Fine-tuning Large Language Models

Yash Patel, Aniket Jivani

EECS 598 Presentation

February 12, 2024

Overview

Plan

Papers

1. Finetuned Language Models are Zero-shot Learners (Wei et al., 2022)
2. LIMA: Less is More for Alignment (Zhou et al., 2023)
3. Principled Recaptioning for Better Image Generation (Segalis et al., 2023)

Plan

Paper Overview

1. Wished to show zero-shot learning achievable with instruction tuning
2. Main contribution: **Empirical** evidence to suggest LLMs satisfy this goal

Zero-shot Learning

1. Conceptually: predict without having seen *any* labels for class
2. Formally, predictor $f_\theta : \mathcal{X} \to \mathcal{Y}$, where

$$\mathcal{D}_{\text{train}} = \{(x_i, y_i)\} \qquad \mathcal{D}_{\text{test}} = \{(\widetilde{x}_i, \widetilde{y}_i)\} \text{ s.t. } \widetilde{y}_i \in (\mathcal{Y} \backslash \bigcup_i y_i)$$

3. Requires having seen "similar" points in training

Fine-Tuning (Review)

1. Predictor f_θ initially trained on $\mathcal{D}_{\text{train}}$
2. Re-trained (in part or fully) on $\mathcal{D}_{\text{ft}}$, i.e.

$$\theta^* \leftarrow \theta - \eta \nabla_\theta \sum_{(x_i, y_i) \in \mathcal{D}_{\text{ft}}} \mathcal{L}(f_\theta(x_i), y_i)$$

3. Often undesirable due to space or time reasons
4. Most often useful if have *several* tasks of the forms $\mathcal{X} \to \mathcal{Y}_1, \ldots, \mathcal{X} \to \mathcal{Y}_N$
5. Use same initial piece $f : \mathcal{X} \to \mathcal{Y}$ and train N separate "fine-tuning" heads, i.e. $g_j : \mathcal{Y} \to \mathcal{Y}_j$ to get $f \circ g_j : \mathcal{X} \to \mathcal{Y}_j$

Prompt Tuning

1. Suppose now specialization tasks are in same space but *qualitatively* differ
2. Formally, $\mathcal{Y}_1, ..., \mathcal{Y}_N = \mathcal{Y}$
3. Can instead change the *inputs* to the model

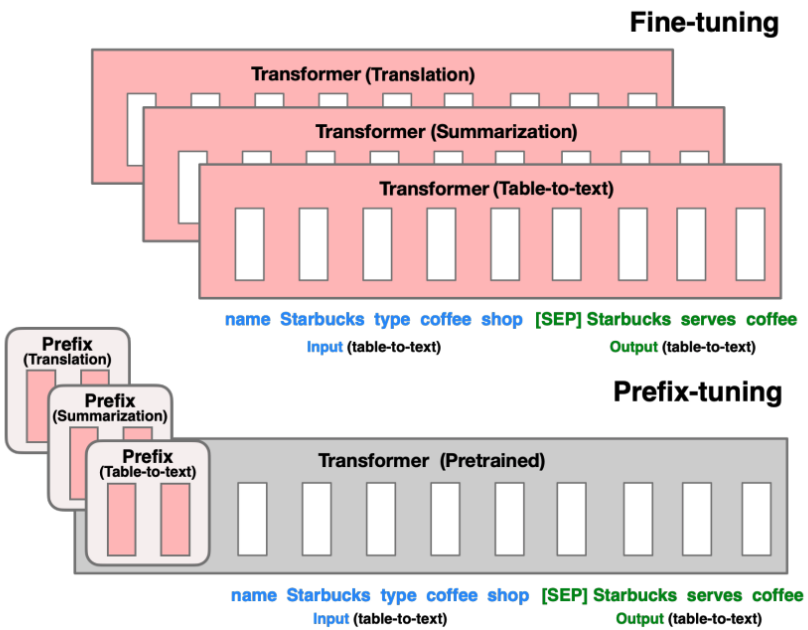


Figure: Prompt tuning overview

Instruction Tuning

1. **Fundamentally** differs from both fine-tuning and prompt-tuning
2. **No weights** are trained! Also called "in-context learning"
3. Only perform inference: no space requirements

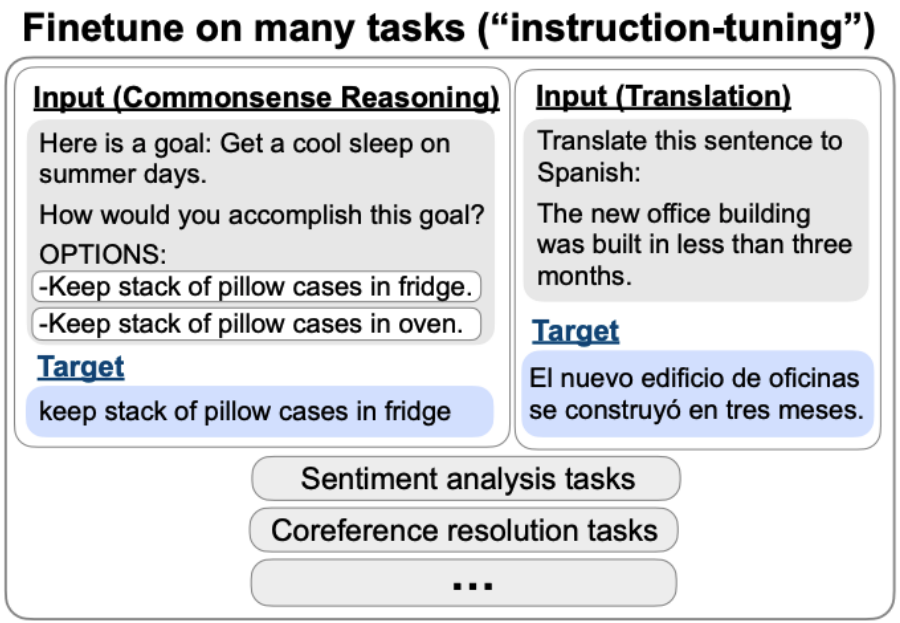


Figure: Instruction tuning overview

Instruction Tuning (Intuition)

1. Surprising that this may work at all
2. Recall a transformer prediction is of the form:

$$\left(\frac{KQ^\top}{\sqrt{n}}\right)V$$

, where K, Q, V are maps from some input X followed by multiplies by W_K, W_Q, W_V

3. Prediction $f_\theta(x_T)$ is a "stateful" computation dependent on $x_{1:T-1}$, i.e. should think of this as $f_\theta(x_T; x_{1:T-1})$
4. Differ from more straightforward applications of ML, where there is no "state" in computation

Instruction Tuning (Intuition)

