Supervised Finetuning (SFT)

Alignment Tuning

LLMs are great for many NLP tasks. However they sometimes exhibit **unintended behaviour:**

- False information
- Inaccurate objectives
- Harmful, misleading, and biased expressions

It lacks **consideration of human values or preferences**

Alignment Tuning: Criteria

Helpfulness:

- Solving tasks/answering questions **concisely and efficiently**
- Might elicit **additional relevant information**
- Challenging since it's difficult to precisely **define and measure intention of users**

Honesty:

- Present **accurate content**
- No **fabricated information**
- Convey **degrees of uncertainty** in it's output
- **"Know unknowns"**(know about your levels of knowledge)

Harmlessness:

- Not **offensive nor discriminatory**
- Detect requests for **malicious purposes**
- Refuse **dangerous actions**

Alignment Tuning: Collecting Human Feedback

- **Human Labeler Selection:** filter labelers by assessing agreement between human labeler and researcher
- **Human Feedback Collection:**
  - **Ranking-based**
  - **Question-based**
  - **Rule-based**

Alignment Tuning: RLHF

- **Reinforcement Learning from Human Feedback**
- 3 components:
  - **Pre-trained LM** to be aligned (generative model with existing pre-trained parameters)
  - **Reward model** learning from human feedback: fine-tuned LM or LM trained de novo using human preference data
  - **RL algorithm** training the LM

Alignment Tuning: RLHF Key Steps

- Supervised Fine-tuning
  - Supervised dataset containing **input prompts** and **desired outputs**
- Reward Model Training
  - Train with human feedback data
- RL Fine-tuning
  - Reward/Penalty model based on divergence between initial LM and current output

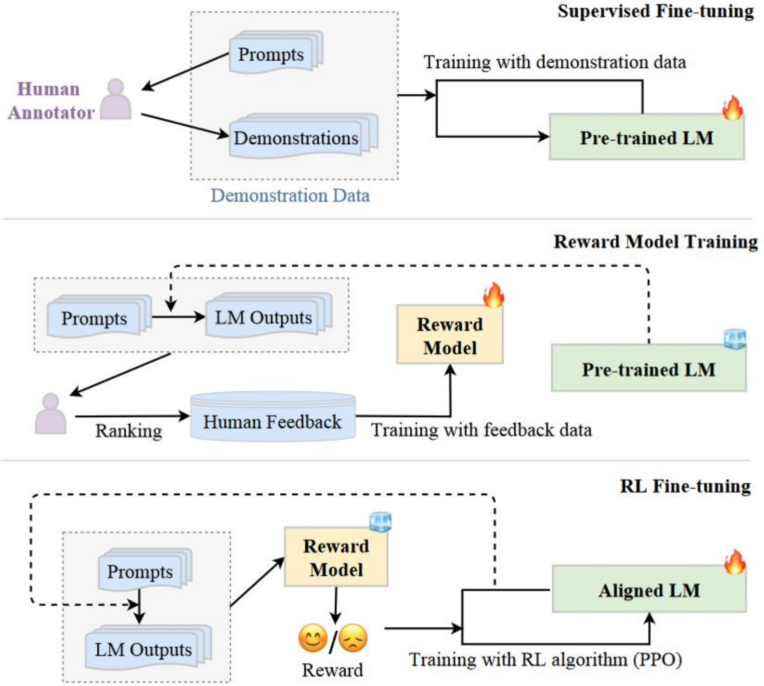


Fig. 12: The workflow of the RLHF algorithm.

Alignment Tuning: Alignment without RLHF

- Problems:
  - RLHF needs to train multiple LMs including **model being aligned, reward model, reference model**
  - Commonly used **PPO** RL-algorithm is complex and sensitive to hyper-parameters
- Alternative: supervised fine-tuning **without reinforcement learning**
  - **Supervised learning** on high-quality **alignment dataset**
  - Assuming that golden rules are integrated in the alignment dataset
- 2 issues:
  - Alignment Data Collection
  - Supervised Alignment Tuning

Alignment Tuning: Alignment without RLHF

Alignment Data Collection

- Construction of alignment data
  - **Align LLM-behaviour** with human preferences
- Reward model based approaches
  - Leverage **existing reward models**
- LLM based generative approaches
  - Powerful LLMs to **generate human-aligned data**
- LLM based interactive approaches
  - **Simulated interaction** with number of LLM agents
  - Central agent **revises original response** based on the suggestions from the other agents

