contextual·ai

Better, Cheaper, and Faster Alignment with KTO

Amanpreet Singh
CTO & Co-Founder

Outline

1. Motivation
2. KTO
3. Results
4. Archangel and Libraries

Alignment Protocol

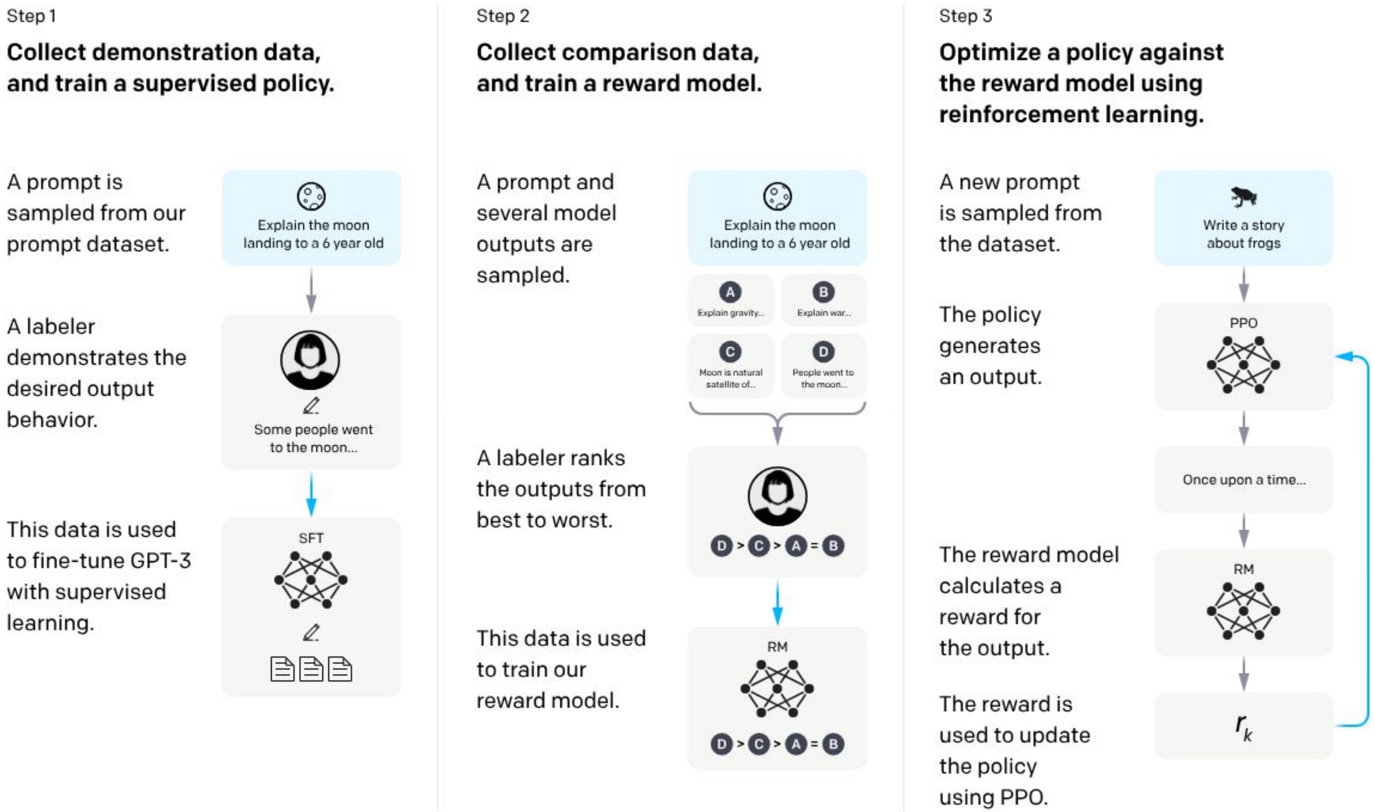


Image credit: https://openai.com/research/instruction-following

Alignment Protocol

Supervised Instruction Finetuning (SFT)