1. So, specifying some prompt $x_{1:T-1}$ may produce predictions equivalent to *as if* θ had been updated
2. Can imagine transformer $f_\theta(x_T; x_{1:T-1})$ "uses" the $x_{1:T-1}$ to train some "internal model" used for future x_T, i.e.

$$f_\theta(x_T; x_{1:T-1}) = g_{\phi(x_{1:T-1})}(x_T)$$

where ϕ are "internal parameters" trained over $x_{1:T-1}$

Instruction Tuning (Formal)

1. Separate work established this formally (detail unimportant):

Coverage Bound

Given a 1-head linear attention layer and the tokens $e_j = (x_j, y_j)$, for $j = 1, ..., N$, one can construct key, query and value matrices W_K, W_Q, W_V as well as the projection matrix P such that a Transformer step on every token e_j is identical to the gradient-induced dynamics $e_j \leftarrow (x_j, y_j) + (0, -\Delta W x_j) = (x_j, y_j) + PVKTq_j$ such that $e_j = (x_j, y_j - \Delta y_j)$. For the test data token (x_{N+1}, y_{N+1}) the dynamics are identical

Experimental Setup

1. Many tasks (specifics unimportant)
2. Zero-shot learning entailed being instruction tuned on other *categories* of tasks and being evaluated on an unobserved *category*

