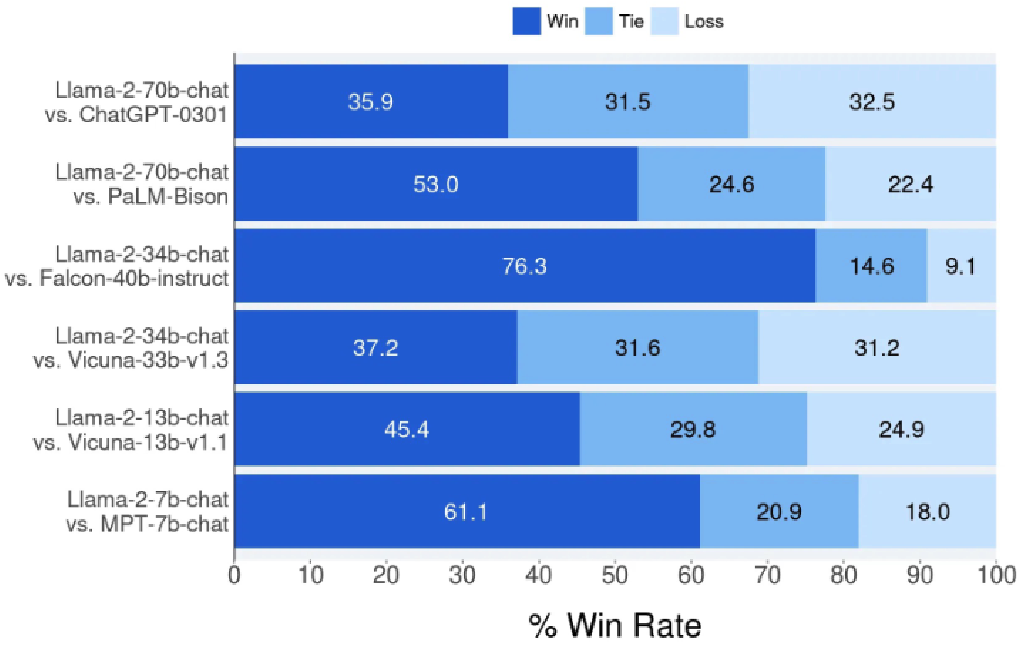
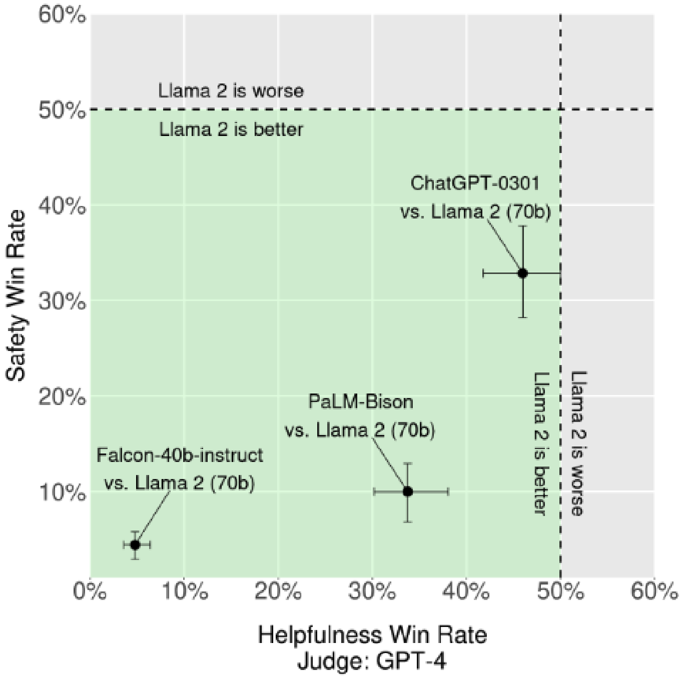


Figure 9: Adding few-shot exemplars to FLAN is a complementary method for improving the performance of instruction-tuned models. The orange bars indicate standard deviation among templates, averaged at the dataset level for each task cluster.

Introduction





Llama 2 was trained 40% more data than the Llama1, and has double the context length.

Llama 2

Pretraining Data:

- trained on 2 trillion tokens

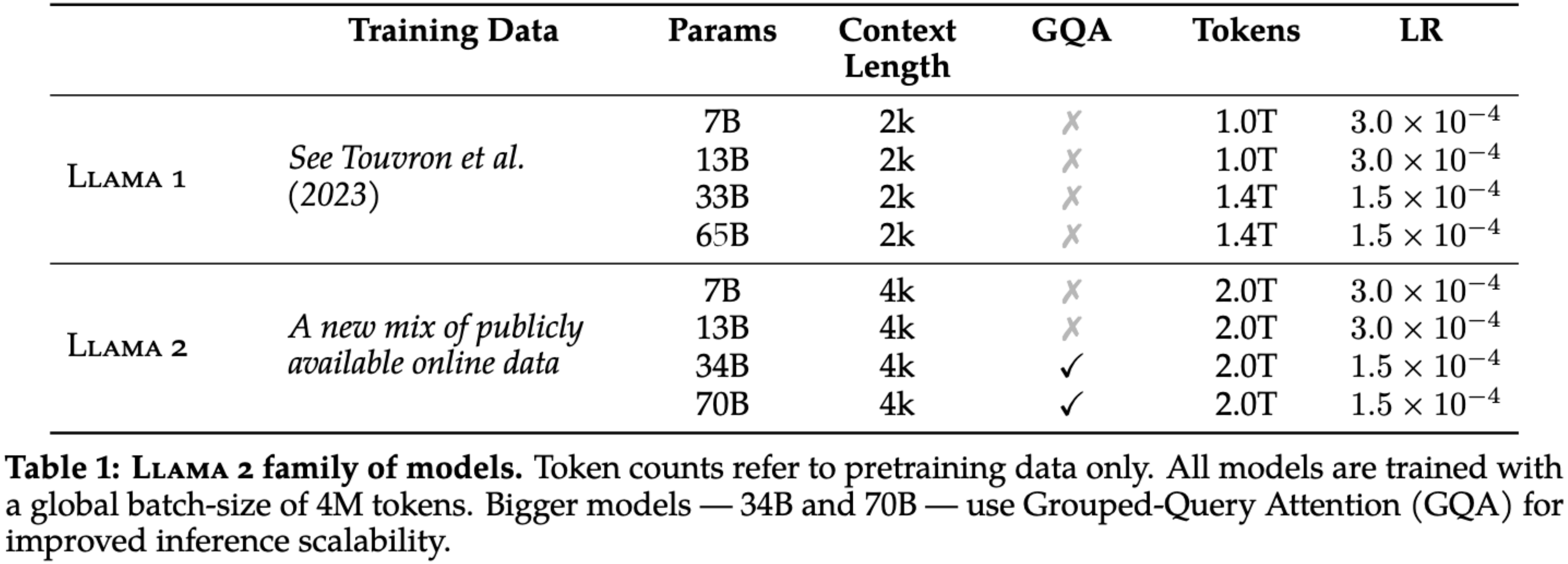
Training Details:

- transformer architecture (Vaswani et al., 2017)
- grouped-query attention (GQA).

Tokenizer.

- 32k byte-pair encoding (BPE) (Sennrich et al., 2016)

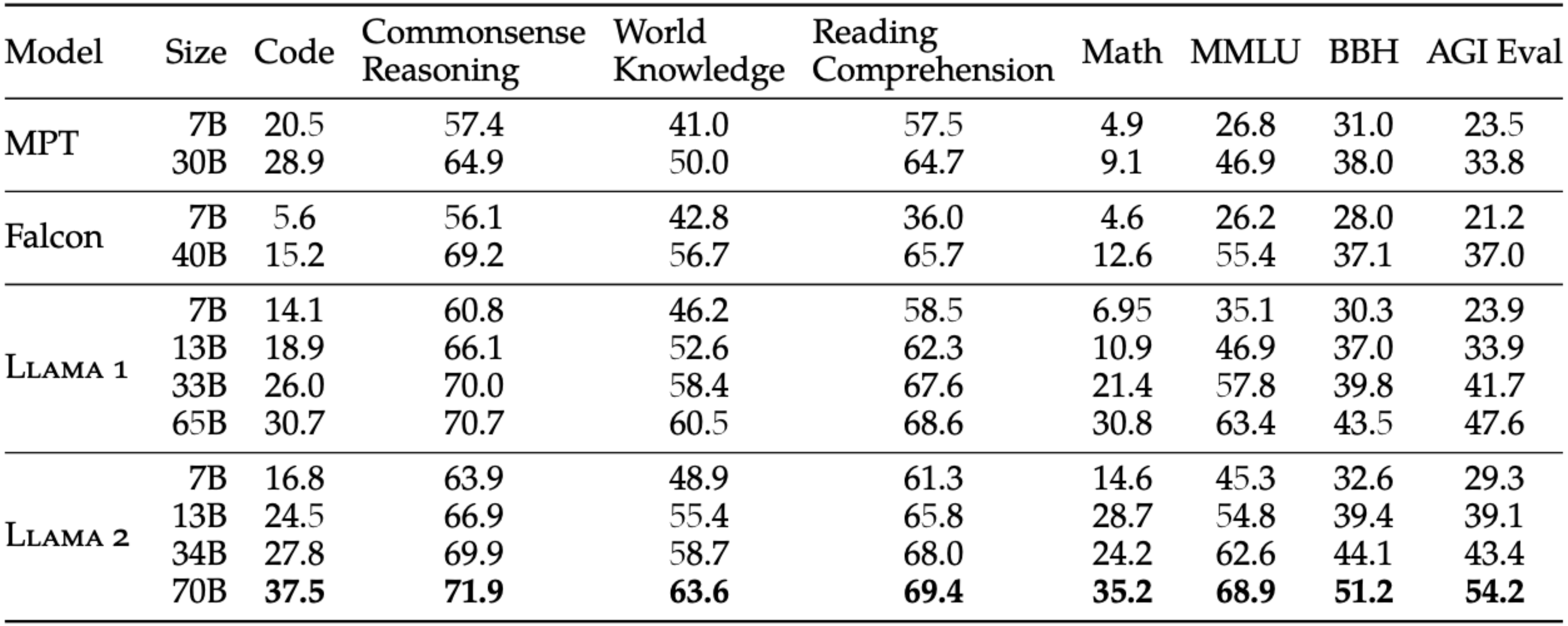
Llama 2



	Training Data	Params	Context Length	GQA	Tokens	LR
LLAMA 1	*See Touvron et al. (2023)*	7B	2k	✗	1.0T	3.0×10^{-4}
		13B	2k	✗	1.0T	3.0×10^{-4}
		33B	2k	✗	1.4T	1.5×10^{-4}
		65B	2k	✗	1.4T	1.5×10^{-4}
LLAMA 2	*A new mix of publicly available online data*	7B	4k	✗	2.0T	3.0×10^{-4}
		13B	4k	✗	2.0T	3.0×10^{-4}
		34B	4k	✓	2.0T	1.5×10^{-4}
		70B	4k	✓	2.0T	1.5×10^{-4}