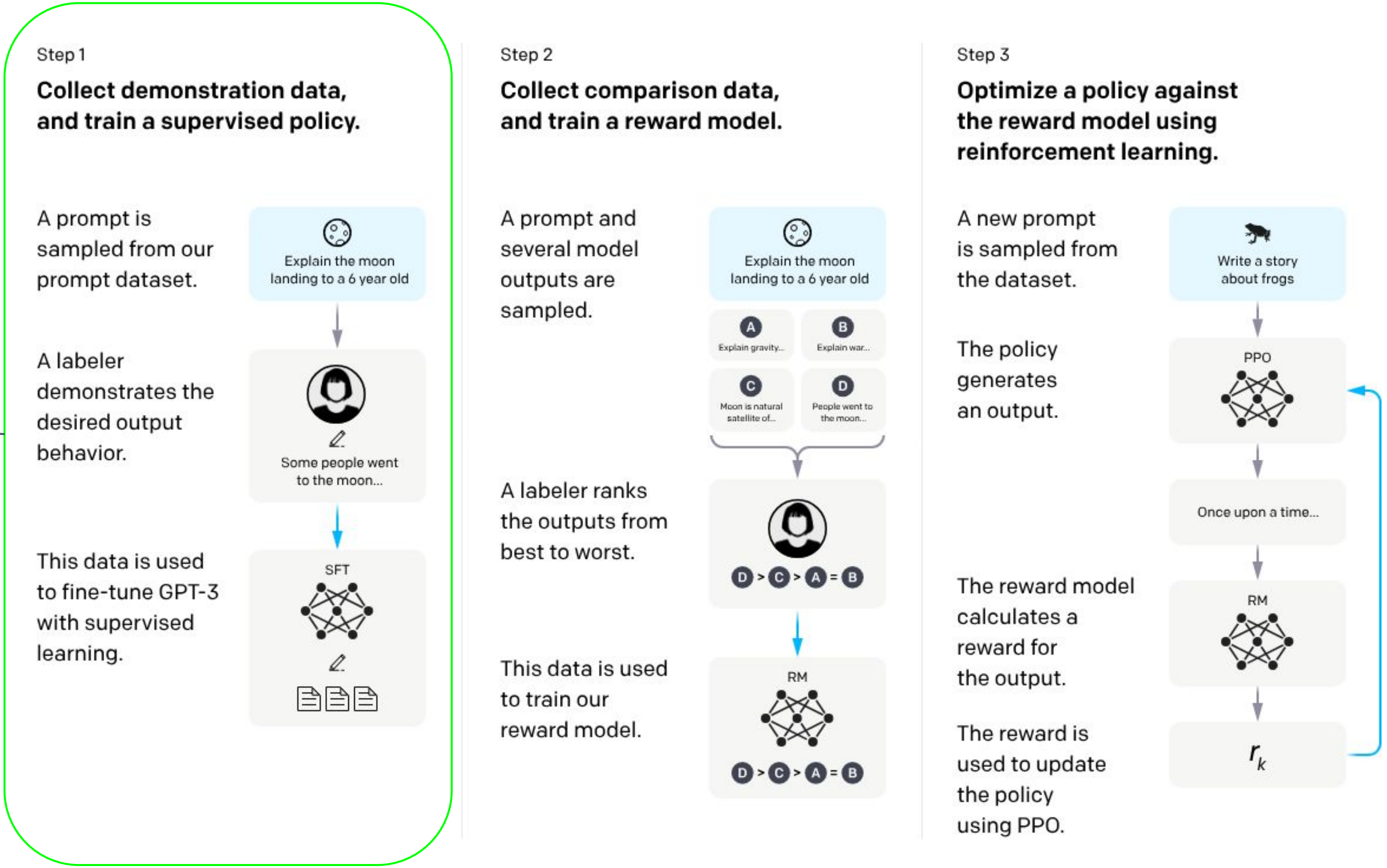


Image credit: https://openai.com/research/instruction-following

Alignment Protocol

Reinforcement Learning from Human Feedback (RLHF)