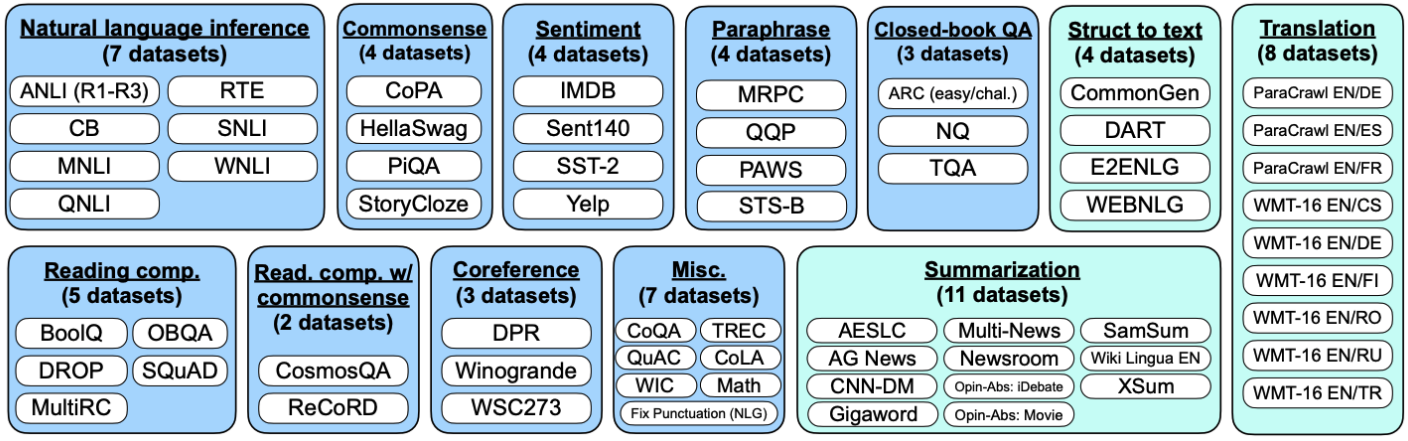


Figure: Task clustering across NLP

Experimental Setup

1. Classification were converted to natural language and so on

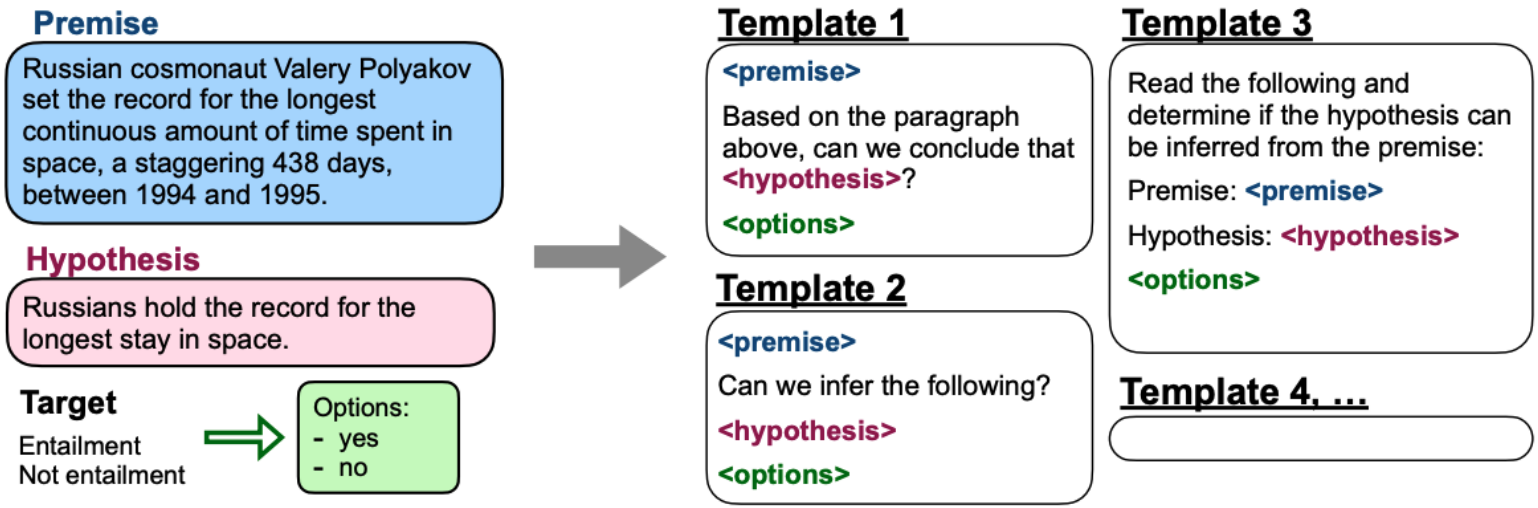


Figure: Prompts had to be constructed with particular structure

Results

1. Predictably good results across the board

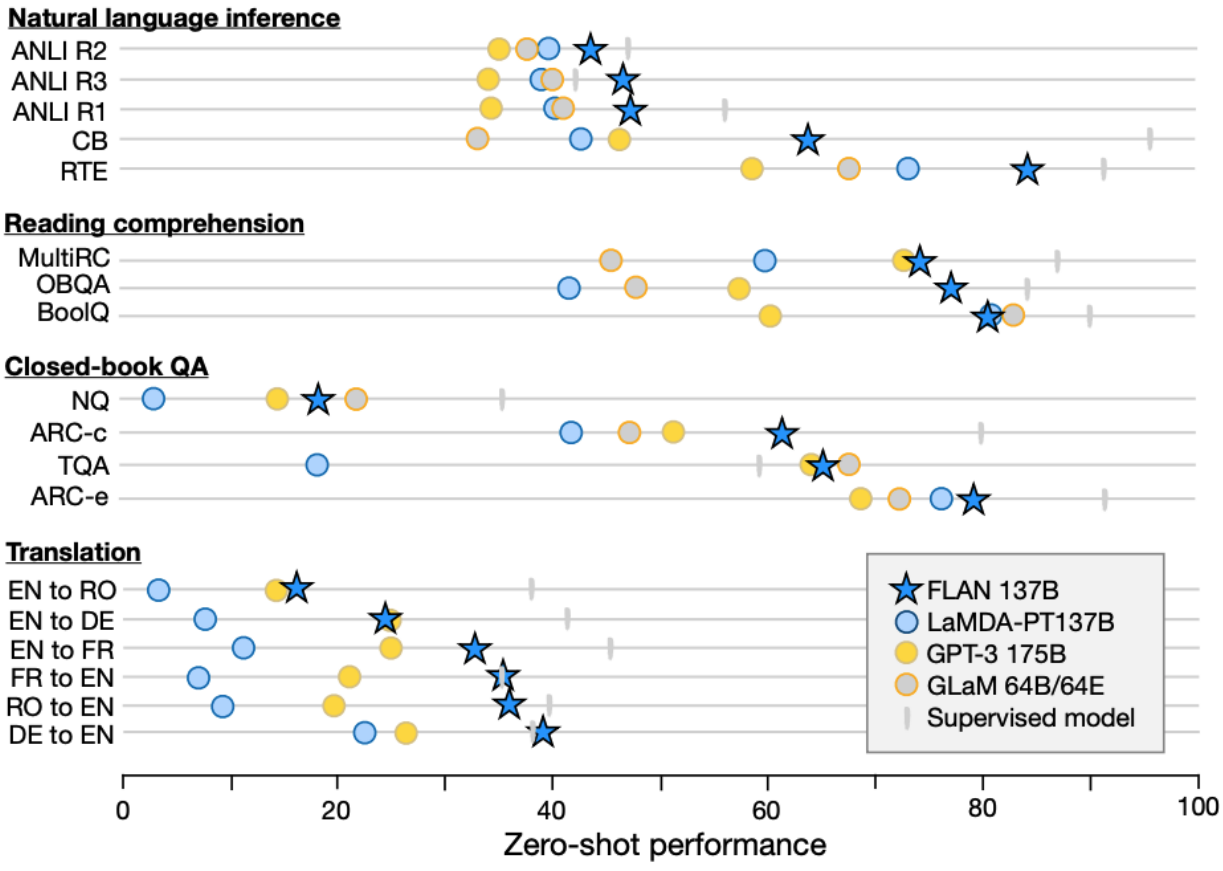


Figure: Outperforms baseline zero-shot learners across the board

Results

1. Ablation on adjacent tasks for instruction tuning

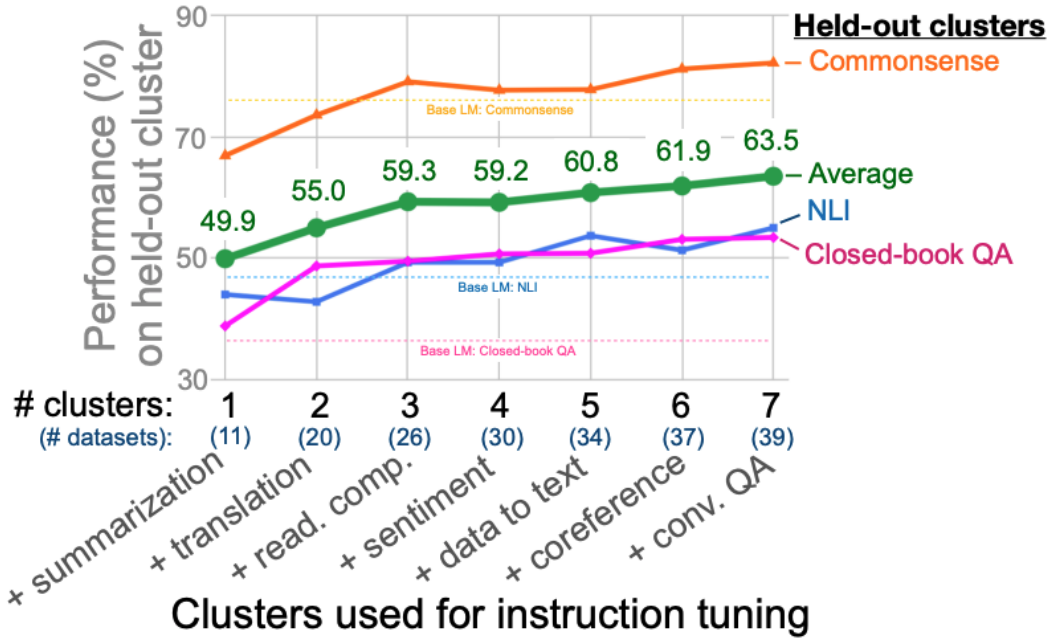


Figure: As expected, as number of tasks increases, performance on holdout also improves

Results

1. Only real surprise in results: performance seems dependent on model size

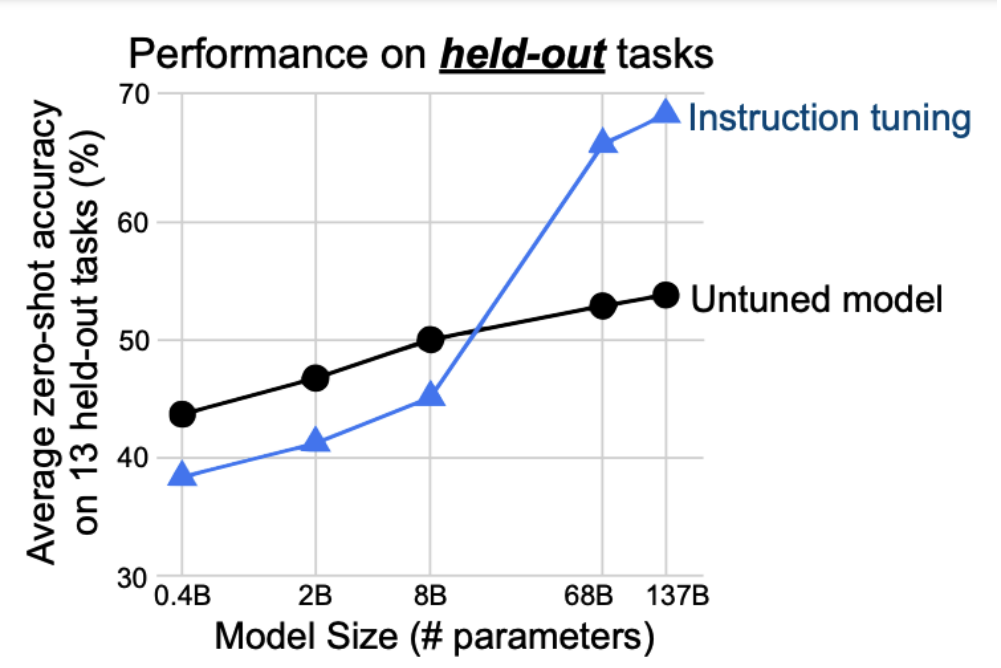


Figure: Apparently instruction tuning fails on small models: mysterious!