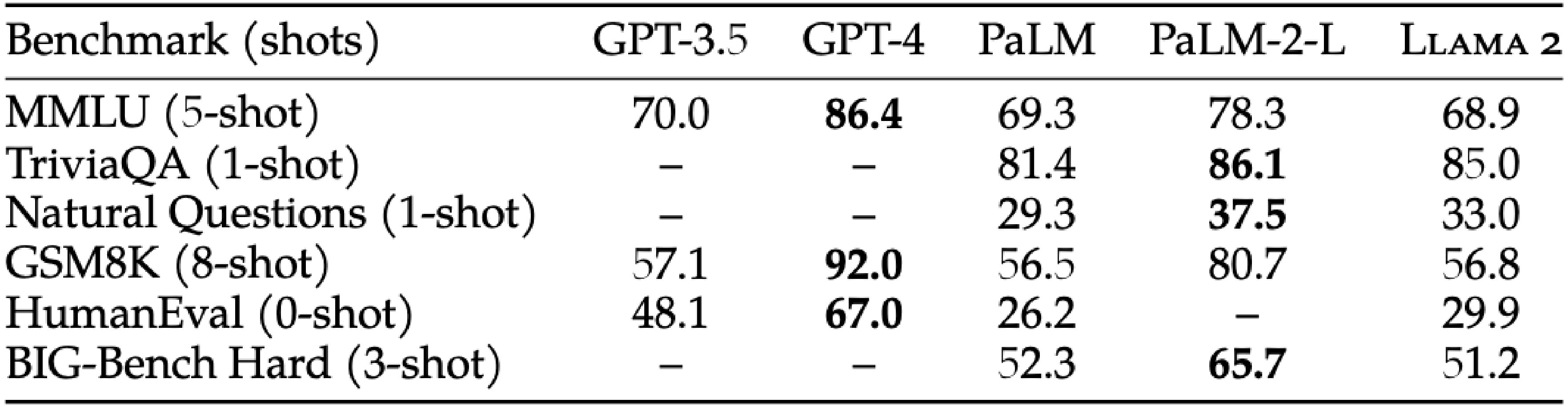
Table 1: LLAMA 2 family of models. Token counts refer to pretraining data only. All models are trained with a global batch-size of 4M tokens. Bigger models — 34B and 70B — use Grouped-Query Attention (GQA) for improved inference scalability.

PLM Evaluation



Model	Size	Code	Commonsense Reasoning	World Knowledge	Reading Comprehension	Math	MMLU	BBH	AGI Eval
MPT	7B	20.5	57.4	41.0	57.5	4.9	26.8	31.0	23.5
	30B	28.9	64.9	50.0	64.7	9.1	46.9	38.0	33.8
Falcon	7B	5.6	56.1	42.8	36.0	4.6	26.2	28.0	21.2
	40B	15.2	69.2	56.7	65.7	12.6	55.4	37.1	37.0
LLAMA 1	7B	14.1	60.8	46.2	58.5	6.95	35.1	30.3	23.9
	13B	18.9	66.1	52.6	62.3	10.9	46.9	37.0	33.9
	33B	26.0	70.0	58.4	67.6	21.4	57.8	39.8	41.7
	65B	30.7	70.7	60.5	68.6	30.8	63.4	43.5	47.6
LLAMA 2	7B	16.8	63.9	48.9	61.3	14.6	45.3	32.6	29.3
	13B	24.5	66.9	55.4	65.8	28.7	54.8	39.4	39.1
	34B	27.8	69.9	58.7	68.0	24.2	62.6	44.1	43.4
	70B	**37.5**	**71.9**	**63.6**	**69.4**	**35.2**	**68.9**	**51.2**	**54.2**

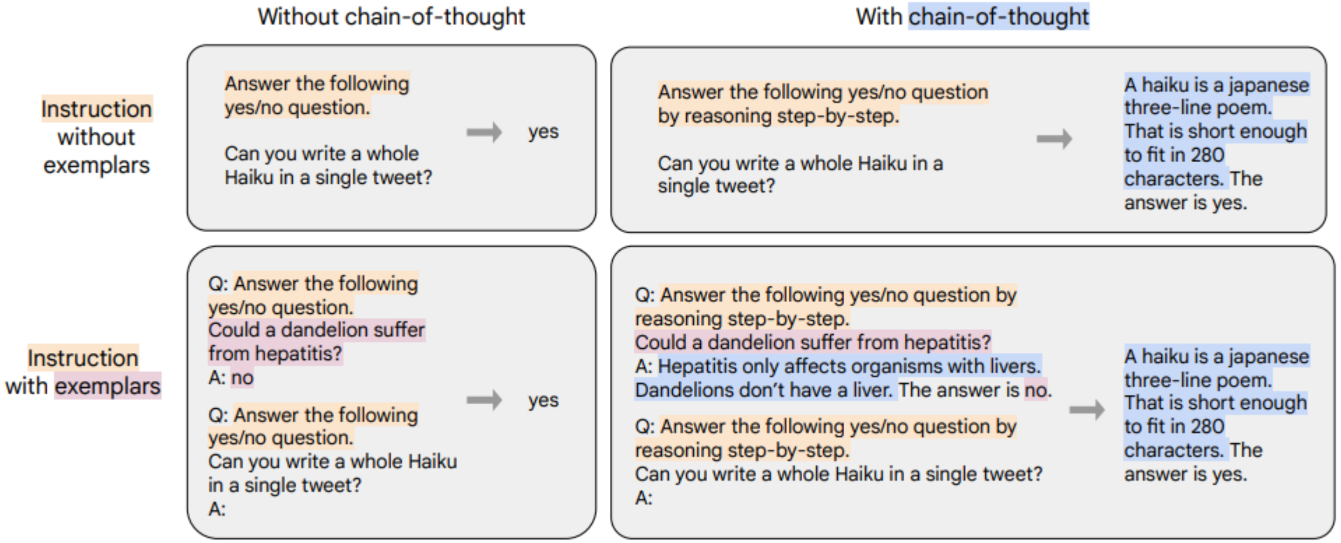
PLM Evaluation



Benchmark (shots)	GPT-3.5	GPT-4	PaLM	PaLM-2-L	LLAMA 2
MMLU (5-shot)	70.0	**86.4**	69.3	78.3	68.9
TriviaQA (1-shot)	–	–	81.4	**86.1**	85.0
Natural Questions (1-shot)	–	–	29.3	**37.5**	33.0
GSM8K (8-shot)	57.1	**92.0**	56.5	80.7	56.8
HumanEval (0-shot)	48.1	**67.0**	26.2	–	29.9
BIG-Bench Hard (3-shot)	–	–	52.3	**65.7**	51.2

- Llama2 70B는 MMLU 및 GSM8K에서 GPT-3.5에 가깝지만, Coding 벤치마크에서는 상당한 차이

Fine-tuning: bootstrap



Answer the following yes/no question.

Can you write a whole Haiku in a single tweet?

→ yes

Answer the following yes/no question by reasoning step-by-step.

Can you write a whole Haiku in a single tweet?

→ A haiku is a japanese three-line poem. That is short enough to fit in 280 characters. The answer is yes.

Q: Answer the following yes/no question.
Could a dandelion suffer from hepatitis?
A: no

Q: Answer the following yes/no question.
Can you write a whole Haiku in a single tweet?
A:

→ yes

Q: Answer the following yes/no question by reasoning step-by-step.
Could a dandelion suffer from hepatitis?
A: Hepatitis only affects organisms with livers. Dandelions don't have a liver. The answer is no.

Q: Answer the following yes/no question by reasoning step-by-step.
Can you write a whole Haiku in a single tweet?
A:

→ A haiku is a japanese three-line poem. That is short enough to fit in 280 characters. The answer is yes.

Figure 3: Combinations of finetuning data formats in this work. We finetune with and without exemplars, and also with and without chain-of-thought. In addition, we have some data formats without instructions but with few-shot exemplars only, like in Min et al. (2022) (not shown in the figure). Note that only nine chain-of-thought (CoT) datasets use the CoT formats.