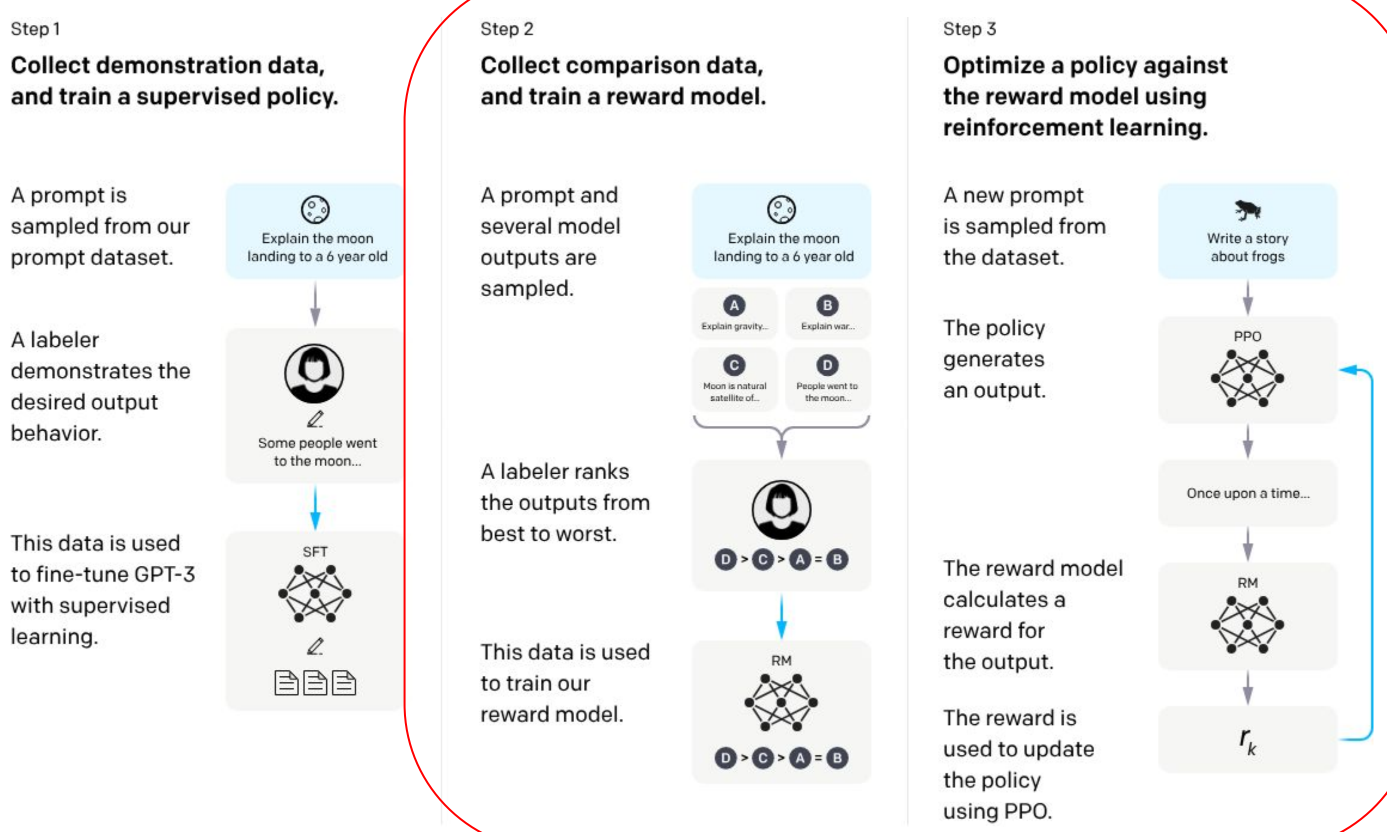


Image credit: https://openai.com/research/instruction-following

Aligning LLMs at scale

1. RLHF is hard.

Aligning LLMs at scale

1. RLHF is hard.
    a. Solution: Direct Preference Optimization (DPO)

Aligning LLMs at scale

1. RLHF is hard.
    a. Solution: Direct Preference Optimization (DPO)

$$L_{DPO}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}\right)\right]$$

Where:

- x is some prompt
- $\pi_\theta(y_w|x)$ and $\pi_\theta(y_l|x)$ are the probabilities of the preferred and dispreferred completions under the current model.
- $\mathbb{E}_{(x,y_w,y_l)\sim D}$ denotes the expectation over the dataset of preferences D.
- β is a parameter controlling the deviation from the base reference policy π_{ref}.