Results

1. Can also augment to have few-shot learning
2. In this case, instructions are $x_1 \oplus y_1 \oplus ... \oplus x_K \oplus y_K$ for K-shot learning

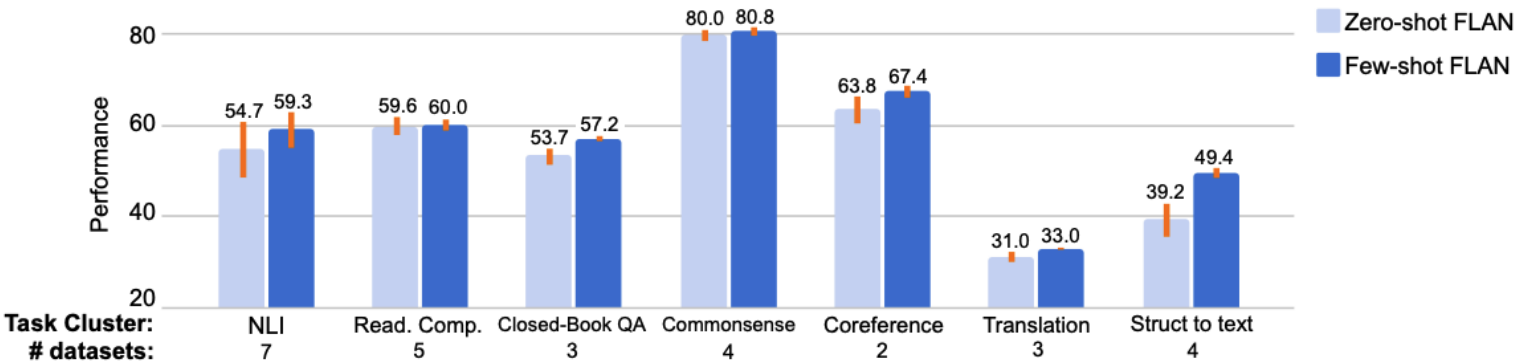


Figure: As expected, improves performance

Results

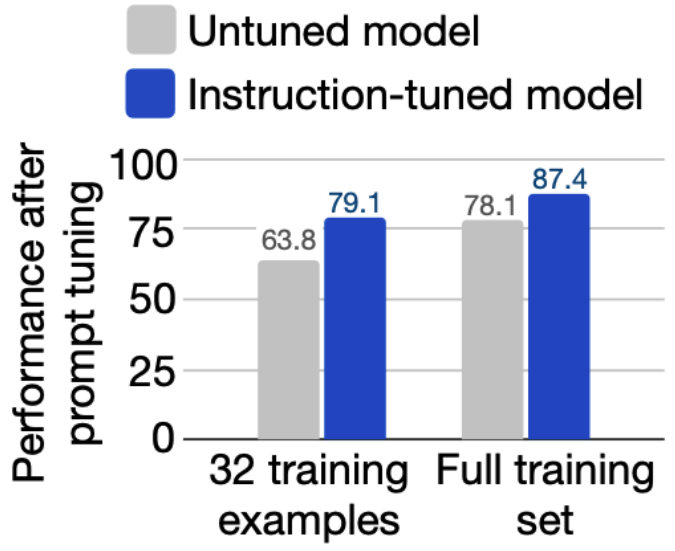


Figure: Similar behavior is seen if we do this on top of prefix-tuning

Future Direction (Systems)

1. Unclear beyond serving as more motivation for inference acceleration

Plan

Paper Overview

1. **Main contribution:** Improved image captions result in higher quality conditional generation

Method

1. Recall conditional generation seeks to draw $\widehat{Y} \sim \mathcal{P}(Y \mid X)$
2. Simple idea based on observation
3. Default image captions are noisy
4. Formally, we might be observing pairs $(\widehat{X}, Y)$ instead of the true (X, Y) pairs
5. Drawing sample for X then will be incorrect

Method

1. Instead of using default X, generate $X \sim \mathcal{P}(X \mid Y)$ from image-to-text captioning
2. Default model is noisy so fine-tune with manual curation

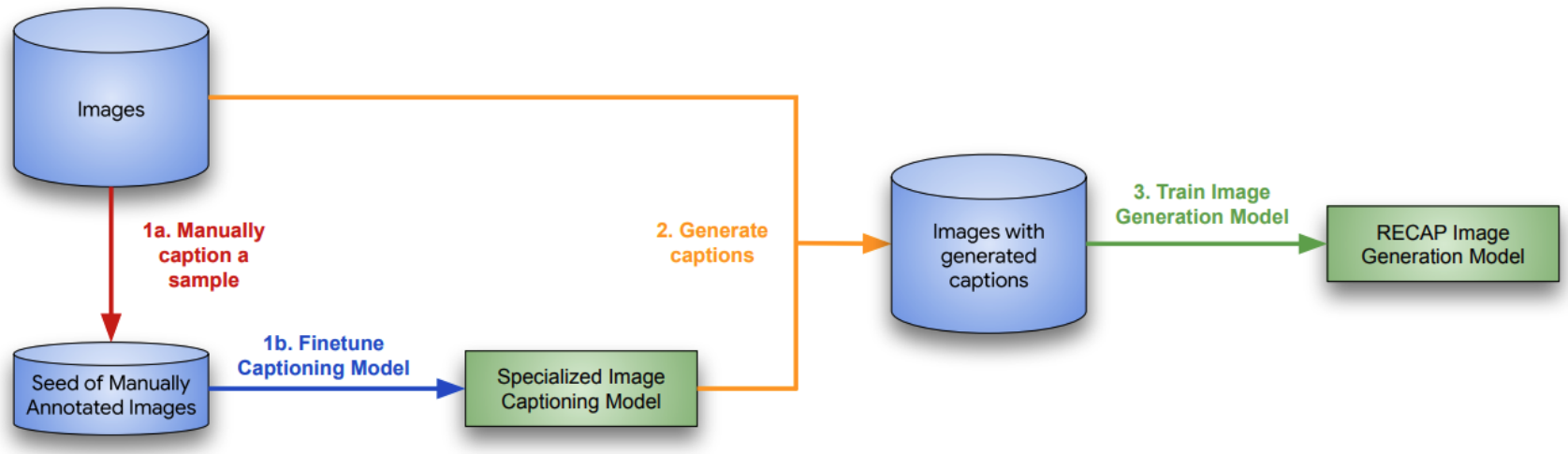


Figure: Overall workflow generates captions and then trains as before

Results

		
Short Caption	A blue willys gasser car	A glass of iced tea
Long Caption	A willys gasser car in blue color. it is placed near the car shed on the floor. in the back, there is a man seated in the chair	A glass of iced tea placed on a saucer decorated with mint leaves. it is located on the wood table
PaLI	a blue coupe with a big engine in it .	a glass of iced tea with a straw and mint .