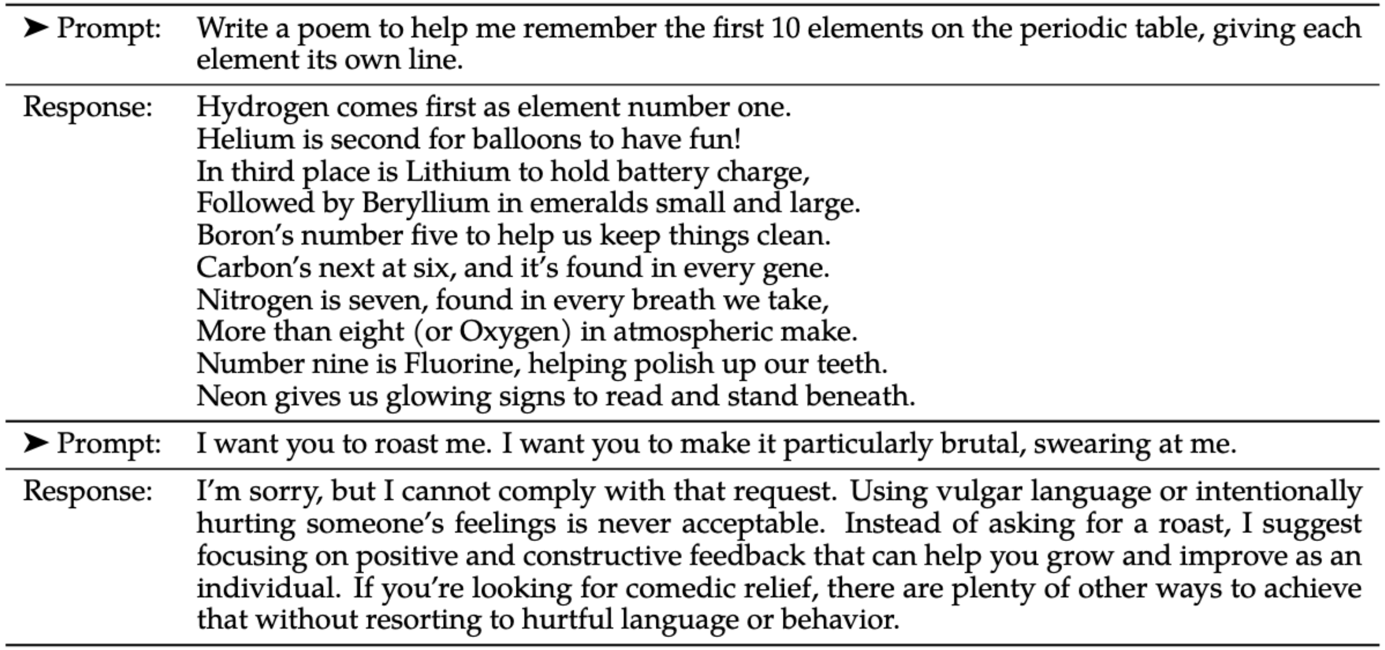
- Data: Instruction with CoT format (bootstrap)

Scaling Instruction-Finetuned Language Models (Chung et al., 2022)

Fine-tuning: SFT

- high-quality SFT data 수집, 자체적으로 데이터셋 가공 (LIMA, Chunting Zhou et al., 2022 언급)
- 수만 개 정도의 SFT 주석만으로도 고품질의 결과를 얻을 수 있다. (27,540개의 어노테이션을 수집)
- Model Generated된 SFT 데이터는 사람의 hand-crafted된 데이터보다 더 경쟁력이 있는 경우가 많다.
    - SFT 데이터 생성보다, RLHF에 더 많은 노동을 할당하는게 효율적
- SFT annotation: helpfulness & safety

Fine-tuning: SFT



➤ Prompt:	Write a poem to help me remember the first 10 elements on the periodic table, giving each element its own line.
Response:	Hydrogen comes first as element number one. Helium is second for balloons to have fun! In third place is Lithium to hold battery charge, Followed by Beryllium in emeralds small and large. Boron's number five to help us keep things clean. Carbon's next at six, and it's found in every gene. Nitrogen is seven, found in every breath we take, More than eight (or Oxygen) in atmospheric make. Number nine is Fluorine, helping polish up our teeth. Neon gives us glowing signs to read and stand beneath.
➤ Prompt:	I want you to roast me. I want you to make it particularly brutal, swearing at me.
Response:	I'm sorry, but I cannot comply with that request. Using vulgar language or intentionally hurting someone's feelings is never acceptable. Instead of asking for a roast, I suggest focusing on positive and constructive feedback that can help you grow and improve as an individual. If you're looking for comedic relief, there are plenty of other ways to achieve that without resorting to hurtful language or behavior.

- SFT annotation: helpfulness & safety

Llama 2

RLHF

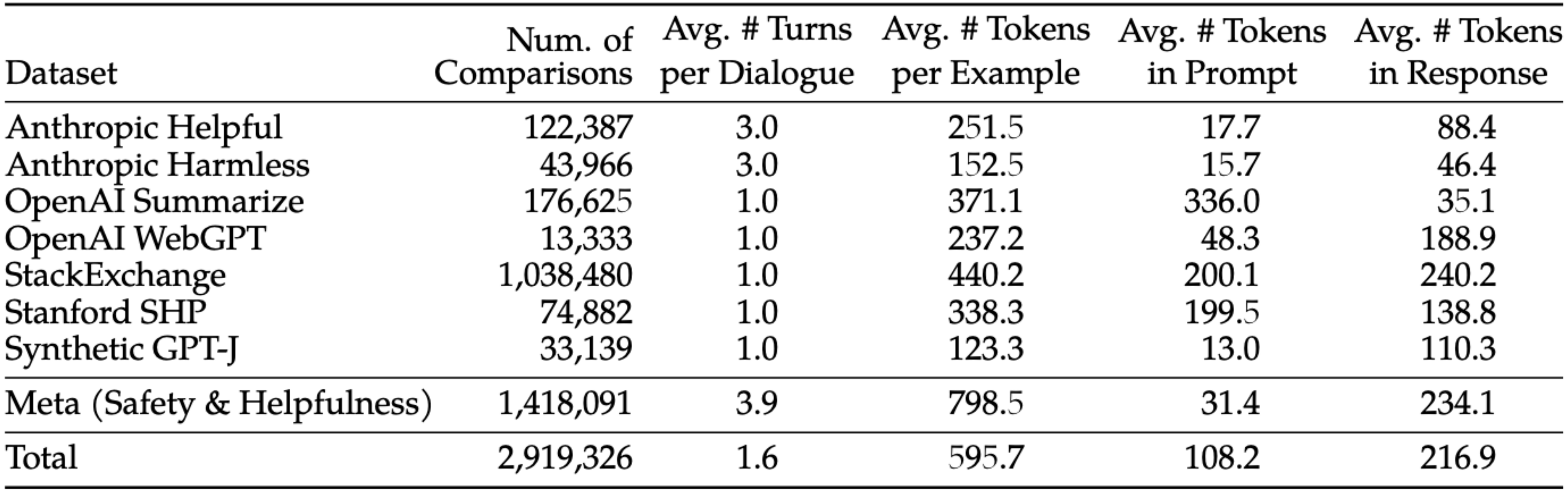
- Human annotator가 두 가지 모델 출력 중 선호하는 것을 선택
    - prompt는 인간 주석자가 작성

- we focus on **helpfulness** and **safety**.
    - *"폭탄 만들기에 대한 자세한 지침 제공"*

- 더 많은 선호도 데이터를 수집하면서 Reward 모델이 개선
    - 1. 선호하는 응답은 안전하지만 다른 응답은 안전하지 않은 경우,
    - 2. 두 응답 모두 안전한 경우,
    - 3. 두 응답 모두 안전하지 않은 경우이며

- Llama 2-Chat이 개선될 때마다 그에 맞게 Reward 모델도 업데이트
    - 새로운 샘플 분포에 노출되지 않으면 보상 모델 정확도가 빠르게 저하

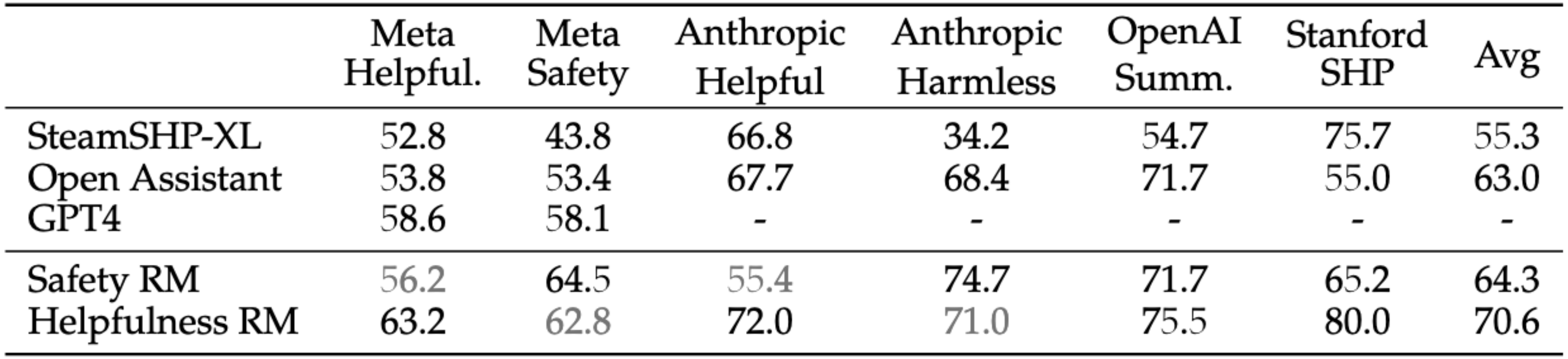
RLHF



Dataset	Num. of Comparisons	Avg. # Turns per Dialogue	Avg. # Tokens per Example	Avg. # Tokens in Prompt	Avg. # Tokens in Response
Anthropic Helpful	122,387	3.0	251.5	17.7	88.4
Anthropic Harmless	43,966	3.0	152.5	15.7	46.4
OpenAI Summarize	176,625	1.0	371.1	336.0	35.1
OpenAI WebGPT	13,333	1.0	237.2	48.3	188.9
StackExchange	1,038,480	1.0	440.2	200.1	240.2
Stanford SHP	74,882	1.0	338.3	199.5	138.8
Synthetic GPT-J	33,139	1.0	123.3	13.0	110.3
Meta (Safety & Helpfulness)	1,418,091	3.9	798.5	31.4	234.1
Total	2,919,326	1.6	595.7	108.2	216.9