Alignment Tuning: Alignment without RLHF

Supervised Alignment Tuning

- **Primary training objective**
  - Primary training loss is still the traditional cross-entropy loss for sequence-to-sequence learning
  - **CoH:** prepend **"helpful answer"** and **"unhelpful answer"** to responses
  - Only compute losses with special masking
- **Direct preference optimization**
  - **Reparameterize** the response rewards using the policy model
  - Original reward modeling objective can be **reformulated** only based on the policy model.
- **Auxiliary optimization objectives**
  - The ranking loss can be used to train the model to **preserve the ranking order of these responses**
  - Contrastive learning to **push up the probability of correct instruction-response pairs** while **pushing down incorrect instruction-response pairs**

LIMA (Less Is More for Alignment)

Superficial Alignment Hypothesis

- Knowledge and capabilities are learned in pre-training
- Small set of examples enough for tuning
- Model needs to learn the subdistribution to use

Experiment

- LIMA, 65B parameter LLM
- 1000 examples of “helpful AI Assistant”
    - 750 Stack Exchange and wikiHow -> High Quality and diversity
    - 250 Manually authored -> task diversity and maintaining a uniform response style

2 Tests

- Human evaluation
- GPT-4 as judge

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.

Training

- 15 epochs
- AdamW with $\beta 1 = 0.9$, $\beta 2 = 0.95$, weight-decay = 0.1
- Batch size of 32
- 2048 tokens per text
- Residual dropout p=0.0 for bottom layer, up to p=0.3 for last

Evaluation

- Each model creates a response for each prompt
  - Nucleus sampling with $p = 0.9$ and a temperature of $\tau = 0.7$

- Annotators rate responses

Evaluation

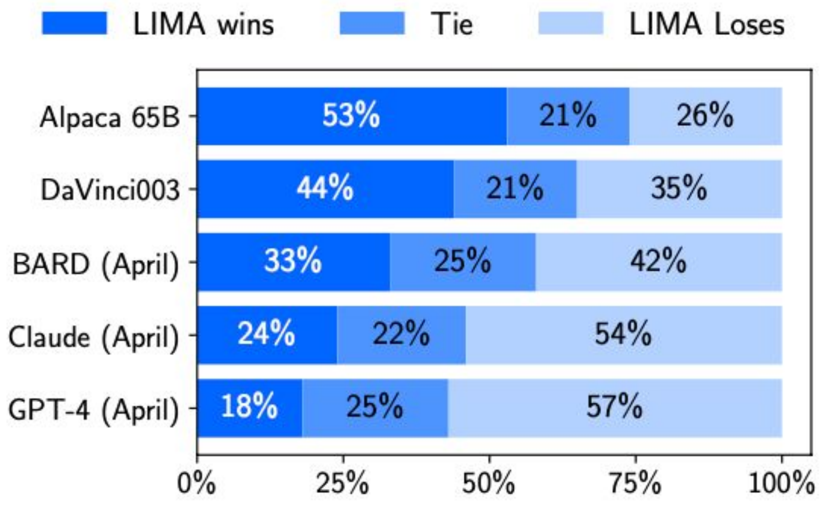


Figure 1: Human preference evaluation, comparing LIMA to 5 different baselines across 300 test prompts.

