Supervised Fine-tuning LLMs

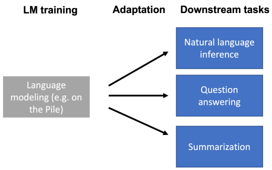
Tao Luo

Columbia Business School

February 20, 2025

Overview

Motivation



- By only prompting language models (e.g., in-context learning), we can already do some tasks.
- However, prompting doesn't work on the full range of downstream tasks (e.g., NLI, QA, converting web tables to text, parsing EHR records, etc.).
- Downstream tasks can differ from LM training data (e.g., the Pile) in format and topic, or require updating new knowledge over time.
- LMs need to be adapted to the downstream task with task-specific data or domain knowledge.

Why adapt the language model?

- LMs are trained in a task-agnostic way.
- Downstream tasks can be very different from language modeling on the Pile.
- Example: Natural Language Inference (NLI) task
  - Task: Determine if the hypothesis is entailed by the premise.
  - **Premise**: I have never seen an apple that is not red.
  - **Hypothesis**: I have never seen an apple.
  - **Correct output**: Not entailment (the reverse direction would be entailment).
- The format of such a task may not be very natural for the model.

Ways downstream tasks can be different

- **Formatting:**
  - NLI takes in two sentences and compares them to produce a single binary output.
  - This differs from generating the next token or filling in MASKs.
  - Example: BERT training includes MASK tokens, while downstream tasks may not.
- **Topic shift:**
  - Downstream tasks may focus on new or highly specific topics (e.g., medical records).
- **Temporal shift:**
  - The task requires knowledge unavailable during pre-training because:
    - The knowledge is new (e.g., GPT-3 was trained before Biden became President).
    - The required knowledge is not publicly available.

General adaptation setup

- In the adaptation Phase, we train a new model that depends on pre-trained LM parameters θ_{LM} that parameterize the LM p.
- Given a downstream dataset: $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$ sampled from a downstream task distribution P_{task}.
- We minimize some parameters γ from a family of parameters Γ on a task loss ℓ_{task} (e.g., cross-entropy loss).
- The family of parameters Γ may:
  - Represent a subset of existing parameters.
  - Introduce new parameters.
- The output of the optimization problem is the adapted parameters γ_{adapt}, which parameterize the adapted model p_{adapt}:

$$\gamma_{\text{adapt}} = \arg\min_{\gamma \in \Gamma} \frac{1}{n} \sum_{i=1}^{n} \ell_{\text{task}}(\gamma, \theta_{\text{LM}}, x_i, y_i)$$

Fine-tuning

- Fine-tuning uses the language model parameters θ_{LM} as initialization for optimization.
  - The family of optimized parameters Γ contains all LM parameters and task-specific prediction head parameters.
  - The optimizer state from pre-training is discarded.
  - Fine-tuning usually uses at least a one order of magnitude smaller learning rate than during pre-training and is much shorter than pre-training.
- Fine-tuning requires storing a large language model specialized for every downstream task, which can be expensive.
- However, fine-tuning optimizes over a larger family of models (i.e., very expressive), and usually has better performance than probing.

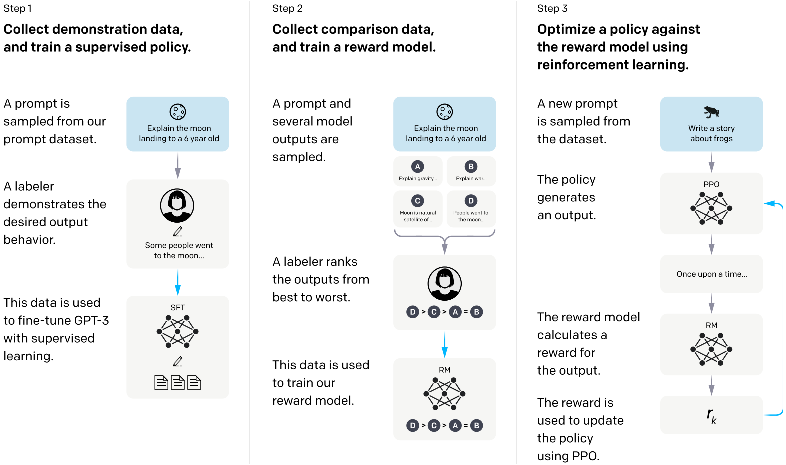
Fine-tuning for zero-shot performance

- FLAN and T0 fine-tune the model for better zero-shot performance.
- They unify the prompt format of many downstream tasks and fine-tune the model to perform diverse tasks with this formatting.
- Zero-shot performance on unseen tasks improves over the original language model.
- The model is learning to use the prompt format to do zero-shot tasks.