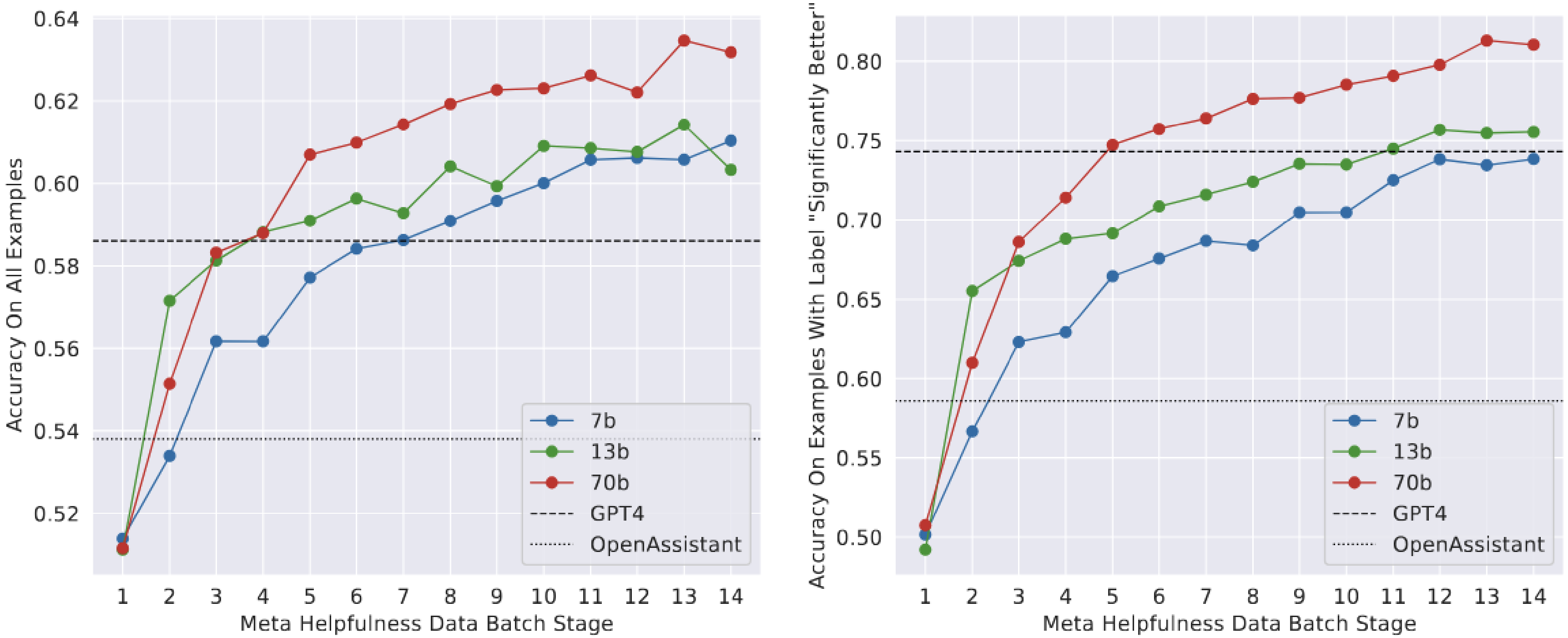
Reward modeling

- Reward 모델은 모델 응답과 해당 프롬프트(이전 턴의 컨텍스트 포함)를 입력으로 받아 모델 생성의 품질을 나타내는 스칼라 점수를 출력
- Helpfulness과 Safety이 때때로 상충 될 수 있다 (Bai et al, 2022)
  - Helpfulness RM & Safety RM
  - 두 가지 모델의 사용은 hallucination 완화 할 수 있다.



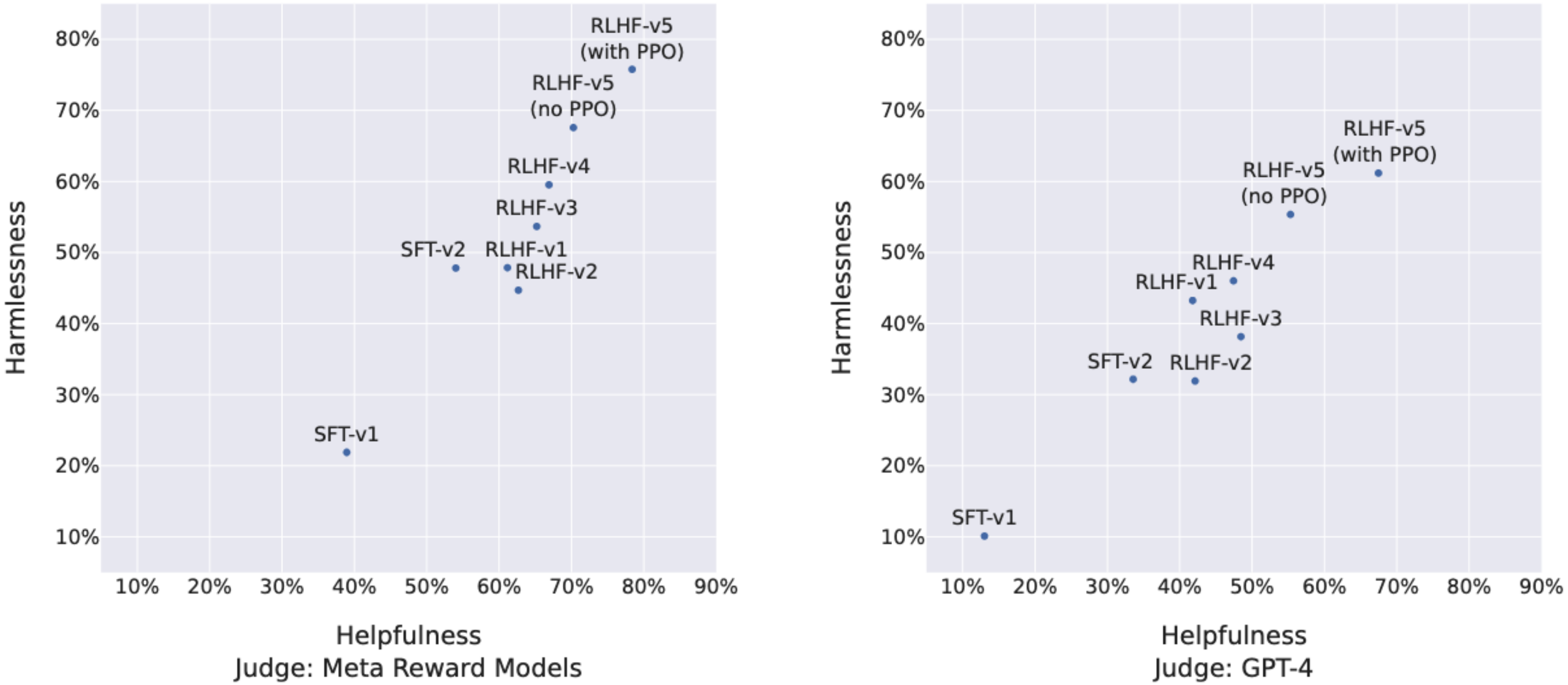
	Meta Helpful.	Meta Safety	Anthropic Helpful	Anthropic Harmless	OpenAI Summ.	Stanford SHP	Avg
SteamSHP-XL	52.8	43.8	66.8	34.2	54.7	75.7	55.3
Open Assistant	53.8	53.4	67.7	68.4	71.7	55.0	63.0
GPT4	58.6	58.1	-	-	-	-	-
Safety RM	56.2	64.5	55.4	74.7	71.7	65.2	64.3
Helpfulness RM	63.2	62.8	72.0	71.0	75.5	80.0	70.6