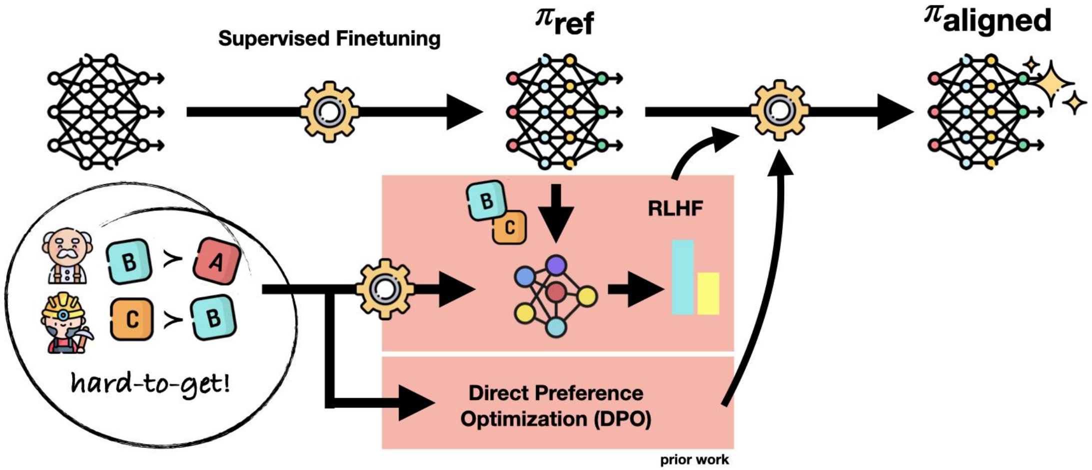
Aligning LLMs at scale

1. RLHF is hard.
    a. Solution: Direct Preference Optimization (DPO)
2. Paired preference data is expensive to collect and scale.

Aligning LLMs at scale

1. RLHF is hard.
    a. Solution: Direct Preference Optimization (DPO)
2. Paired preference data is expensive to collect and scale.
    a. Solution: KTO

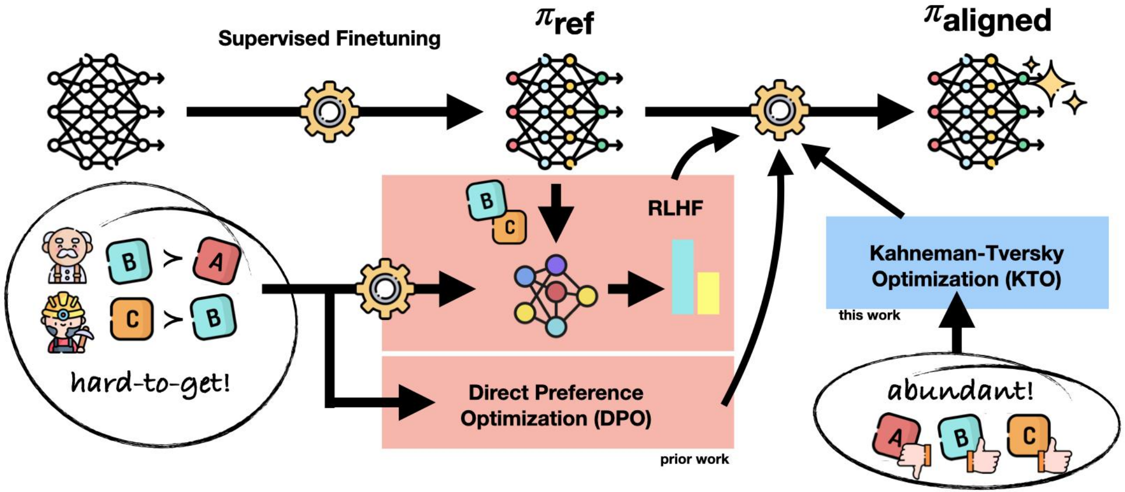
Enter KTO



Supervised Finetuning
π_{ref}
π_{aligned}
B
C
RLHF
B ≻ A
C ≻ B
hard-to-get!
Direct Preference Optimization (DPO)
prior work