Figure: Improves captions considerably

Results

	FID↓	O-FID↓	SOA-C↑	SOA-I↑	CA↓	PA↑	RP↑
Baseline	17.87	8.19	78.90	80.80	1.44	57.60	92.78
Alttext	17.53	8.90	78.99	80.85	1.47	57.40	91.32
RECAP	**14.84**	**6.23**	**84.34**	**86.17**	**1.32**	**62.42**	**93.80**
Real Images	2.62	0.00	90.02	91.19	1.05	100.0	83.54

Table 1. Results for the automated metrics for RECAP model vs. baseline and Alttext models. RECAP model improves across all metrics.

	MS-COCO Successful Images	MS-COCO Successful Prompts	DrawBench Successful Images	DrawBench Successful Prompts
Baseline	29.3%	53.5%	15.6%	33.1%
Alttext	33.4%	60.0%	13.2%	33.8%
RECAP	**48.1%**	**76.0%**	**22.1%**	**45.5%**

Table 2. Human evaluation results comparing RECAP, Alttext and Baseline models, on two benchmarks (MS-COCO and DrawBench) for two metrics: percentage of images generated that fully follow the prompt, and percentage of prompts with at least one generated image (out of four seeds) that fully followed the prompt.

Figure: In turn improves downstream generation

References I

Segalis, E., D. Valevski, D. Lumen, Y. Matias, and Y. Leviathan (2023, October). A picture is worth a thousand words: principled recaptioning improves image generation. arXiv:2310.16656 [cs].

Wei, J., M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le (2022). Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Zhou, C., P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, S. Zhang, G. Ghosh, M. Lewis, L. Zettlemoyer, and O. Levy (2023, May). Lima: less is more for alignment. arXiv:2305.11206 [cs].

Plan

Thank you!

LIMA : Less is More For Alignment

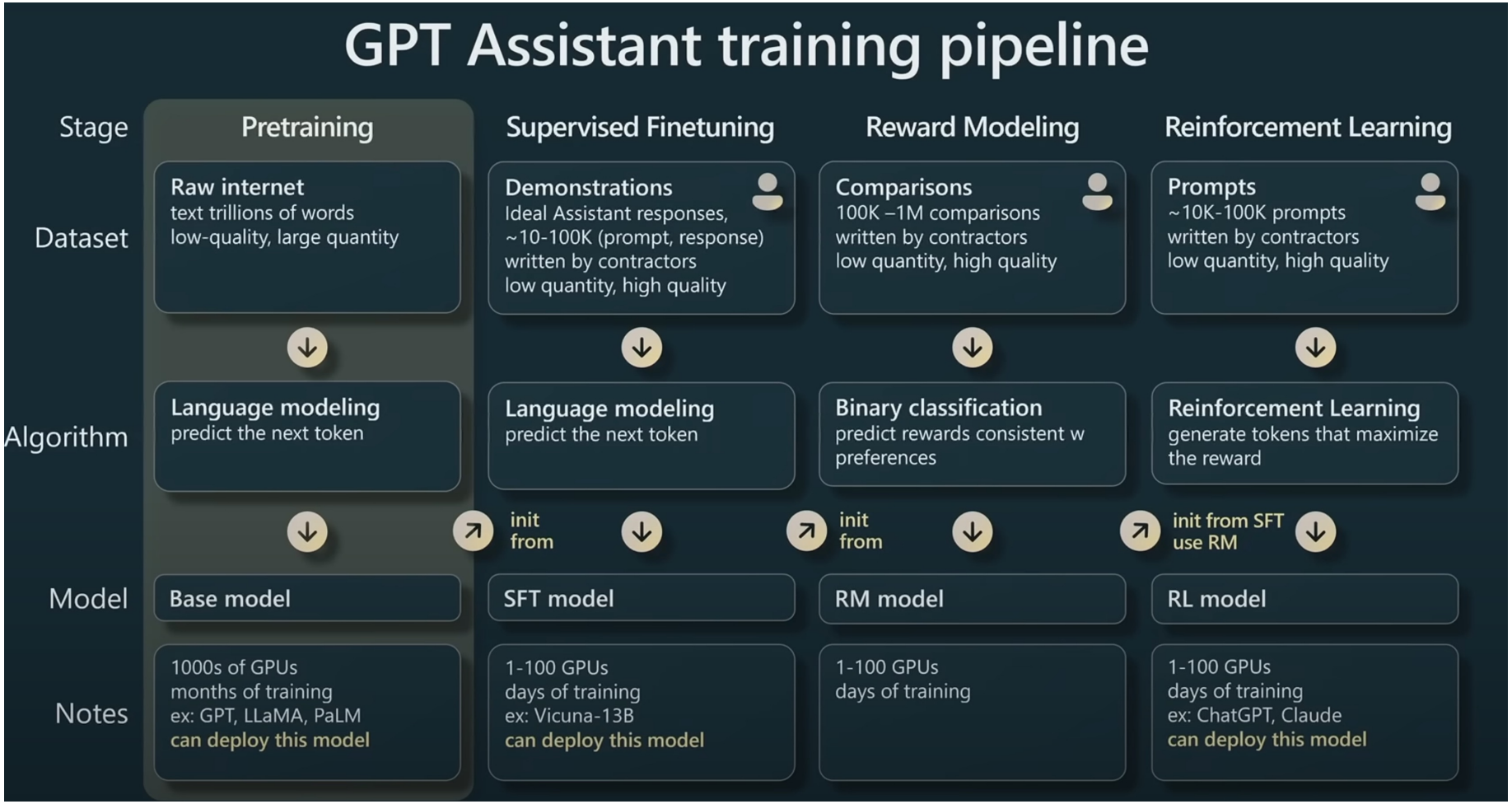
Zhou, Chunting, et al. *LIMA: Less Is More for Alignment*. arXiv, 18 May 2023. *arXiv.org*, https://doi.org/10.48550/arXiv.2305.11206.