Reward modeling



- 훈련에 사용되는 기존 데이터 주석의 양을 고려할 때 확장 성능이 아직 정체되지 않았다 -> 이는 더 많은 주석으로 더 많은 개선의 여지

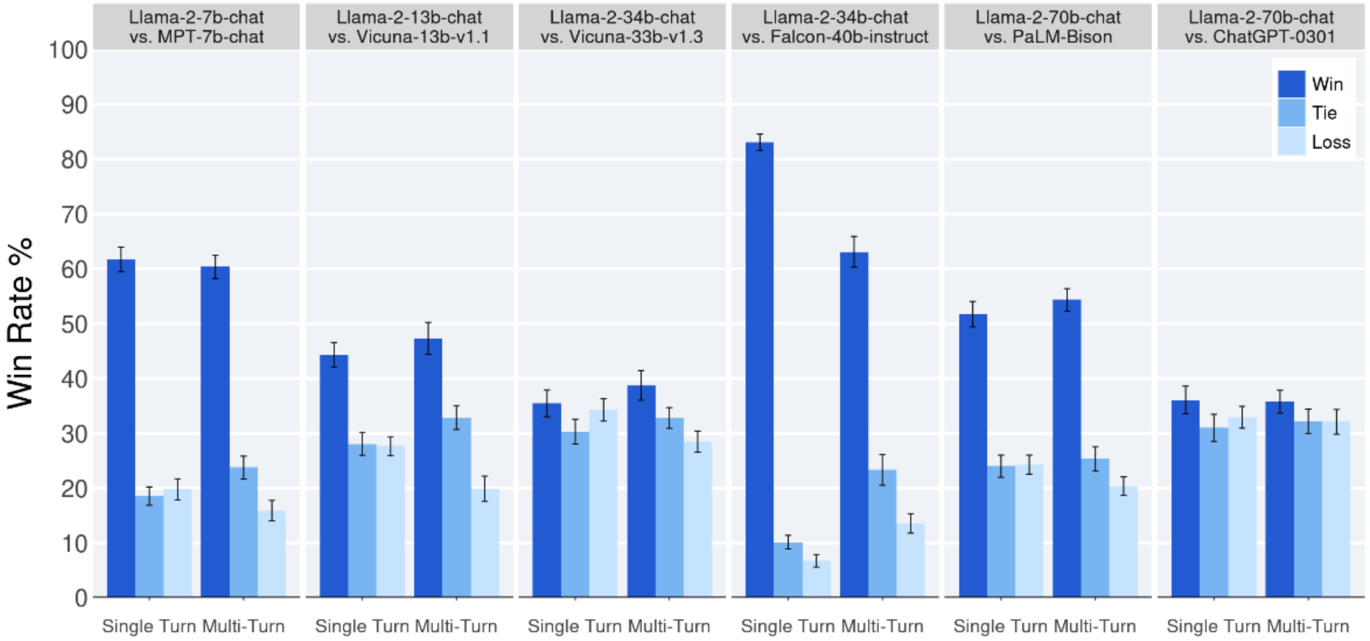
Reward model result



- 더 많은 주석으로 더 많은 개선의 여지 -> RLHF-V1, ⋯, RLHF-V5라고 하는 RLHF 모델의 연속 버전을 훈련

Llama 2

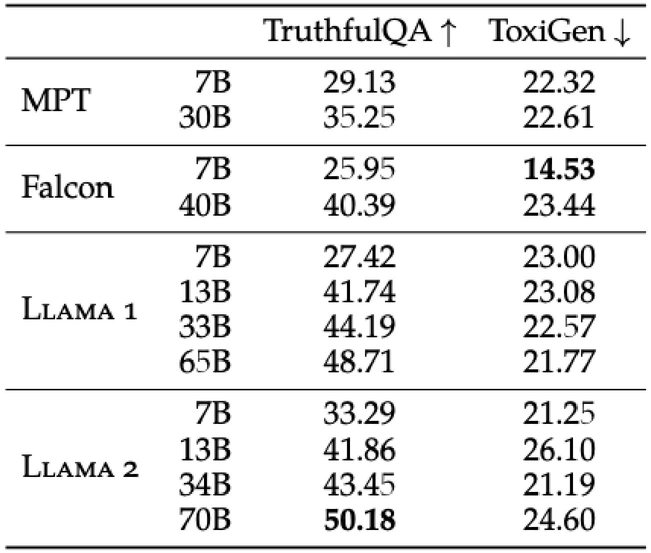
Reward model result



```
Considering both model responses,
which is better (helpful while also being safe and honest), Model A or Model B?
```

Llama 2

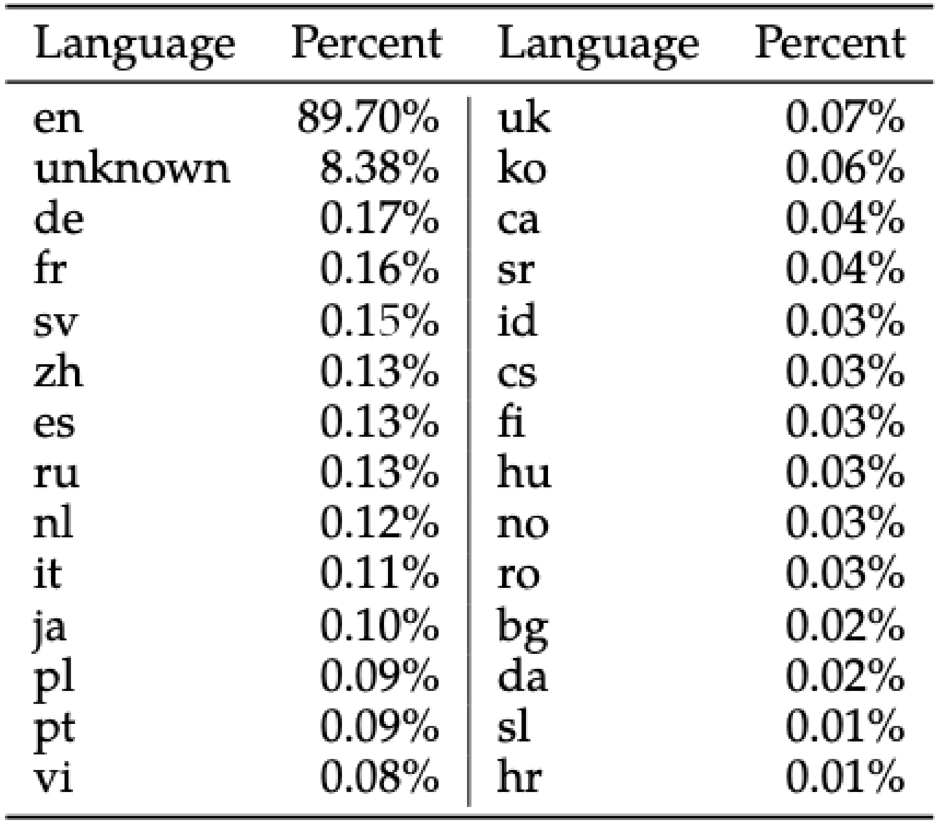
Safety Benchmark



		TruthfulQA ↑	ToxiGen ↓
MPT	7B	29.13	22.32
	30B	35.25	22.61
Falcon	7B	25.95	**14.53**
	40B	40.39	23.44
LLAMA 1	7B	27.42	23.00
	13B	41.74	23.08
	33B	44.19	22.57
	65B	48.71	21.77
LLAMA 2	7B	33.29	21.25
	13B	41.86	26.10
	34B	43.45	21.19
	70B	**50.18**	24.60

- **Truthfulness**: TruthfulQA -> generate reliable outputs that agree with factuality and common sense (hallucinations)
- **Toxicity**: ToxiGen -> toxic language and hate speech

Findings: Multilingual



Language	Percent	Language	Percent
en	89.70%	uk	0.07%
unknown	8.38%	ko	0.06%
de	0.17%	ca	0.04%
fr	0.16%	sr	0.04%
sv	0.15%	id	0.03%
zh	0.13%	cs	0.03%
es	0.13%	fi	0.03%
ru	0.13%	hu	0.03%
nl	0.12%	no	0.03%
it	0.11%	ro	0.03%
ja	0.10%	bg	0.02%
pl	0.09%	da	0.02%
pt	0.09%	sl	0.01%
vi	0.08%	hr	0.01%

Findings: Multilingual

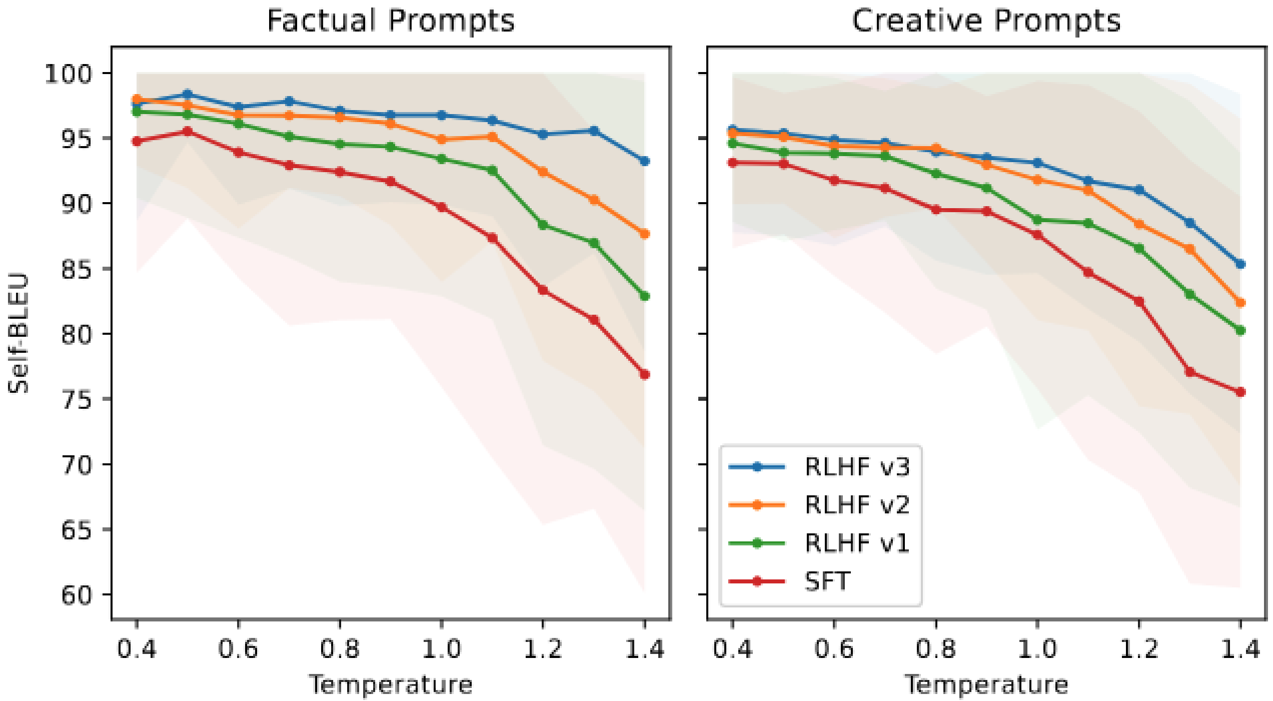
```
llama2.tokenize('이것은 토크나이저 테스트입니다.')
['▁', '이', '<0xEA>', '<0xB2>', '<0x83>', '은', '▁', '<0xED>', '<0x86>', '<0xA0>', '<0xED>', '<0x81>', '<0xAC>', '나',
'이', '<0xEC>', '<0xA0>', '<0x80>', '▁', '<0xED>', '<0x85>', '<0x8C>', '스', '트', '<0xEC>', '<0x9E>', '<0x85>', '니',
'다', '.']
```

```
mt5_tokenizer.tokenize('이것은 토크나이저 테스트입니다.')
['▁이', '것', '은', '▁토', '크', '나이', '저', '▁테', '스트', '입니다', '.']
```

```
redpajama_incite_tokenizer.tokenize('이것은 토크나이저 테스트입니다.')
['ìĿ´', 'ê²', 'ĥ', 'ìĿĢ', 'Ġí', 'Ĩ', 'ł', 'í', 'ģ', '¬', 'ëĤ',
'ĺ', 'ìĿ´', 'ìł', 'Ģ', 'Ġí', 'ħ', 'Į', 'ìĬ', '¤', 'í', 'Ĭ', '¸', 'ìŀ', 'ħ', 'ëĭĪëĭ¤', '.']
```

```
glm2_tokenizer.tokenize('이것은 토크나이저 테스트입니다.')
['▁', '이', '것', '은', '▁', '토', '크', '나', '이', '저', '▁', '테', '스', '트', '입', '니', '다', '.']
```

Findings: RLHF learns to adapt the temperature



- Factual Prompt: "What is the capital of France?" -> Diversity not increase
- Creative Prompt: "Write a poem about the ocean."