Enter KTO

- doesn't require preference datasets
- works directly on abundantly available feedback data
- is more data efficient compared to other alignment methods



KTO Loss

- Directly maximizes the expected utility of an LM's outputs
  - Optimize the model to generate outputs that have higher utility values
- Utility function is inspired by Kahneman and Tversky's prospect theory
  - Determines the utility or desirability of an output from a human perspective

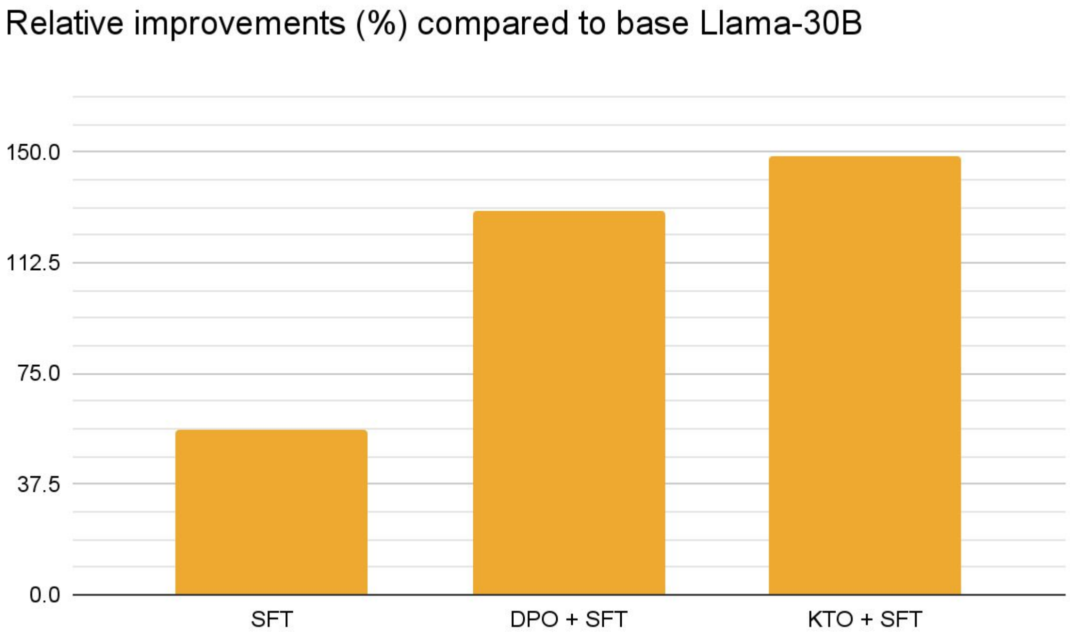
KTO Loss

- Directly maximizes the expected utility of an LM's outputs
  - Optimize the model to generate outputs that have higher utility values
- Utility function is inspired by Kahneman and Tversky's prospect theory
  - Determines the utility or desirability of an output from a human perspective

$$L_{\mathrm{KTO}}(\pi_\theta, \pi_{\mathrm{ref}}; \beta) = \mathbb{E}_{x,y\sim D}[1 - \hat{h}(x, y; \beta)]$$

$$\text{where } \hat{h}(x, y; \beta) = \begin{cases} \sigma(\beta \log \frac{\pi_\theta(y|x)}{\pi_{\mathrm{ref}}(y|x)} - \beta\, \mathrm{KL}(\pi_\theta \| \pi_{\mathrm{ref}})) & \text{if } y \sim y_{\mathrm{desirable}}|x \\ \sigma(\beta\, \mathrm{KL}(\pi_\theta \| \pi_{\mathrm{ref}}) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\mathrm{ref}}(y|x)}) & \text{if } y \sim y_{\mathrm{undesirable}}|x \end{cases}$$