Outline

- Motivation
- Background
- Methodology – Response generation, evaluation
- Results – Comparisons based on datasets, models etc.
- Follow up works
- Summing up

Introduction

Source: State of GPT Talk, 2023: https://www.youtube.com/watch?v=bZQun8Y4L2A –
Multiple stages following pre-training for models to be useful – "alignment"



RLHF

Alignment: "we use a framework similar to Askell et al. (2021), who define models to be aligned if they are **helpful, honest, and harmless."**

Related Works:

https://arxiv.org/pdf/2204.05862.pdf

Bai et al (2022) – (close to 50K helpfulness and red-teaming response datasets)

https://arxiv.org/pdf/2203.02155.pdf
Ouyang et al. (2022) – Instruct GPT

1. Collect demonstration data – train supervised policy
2. Comparison between model outputs – train reward model for human preferred outputs
3. Optimize policy based on output from reward model for sampled prompts-responses.

Generally, costly and compute heavy process – difficult to achieve GPT levels of performance otherwise on open models

Objective

- **Superficial Alignment Hypothesis:** A model's knowledge and capabilities are learnt almost entirely during pretraining, while alignment teaches it which subdistribution of formats should be used when interacting with users

Prior work : Kirstain et al. (2021) - https://arxiv.org/abs/2110.04374

- Model learns style or format of interaction through carefully curated examples
  - From existing datasets
  - Manually created prompts and responses

- Finetune a 65B parameter LLaMa model on above (1000 examples)

Analyze the following:

- (Relative) Human preferred response comparison – GPT-4, Claude, Bard, LIMA
- (Absolute) Evaluations – Meets prompt requirements?
- Scaling with data quantity and / or prompt diversity

Curation of Examples

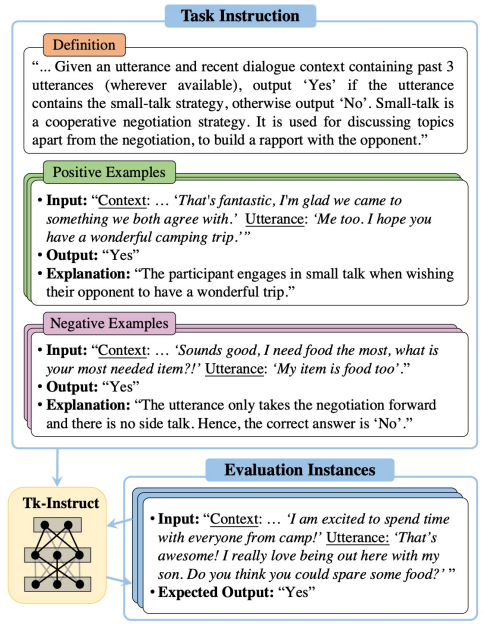
https://arxiv.org/pdf/2204.07705.pdf

200 manually authored examples

SuperNatural Instructions dataset – 50 prompts

Source	#Examples	Avg Input Len.	Avg Output Len.
Training			
Stack Exchange (STEM)	200	117	523
Stack Exchange (Other)	200	119	530
wikiHow	200	12	1,811
Pushshift r/WritingPrompts	150	34	274
Natural Instructions	50	236	92
Paper Authors (Group A)	200	40	334
Dev			
Paper Authors (Group A)	50	36	N/A
Test			
Pushshift r/AskReddit	70	30	N/A
Paper Authors (Group B)	230	31	N/A

Table 1: Sources of training prompts (inputs) and responses (outputs), and test prompts. The total amount of training data is roughly 750,000 tokens, split over exactly 1,000 sequences.



Evaluations – Setup

- Select Baselines – Alpaca 65B, Bard, Claude, DaVinci003 and GPT-4
- Generation – Single response, limited to 2048 tokens
- Annotators – Prompt and 2 possible responses from different models
- (Tie-discounted) Points for Inter-Annotator Agreement
- e.g.
  - Author-Author
  - Author-Crowd
  - Crowd-Crowd

  1- Both agree
  ½ - Either one assigns a tie
  0 – Disagree, no ties

Evaluations – Preference Evaluation

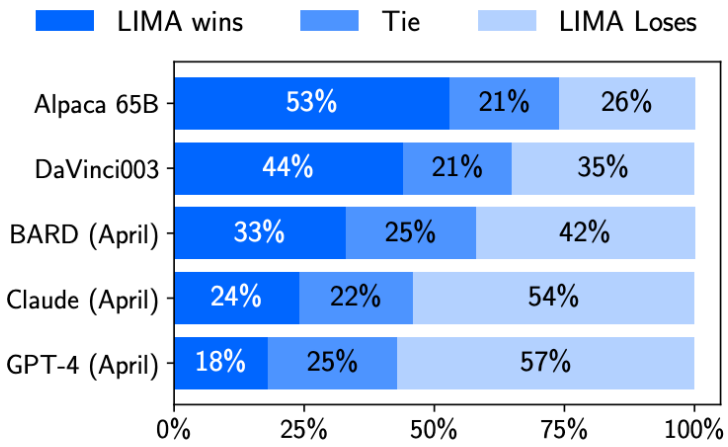


Figure 1: Human preference evaluation, comparing LIMA to 5 different baselines across 300 test prompts.

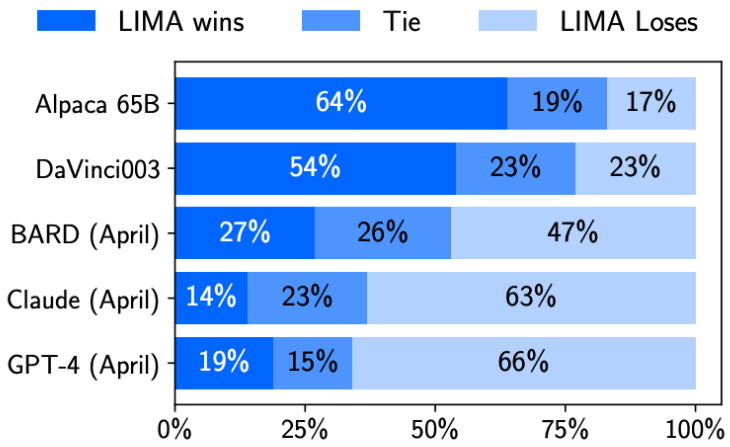


Figure 2: Preference evaluation using GPT-4 as the annotator, given the same instructions provided to humans.