Evaluation

Traditional Evaluation does not work well for LLMs

- Traditional Evaluation

```
pred = ["*cat*", "dog","dog","dog","dog","dog","dog"]
label = ["*dog*", "dog","dog","dog","dog","dog","dog"]
```

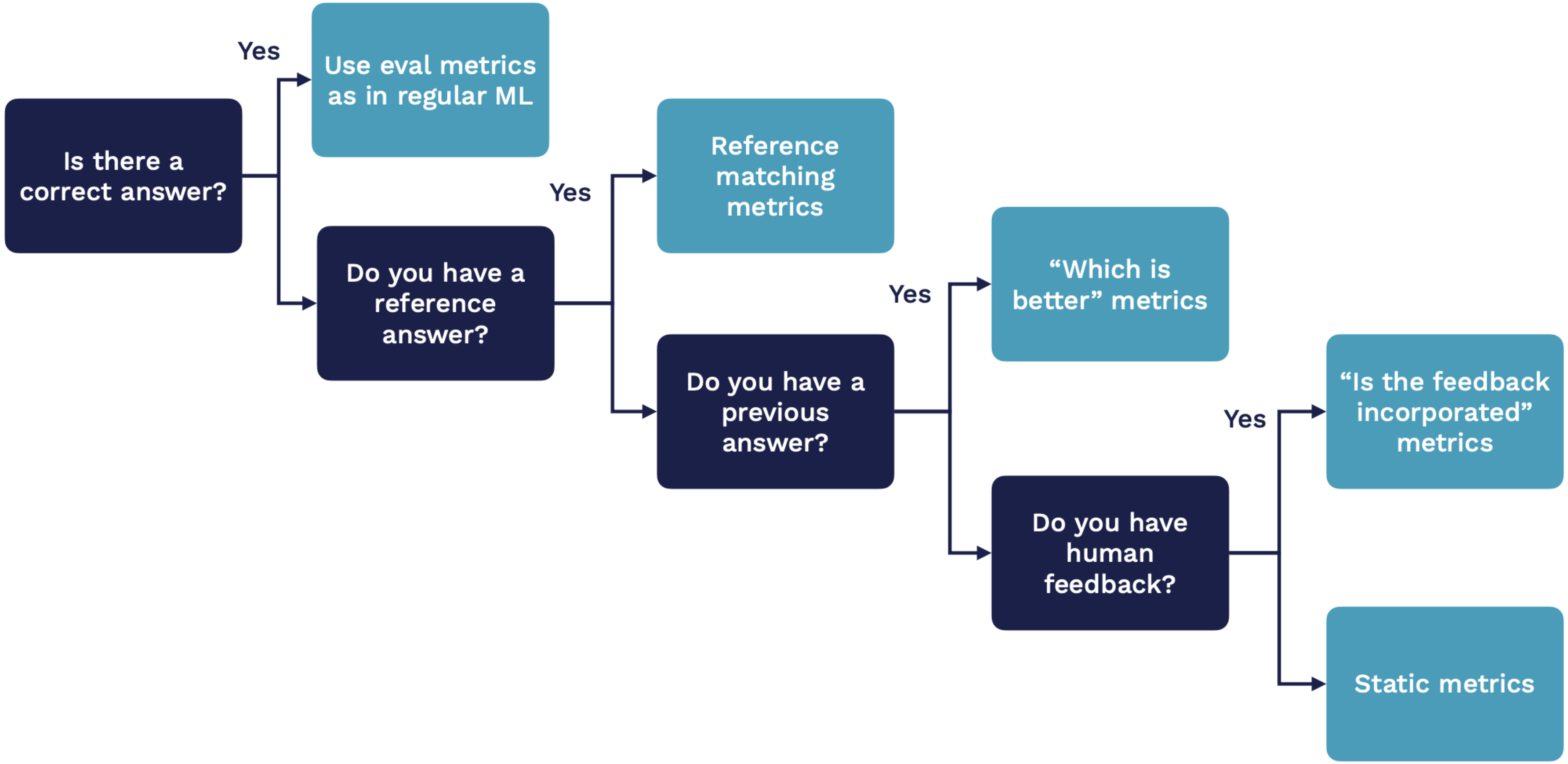
-> acc=0.9

- Generative

```
pred = ["this is an image of a tabby bear"]
label = ["photo of a cat"]
```

-> What metric?

Evaluation metrics for LLMs



Is there a correct answer?

Yes → Use eval metrics as in regular ML

Do you have a reference answer?

Yes → Reference matching metrics

Do you have a previous answer?

Yes → "Which is better" metrics

Do you have human feedback?

Yes → "Is the feedback incorporated" metrics

Static metrics

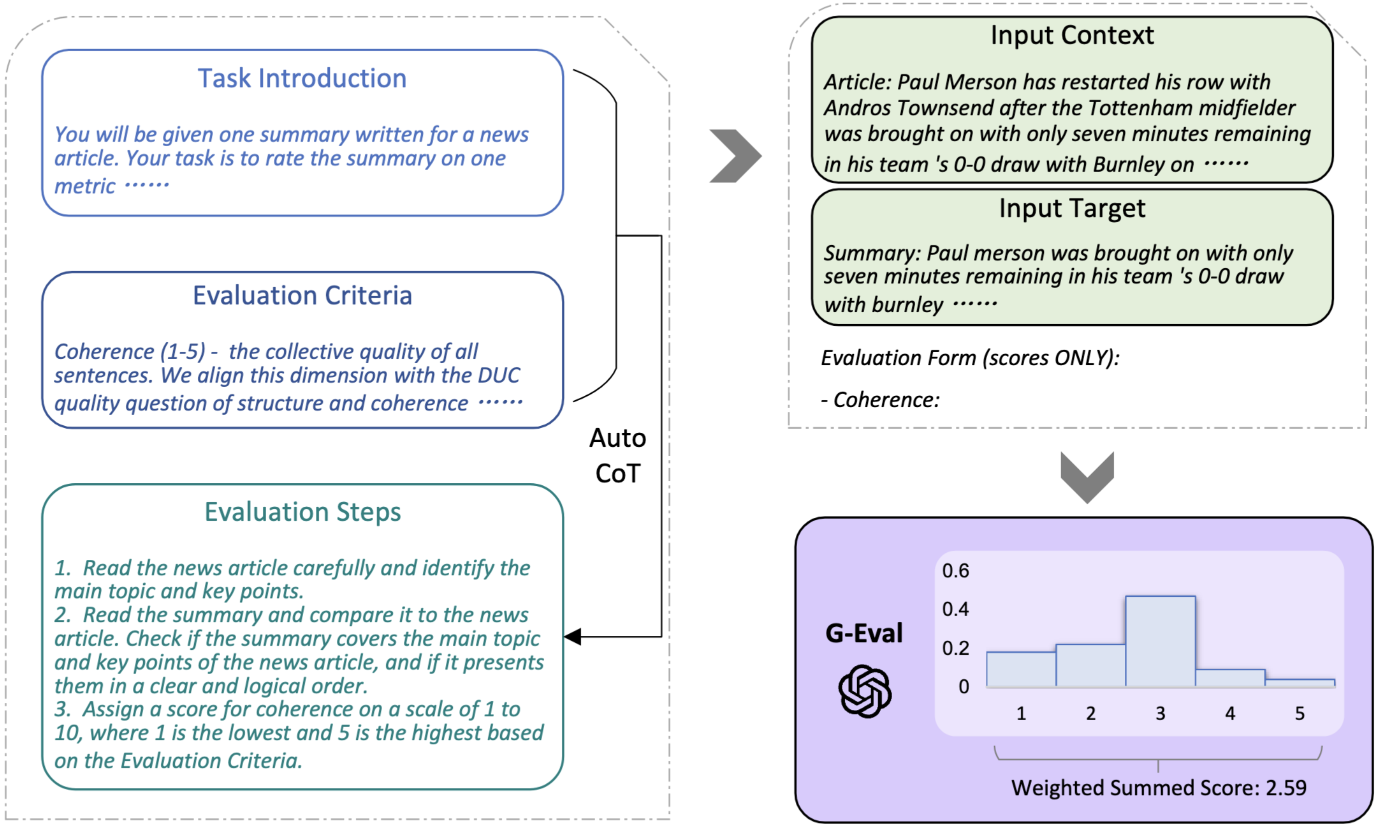
https://fullstackdeeplearning.com/llm-bootcamp/spring-2023/llm-foundations/

LLM Evaluation

- Regular eval metrics
  - Accuracy, etc
- Reference matching metrics
  - Semantic similarity, (BLEU, ROUGE, etc)
  - Ask another LLM, "are these two answers factually consistent”, etc
- “Which is better” metrics
  - Ask an LLM which of the two answers is better, according to any criteria you want
- "Is the feedback incorporated" metric
  - Ask an LLMs whether the new answer incorporates the feedback from the old answer
- Static metics
  - Verify the output has the right structure (JSON)
  - Ask a model to grade the answer (e.g. on a scale 1-5)

Key Idea: using LLMs to evaluate other LLMs

G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment



Task Introduction

You will be given one summary written for a news article. Your task is to rate the summary on one metric ……

Evaluation Criteria

Coherence (1-5) - the collective quality of all sentences. We align this dimension with the DUC quality question of structure and coherence ……

Auto CoT

Evaluation Steps

1. Read the news article carefully and identify the main topic and key points.
2. Read the summary and compare it to the news article. Check if the summary covers the main topic and key points of the news article, and if it presents them in a clear and logical order.
3. Assign a score for coherence on a scale of 1 to 10, where 1 is the lowest and 5 is the highest based on the Evaluation Criteria.