KTO Performance



KTO Performance

	Win rate against		
	Llama-7B (SFT)	Llama-13B (SFT)	Llama-30B (SFT)
DPO	-20%	-8%	4%
DPO+SFT	-7%	0%	12%
KTO	-9%	-3%	**16%**
KTO+SFT	-2%	2%	15%

KTO Performance

	Win rate against		
	Llama-7B (SFT)	Llama-13B (SFT)	Llama-30B (SFT)
DPO	-20%	-8%	4%
DPO+SFT	-7%	0%	12%
KTO	-9%	-3%	**16%**
KTO+SFT	-2%	2%	15%

KTO Performance

	Win rate against		
	Llama-7B (SFT)	Llama-13B (SFT)	Llama-30B (SFT)
DPO	-20%	-8%	4%
DPO+SFT	-7%	0%	12%
KTO	-9%	-3%	**16%**
KTO+SFT	-2%	2%	15%

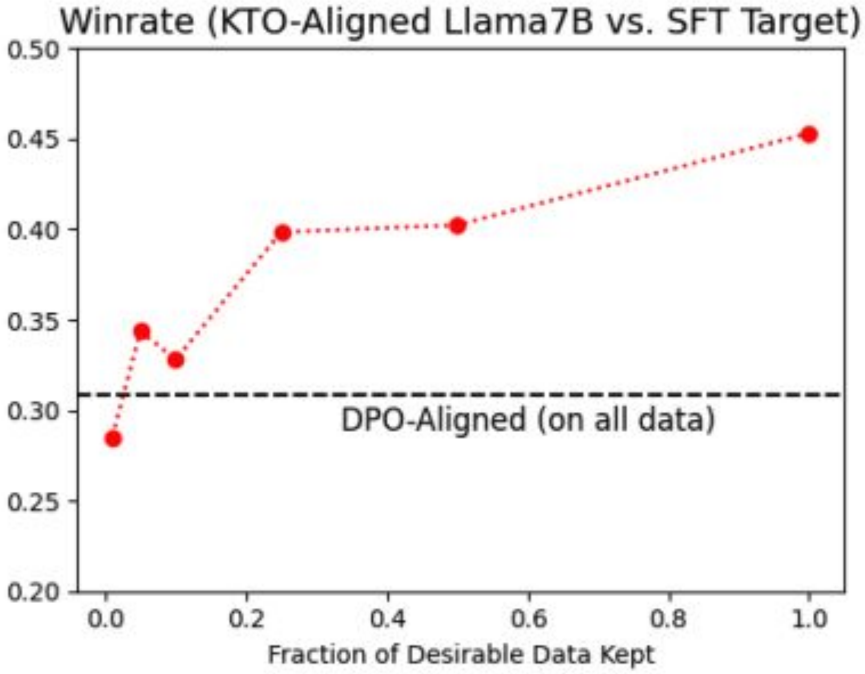
KTO Performance

	Win rate against		
	Llama-7B (SFT)	Llama-13B (SFT)	Llama-30B (SFT)
DPO	-20%	-8%	4%
DPO+SFT	-7%	0%	12%
KTO	-9%	-3%	**16%**
KTO+SFT	-2%	2%	15%

KTO Performance

	Win rate against		
	Llama-7B (SFT)	Llama-13B (SFT)	Llama-30B (SFT)
DPO	-20%	-8%	4%
DPO+SFT	-7%	0%	12%
KTO	-9%	-3%	**16%**
KTO+SFT	-2%	2%	15%