- Diminishing preference for LIMA going from Alpaca65B to GPT-4

Absolute assessment (50 random examples)

Quality of the response – did not meet requirements of prompt vs excellent response

Safety – Response to sensitive prompts

- Responds safely to 80%
- Refuses to perform task
- Unsafe response

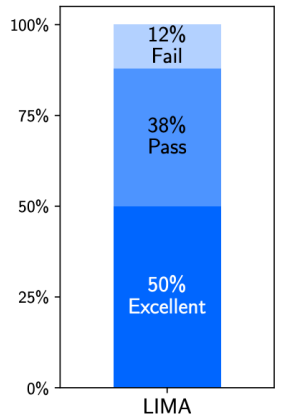


Figure 3: Analysis of LIMA over 50 test prompts.

Ablation Studies

Evaluate 5 responses from test set prompt on a 1-6 Likert scale through GPT-4
Prompt example:

You are evaluating a response that has been submitted for a particular task, using a specific set of standards. Below is the data:

[BEGIN DATA]
***
[Task]: {task}
***
[Submission]: {submission}
***
[Criterion]: helpfulness:
"1": "Not helpful - The generated text is completely irrelevant, unclear, or incomplete. It does not provide any useful information to the user."
"2": "Somewhat helpful - The generated text has some relevance to the user's question, but it may be unclear or incomplete. It provides only partial information, or the information provided may not be useful for the user's needs."
"3": "Moderately helpful - The generated text is relevant to the user's question, and it provides a clear and complete answer. However, it may lack detail or explanation that would be helpful for the user."
"4": "Helpful - The generated text is quite relevant to the user's question, and it provides a clear, complete, and detailed answer. It offers additional information or explanations that are useful for the user. However, some of the points of the response are somewhat repetitive or could be combined for greater clarity and concision"
"5": "Very helpful - The generated text is highly relevant to the user's question, and it provides a clear, complete, and detailed answer. It offers additional information, explanations, or analogies that are not only useful but also insightful and valuable to the user. However, the structured of the response is not well-organized and there is no clear progression or logical sequence of different points in the response."
"6": "Highly helpful - The generated text provides a clear, complete, and detailed answer. It offers additional information or explanations that are not only useful but also insightful and valuable to the user. The response is also in a logical and easy-to-follow manner by explicitly using headings, bullet points, or numbered lists to break up the information and make it easier to read."
***
[END DATA]

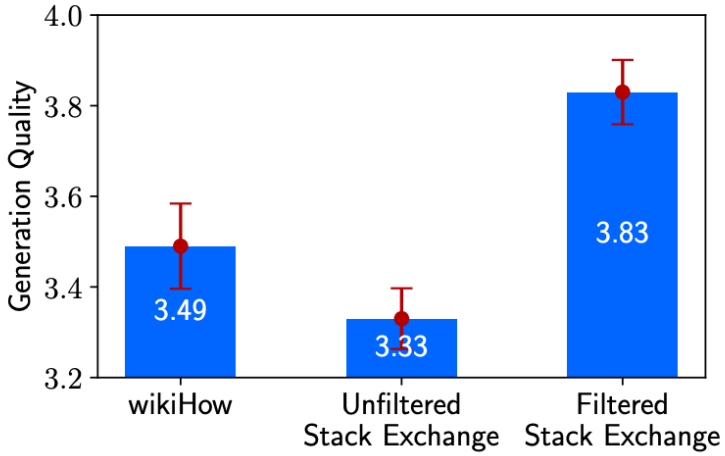
Does the submission meet the criterion? First, write out in a step by step manner your reasoning about the criterion to be sure that your conclusion is correct. Avoid simply stating the correct answers at the outset. Then print the choice only from "1, 2, 3, 4, 5, 6" (without quotes or punctuation) on its own line corresponding to the correct answer. At the end, repeat just the selected choice again by itself on a new line.

Figure 12: Prompt for ChatGPT evaluation with a 6-scale Likert score. The placeholders "task" and "submission" will be replaced by specific details from the actual case being evaluated.

Diversity, Data Quantity, Multi-Turn Dialogue

2000 training examples from sources with homogeneous vs heterogeneous prompts

Diverse prompts (with filtering) from SE data do better on test prompts compared to unfiltered and wikiHow data, increasing quality filtered data does not improve generation quality.



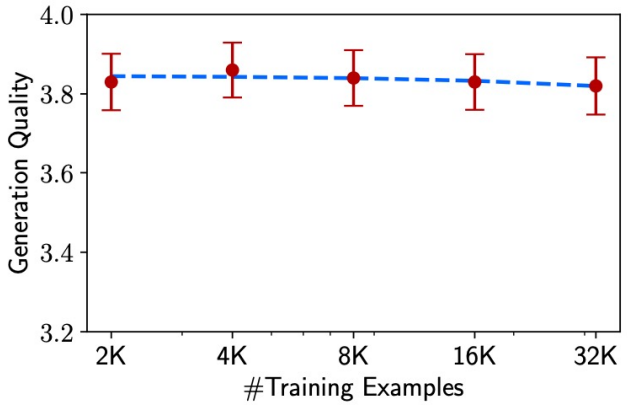


Figure 6: Performance of 7B models trained with exponentially increasing amounts of data, sampled from (quality-filtered) Stack Exchange. Despite an up to 16-fold increase in data size, performance as measured by ChatGPT plateaus.

Fine-tuned model from original 1000 examples + 30 multi-turn dialogues

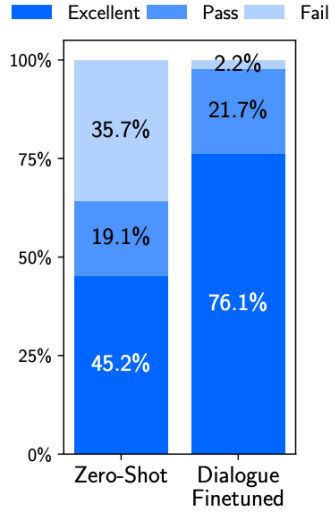


Figure 7: Analysis of dialogue turns, averaged over 10 test chats.