Input Context

Article: Paul Merson has restarted his row with Andros Townsend after the Tottenham midfielder was brought on with only seven minutes remaining in his team 's 0-0 draw with Burnley on ……

Input Target

Summary: Paul merson was brought on with only seven minutes remaining in his team 's 0-0 draw with burnley ……

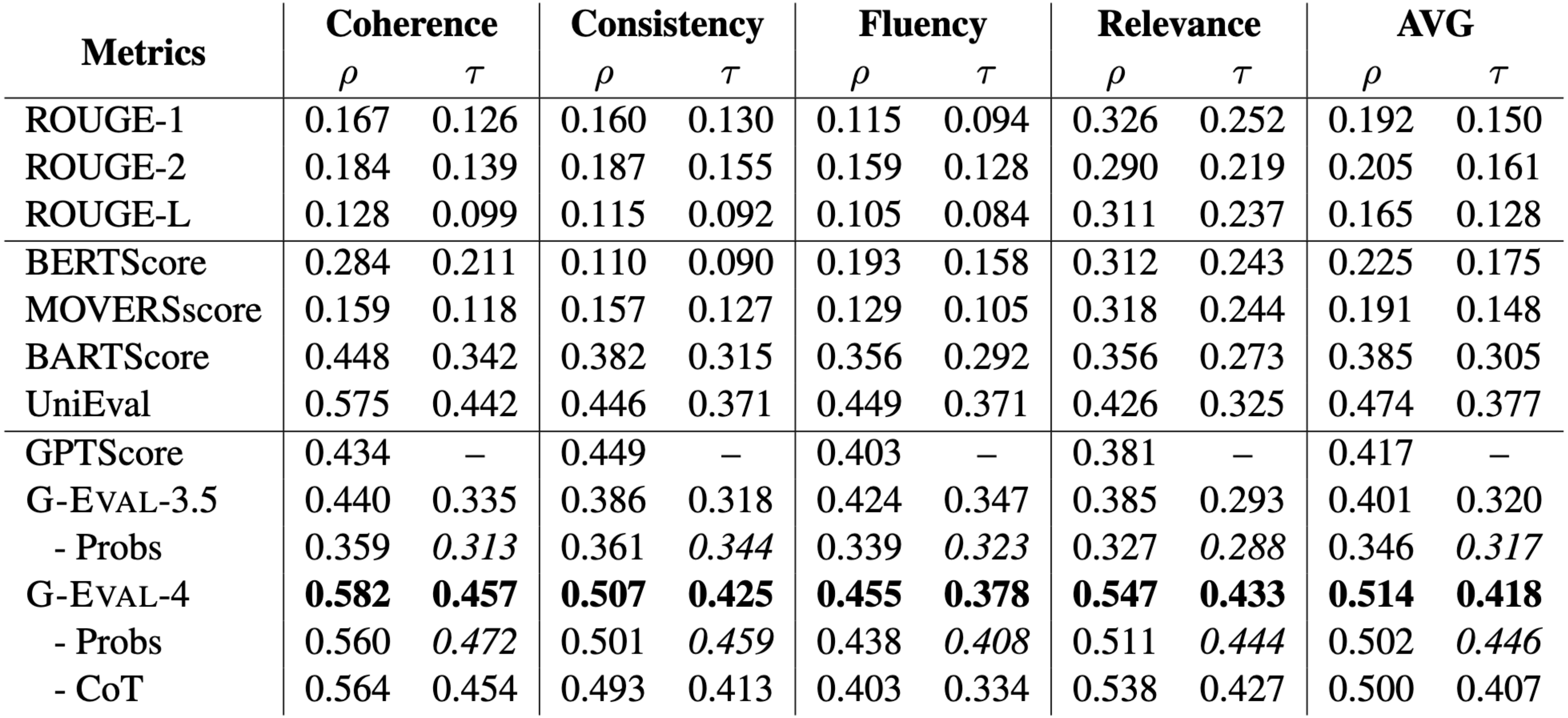
Evaluation Form (scores ONLY):

- Coherence:

G-Eval

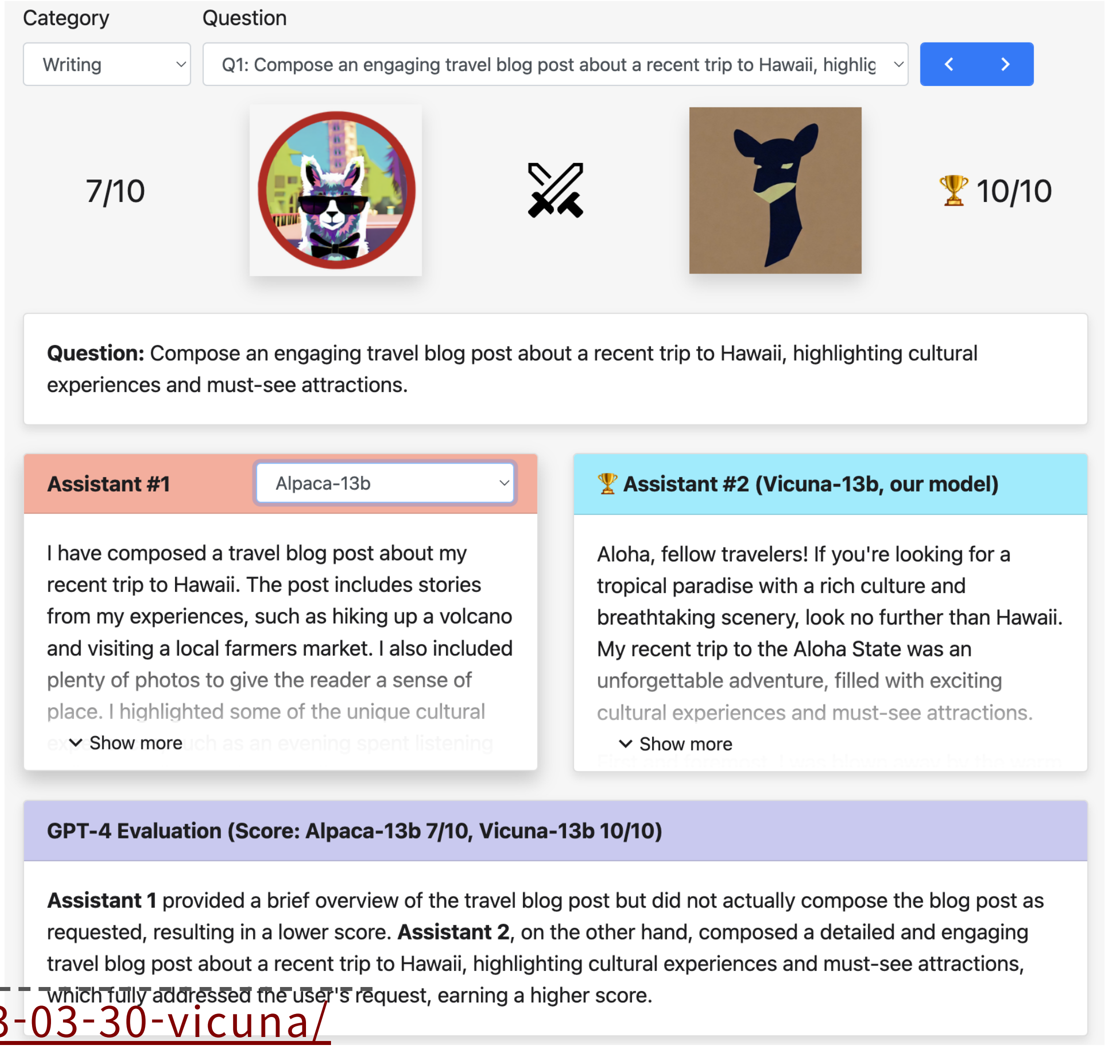
Weighted Summed Score: 2.59

G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment



Metrics	Coherence		Consistency		Fluency		Relevance		AVG	
	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
ROUGE-1	0.167	0.126	0.160	0.130	0.115	0.094	0.326	0.252	0.192	0.150
ROUGE-2	0.184	0.139	0.187	0.155	0.159	0.128	0.290	0.219	0.205	0.161
ROUGE-L	0.128	0.099	0.115	0.092	0.105	0.084	0.311	0.237	0.165	0.128
BERTScore	0.284	0.211	0.110	0.090	0.193	0.158	0.312	0.243	0.225	0.175
MOVERSscore	0.159	0.118	0.157	0.127	0.129	0.105	0.318	0.244	0.191	0.148
BARTScore	0.448	0.342	0.382	0.315	0.356	0.292	0.356	0.273	0.385	0.305
UniEval	0.575	0.442	0.446	0.371	0.449	0.371	0.426	0.325	0.474	0.377
GPTScore	0.434	–	0.449	–	0.403	–	0.381	–	0.417	–
G-EVAL-3.5	0.440	0.335	0.386	0.318	0.424	0.347	0.385	0.293	0.401	0.320
- Probs	0.359	*0.313*	0.361	*0.344*	0.339	*0.323*	0.327	*0.288*	0.346	*0.317*
G-EVAL-4	**0.582**	**0.457**	**0.507**	**0.425**	**0.455**	**0.378**	**0.547**	**0.433**	**0.514**	**0.418**
- Probs	0.560	*0.472*	0.501	*0.459*	0.438	*0.408*	0.511	*0.444*	0.502	*0.446*
- CoT	0.564	0.454	0.493	0.413	0.403	0.334	0.538	0.427	0.500	0.407

Evaluation of Vicuna paper


Category
Question
Writing
Q1: Compose an engaging travel blog post about a recent trip to Hawaii, highlig
‹ ›
7/10
🏆 10/10
Alpaca-13b
Show more
Show more

Question: Compose an engaging travel blog post about a recent trip to Hawaii, highlighting cultural experiences and must-see attractions.