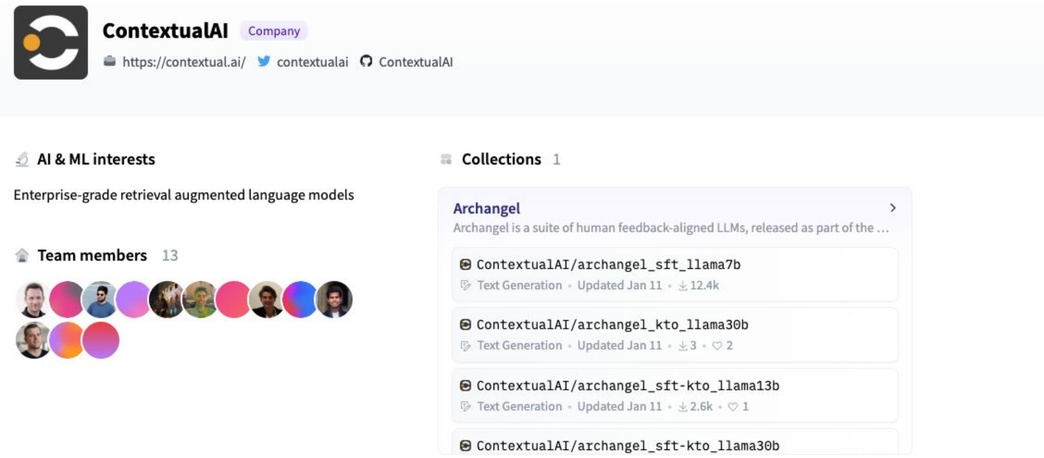
KTO Performance



Archangel Suite

56 human-feedback aligned models

- on 7 different sizes (1B to 30B)
- using 8 different methods
- with 3 of the largest public human feedback datasets
- All open source



More details on KTO can be found in our blog post:

http://tinyurl.com/kto-ctxl

KTO in the wild

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.

Version: AlpacaEval | AlpacaEval 2.0   Filter: Community | Verified

Baseline: GPT-4 Turbo | Auto-annotator: GPT-4 Turbo

Model Name	Win Rate	Length
GPT-4 Turbo	50.00%	2049
Contextual AI (KTO-Mistral-PairRM)	33.23%	2521
Yi 34B Chat	29.66%	2123
Claude 3 Opus (02/29)	29.04%	1388
Claude 3 Sonnet (02/29)	25.56%	1420
GPT-4	23.58%	1365
GPT-4 0314	22.07%	1371
Mistral Medium	21.86%	1500
Mixtral 8x7B v0.1	18.26%	1465
Claude 2	17.19%	1069
Claude	16.99%	1082
Tulu 2+DPO 70B	15.98%	1418
GPT-4 0613	15.76%	1140
Claude 2.1	15.73%	1096

Using KTO

- KTO Github Repo - ContextualAI/HALOs
- Hugging Face TRL - huggingface/trl
- NVIDIA's NeMo-Aligner (April 2024) - NVIDIA/NeMo-Aligner