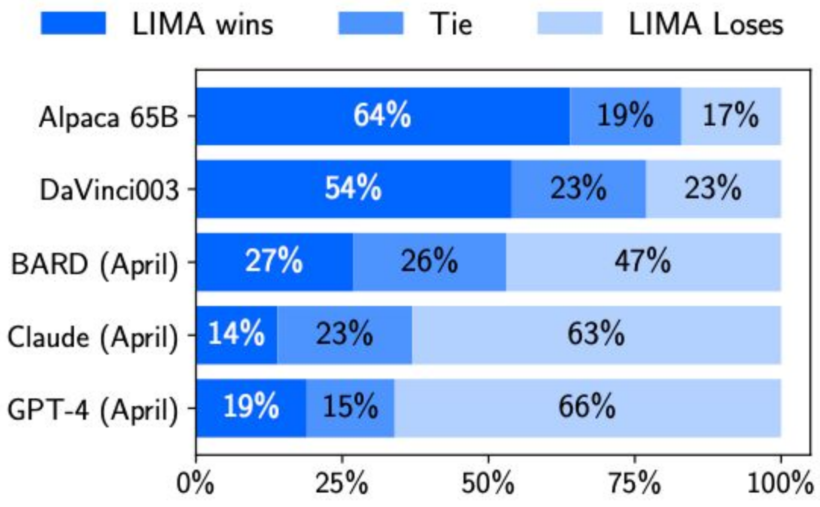
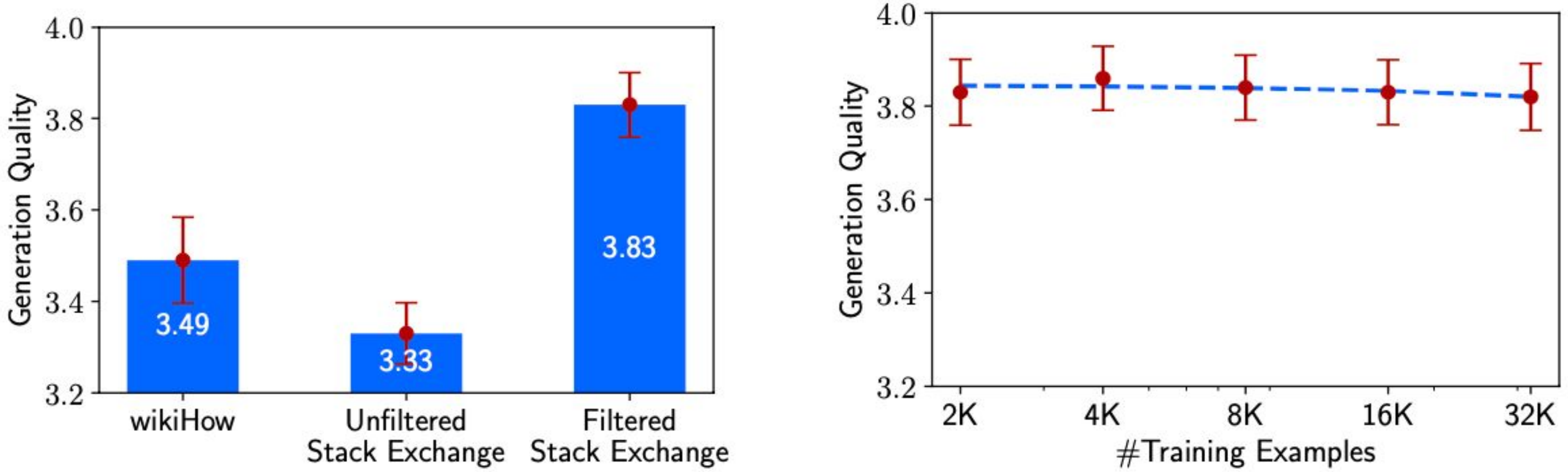


Figure 2: Preference evaluation using GPT-4 as the annotator, given the same instructions provided to humans.



Generation Quality
4.0
3.8
3.6
3.4
3.2
3.49
3.33
3.83
wikiHow
Unfiltered Stack Exchange
Filtered Stack Exchange
Generation Quality
4.0
3.8
3.6
3.4
3.2
2K
4K
8K
16K
32K
#Training Examples

- 50 tests
    - 43 are similar to tests in training data
- Only “slightly” worse for out of distribution
    - Small study
- No clear definition of Fail, Pass, Excellent

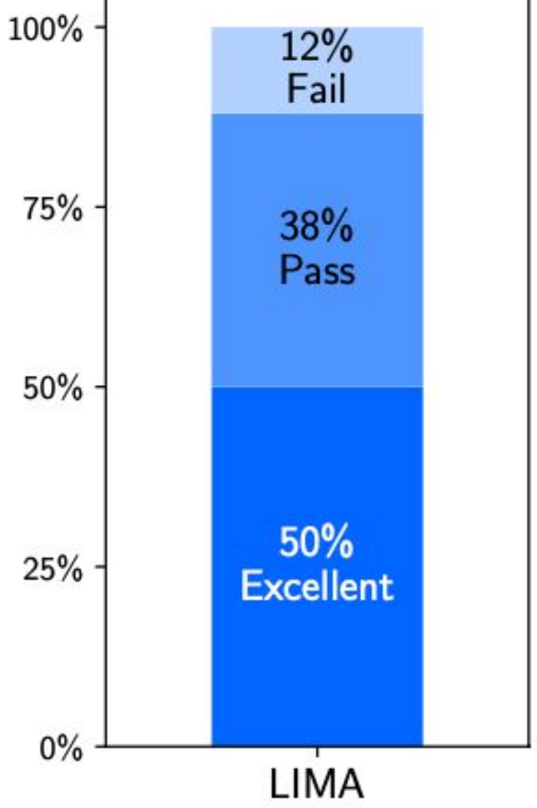


Figure 3: Analysis of LIMA over 50 test prompts.

Advantages

- "Relative" good performance after training on few but diverse prompts
- Able to generalize well

Limits

- After 3 interactions in 6/10 interactions LIMA fails to follow the prompt
    - Adding Dialogue Data to it improved it
- Mental effort to manually sort/filter and craft examples
    - Difficult to scale
- Not as robust as product-grade models
    - “Unlucky sample” during decoding or adversarial prompt can lead to weak response
- It loses in the benchmarks for comparably sized models
- Data leak in annotation groups before labeling