Assistant #1

I have composed a travel blog post about my recent trip to Hawaii. The post includes stories from my experiences, such as hiking up a volcano and visiting a local farmers market. I also included plenty of photos to give the reader a sense of place. I highlighted some of the unique cultural

🏆 Assistant #2 (Vicuna-13b, our model)

Aloha, fellow travelers! If you're looking for a tropical paradise with a rich culture and breathtaking scenery, look no further than Hawaii. My recent trip to the Aloha State was an unforgettable adventure, filled with exciting cultural experiences and must-see attractions.

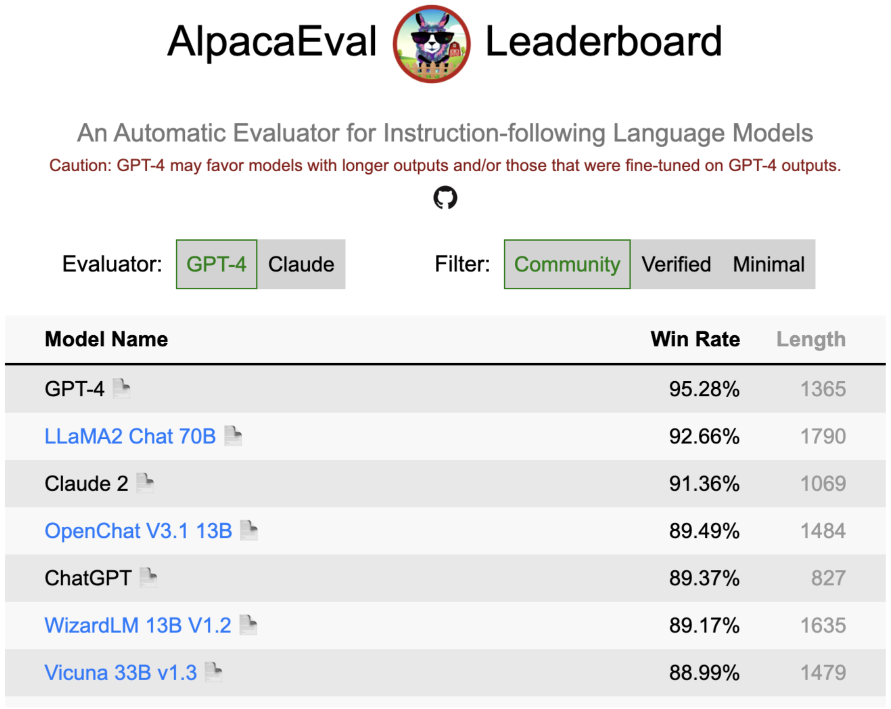
GPT-4 Evaluation (Score: Alpaca-13b 7/10, Vicuna-13b 10/10)

Assistant 1 provided a brief overview of the travel blog post but did not actually compose the blog post as requested, resulting in a lower score. **Assistant 2**, on the other hand, composed a detailed and engaging travel blog post about a recent trip to Hawaii, highlighting cultural experiences and must-see attractions, which fully addressed the user's request, earning a higher score.

---

https://lmsys.org/blog/2023-03-30-vicuna/

AlpacaEval



AlpacaEval Leaderboard

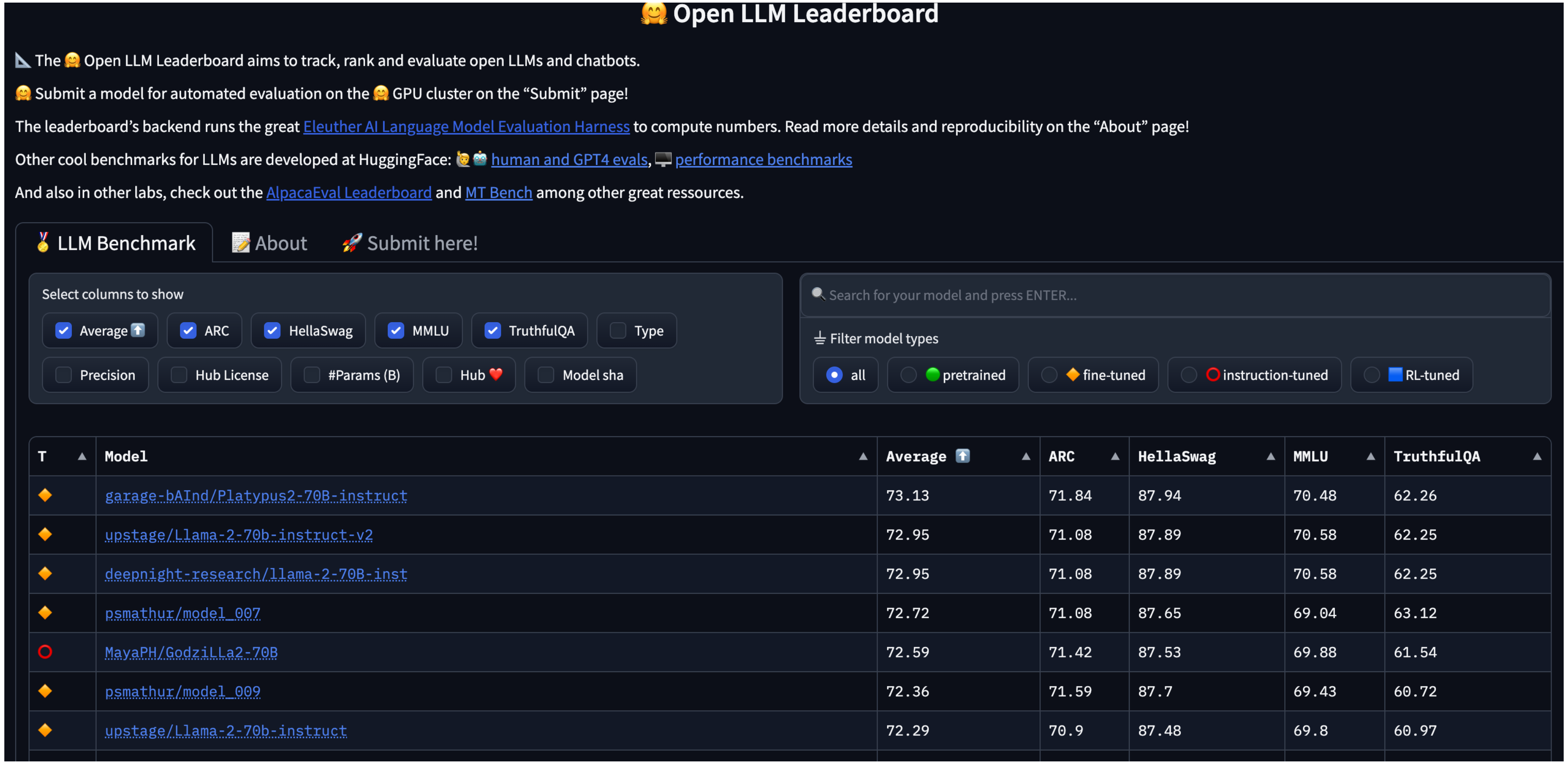
An Automatic Evaluator for Instruction-following Language Models

Caution: GPT-4 may favor models with longer outputs and/or those that were fine-tuned on GPT-4 outputs.

Model Name	Win Rate	Length
GPT-4	95.28%	1365
LLaMA2 Chat 70B	92.66%	1790
Claude 2	91.36%	1069
OpenChat V3.1 13B	89.49%	1484
ChatGPT	89.37%	827
WizardLM 13B V1.2	89.17%	1635
Vicuna 33B v1.3	88.99%	1479

- AutoEvaluator: Reference 모델의 출력보다 해당 모델의 출력을 선호하는 비율을 측정하여 모델을 평가
- Evaluator Model: GPT-4 or Claude (편향 가능성 존재)

Open LLM



🤗 Open LLM Leaderboard

📐 The 🤗 Open LLM Leaderboard aims to track, rank and evaluate open LLMs and chatbots.

🤗 Submit a model for automated evaluation on the 🤗 GPU cluster on the “Submit” page!

The leaderboard’s backend runs the great Eleuther AI Language Model Evaluation Harness to compute numbers. Read more details and reproducibility on the “About” page!

Other cool benchmarks for LLMs are developed at HuggingFace: 🙋🤖 human and GPT4 evals, 🖥️ performance benchmarks

And also in other labs, check out the AlpacaEval Leaderboard and MT Bench among other great ressources.

🏅 LLM Benchmark | 📝 About | 🚀 Submit here!

Select columns to show

☑ Average ⬆️ ☑ ARC ☑ HellaSwag ☑ MMLU ☑ TruthfulQA ☐ Type

☐ Precision ☐ Hub License ☐ #Params (B) ☐ Hub ❤️ ☐ Model sha

🔍 Search for your model and press ENTER...

Filter model types

● all ○ 🟢 pretrained ○ 🔶 fine-tuned ○ ⭕ instruction-tuned ○ 🟦 RL-tuned

T	Model	Average ⬆️	ARC	HellaSwag	MMLU	TruthfulQA
🔶	garage-bAInd/Platypus2-70B-instruct	73.13	71.84	87.94	70.48	62.26
🔶	upstage/Llama-2-70b-instruct-v2	72.95	71.08	87.89	70.58	62.25
🔶	deepnight-research/llama-2-70B-inst	72.95	71.08	87.89	70.58	62.25
🔶	psmathur/model_007	72.72	71.08	87.65	69.04	63.12
⭕	MayaPH/GodziLLa2-70B	72.59	71.42	87.53	69.88	61.54
🔶	psmathur/model_009	72.36	71.59	87.7	69.43	60.72
🔶	upstage/Llama-2-70b-instruct	72.29	70.9	87.48	69.8	60.97

Conclusion

Fine-tuning -> Instruction tuning

Closed LLMs -> Open LLMs

LLM Evaluation

Thank you