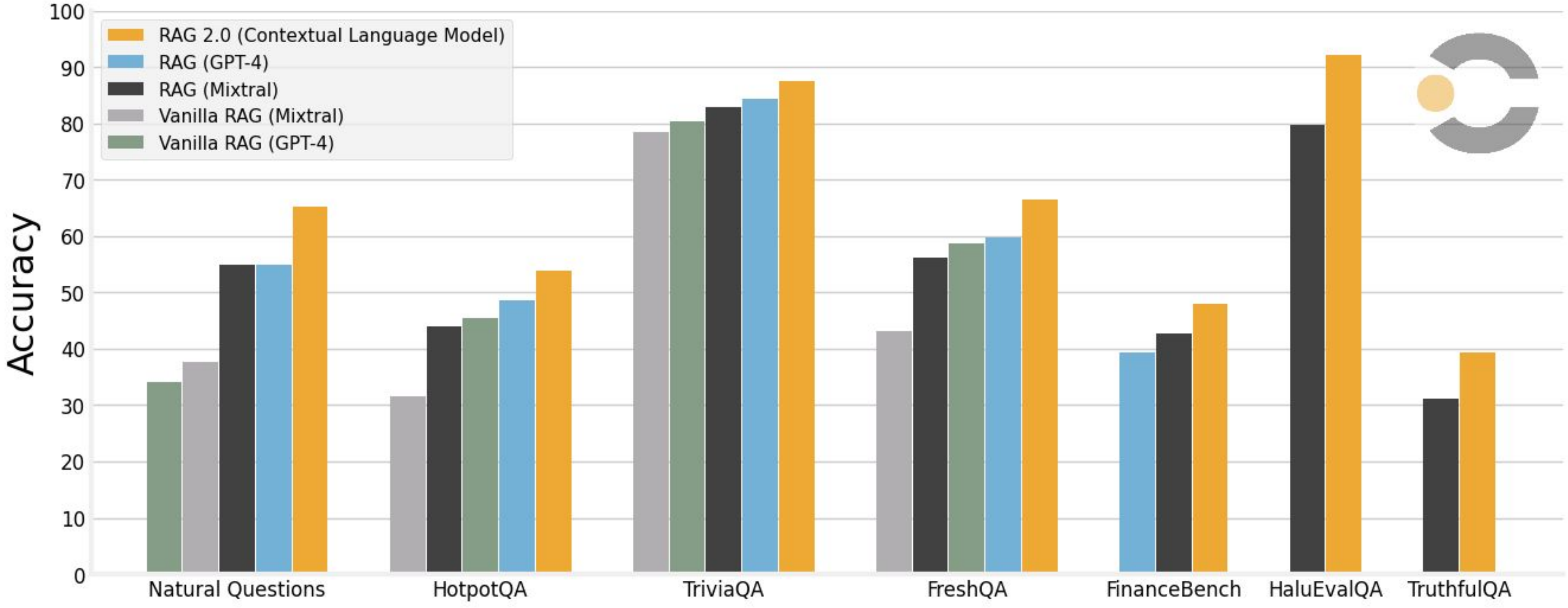
Takeaways

KTO:

1. is a strong alternative to DPO and RLHF
2. works well when used directly without any SFT
3. can directly work in production on abundantly available feedback data
4. can work directly on any kind of feedback signal
5. can also work with imbalanced data

RAG 2.0 @ Contextual AI

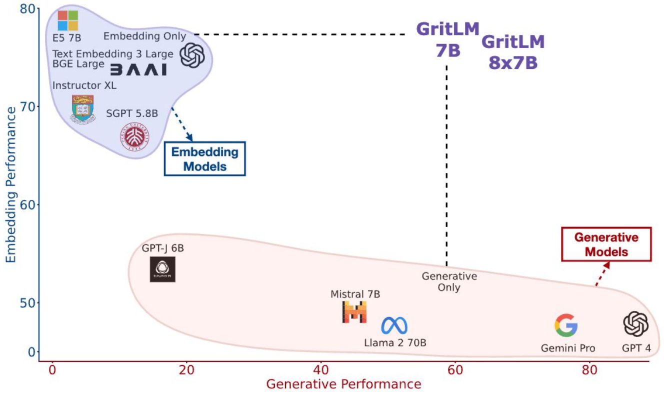
Read more at rag2.ai



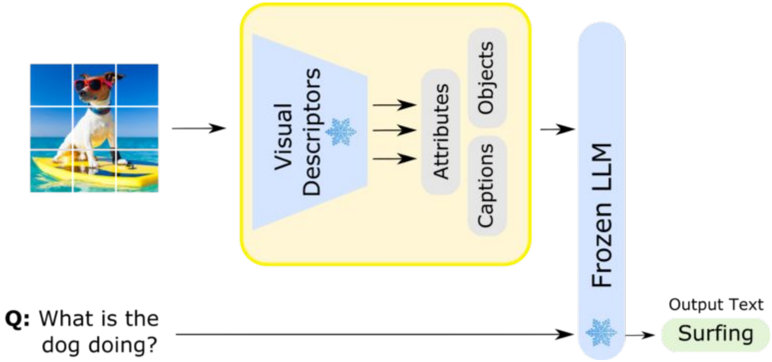
Other research from Contextual AI

Read more at contextual.ai

GRIT: State of the art embedder, reranker and LLM in a single model



LENS: Add vision capabilities to any LLM out of the box



Twitter: @apsdehal
aman@contextual.ai