Criticisms

- Inclusion of response evaluation from the pre-trained model without fine-tuning on the 1000 examples would suggest a better picture of its contribution

- Homogeneity vs Heterogeneity is presented as the diversity of prompt styles rather than the diversity of the content itself (they considered STEM categories on SE and article categories on wikiHow – there may not be overlap between the two so the performance may depend on the category of tasks in the test set rather than the prompt styles used)

Other Methods: Synthetic Data Generation

Synthetic Data For Fine-Tuning - https://eugeneyan.com/writing/synthetic/

Self-Instruct – Non-finetuned models classify and filter responses to test tasks - https://arxiv.org/abs/2212.10560

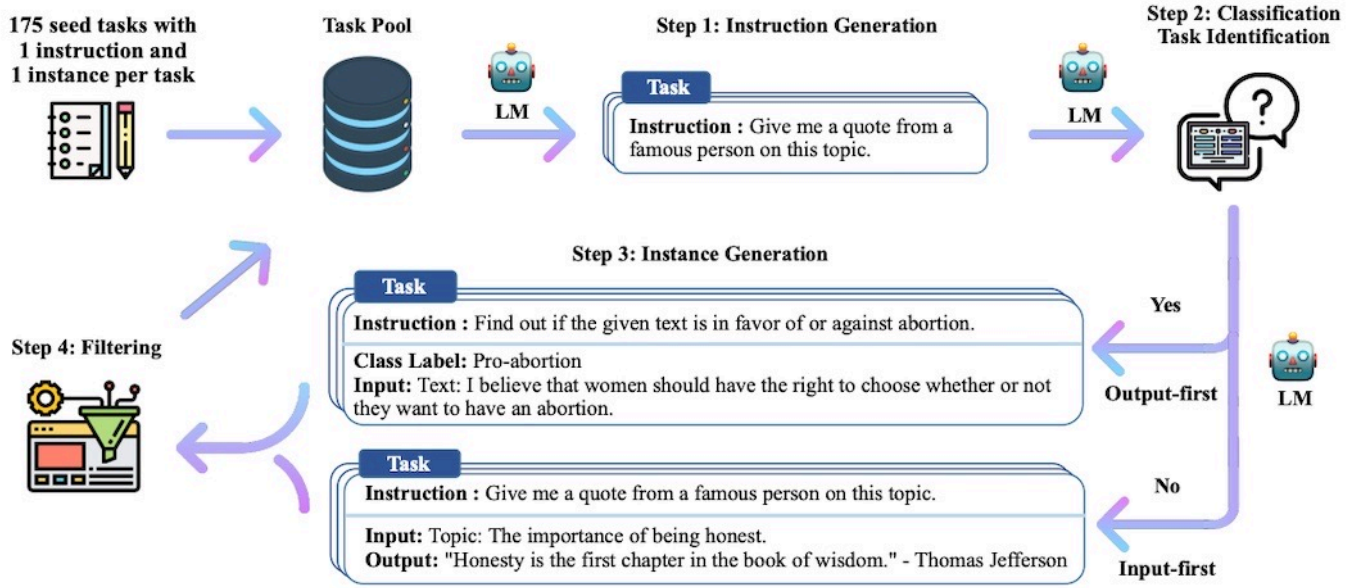


Figure 2: A high-level overview of SELF-INSTRUCT. The process starts with a small seed set of tasks as the task pool. Random tasks are sampled from the task pool, and used to prompt an off-the-shelf LM to generate both new instructions and corresponding instances, followed by filtering low-quality or similar generations, and then added back to the initial repository of tasks. The resulting data can be used for the instruction tuning of the language model itself later to follow instructions better. Tasks shown in the figure are generated by GPT3.

Other Methods: Instruction Backtranslation

https://arxiv.org/abs/2308.06259

Generate instructions given a human-written target response (so-called ”backwards” model) – In subsequent stages, finetuned models score instruction-response pairs

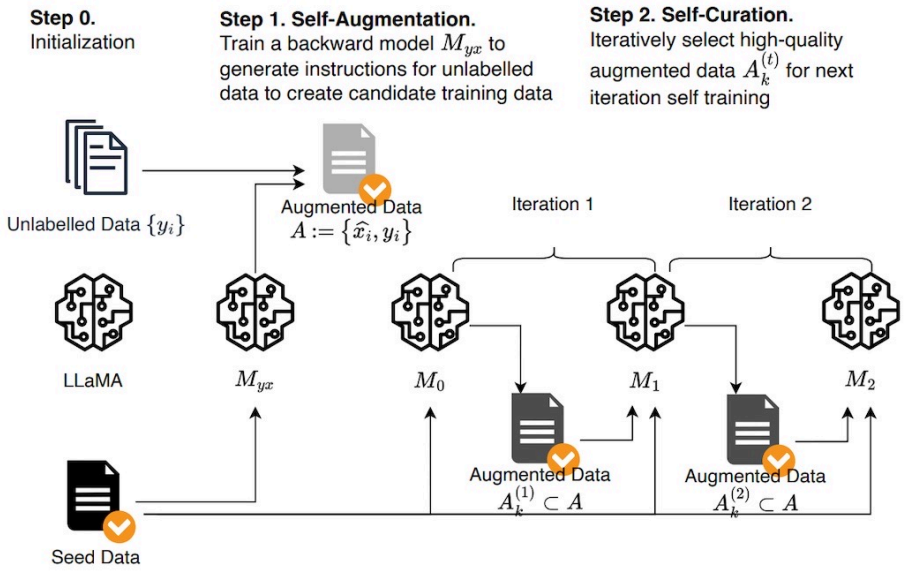


Figure 1: An overview of our **instruction backtranslation** method. We start from a base language model, e.g. LLaMa, a small amount of seed examples of (instruction, output) pairs, and a collection of unlabelled documents which are considered candidate outputs for unknown instructions. **Self-augmentation**: the base model is finetuned with (output, instruction) pairs from the seed examples as an instruction prediction model M_{yx}, which is used to generate candidate instructions for outputs from the unlabelled data. **Self-curation**: starting from an intermediate instruction-following model M_0 finetuned from seed examples only, it selects high-quality (instruction, output) pairs $\mathcal{A}_k^{(1)}$ from the candidates from the previous step, and uses them as finetuning data for the next intermediate model M_1, which is in turn used to select training data for obtaining M_2.

Other Methods: Dataset Selection with Datamodels (DsDm)

$$S^* := \underset{S \subset \mathcal{S}, |S|=k}{\arg\min} \mathcal{L}_{\mathcal{D}_{\text{targ}}}(S),$$

$$\text{where } \mathcal{L}_{\mathcal{D}}(S) := \mathbb{E}_{x \sim \mathcal{D}} \left[\ell(x; \mathcal{A}(S)) \right],$$

https://arxiv.org/abs/2401.12926

Choose size-k subset from pool of training data for minimizing expected loss from learning algorithm A over target task

Since directly optimizing the above is infeasible, learn parameters of a “scoring” function that can predict model loss over distribution of training subsets.

$$\tau_{\theta_x} : \{0,1\}^{|\mathcal{S}|} \to \mathbb{R}, \qquad \text{where} \qquad \theta_x = \arg\min_{\theta} \widehat{\mathbb{E}}^{(m)}_{S_i \sim \mathcal{D}_{\mathcal{S}}} \left[L_{\text{reg}} \left(\tau_\theta(\mathbb{1}_{S_i}),\ \mathcal{L}_x(S_i) \right) \right]$$

Thank you!

Backup

Discuss the following in some more detail:

- Fine-tuning protocol – residual dropout, hyperparameter selection
- Likert scale
- Nucleus sampling and repetition penalty for previously generated tokens
- Superficial Alignment
- Other definitions of alignment