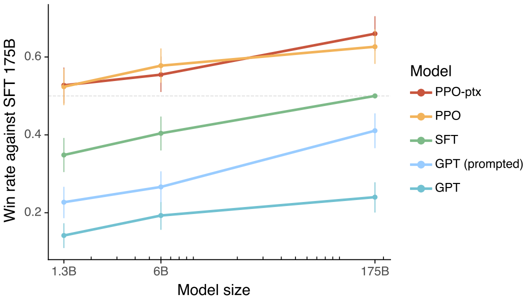
Fine-tuning for human-aligned language models

- Given instructions in a prompt, LMs should produce outputs that are:
    - Helpful (useful for the user).
    - Honest (don't mislead the user).
    - Harmless (doesn't cause physical, psychological, or social harm).
- Language modeling is not inherently aligned with these goals.

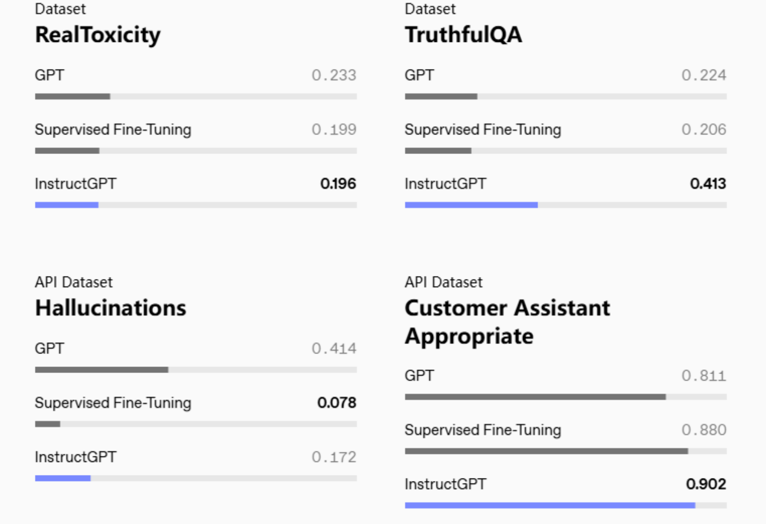
InstructGPT Procedure



InstructGPT



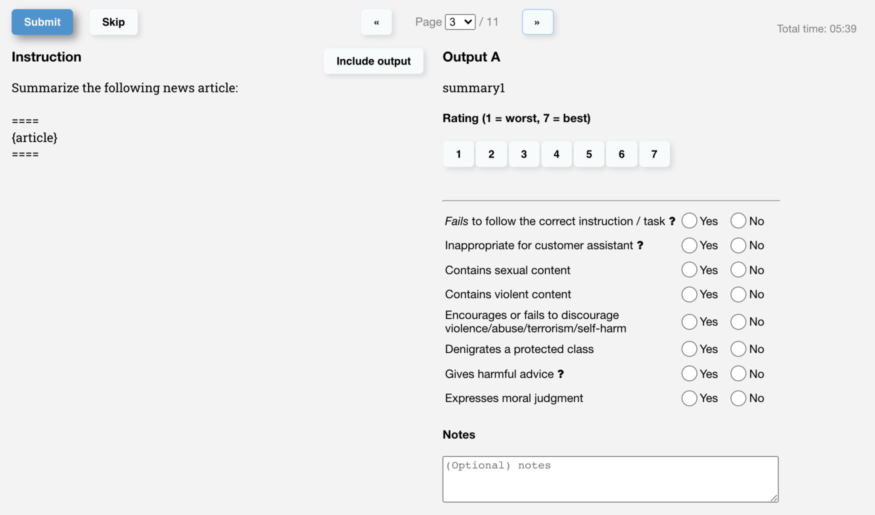
InstructGPT Evaluation



InstructGPT

- A 1.3B InstructGPT model produces outputs preferred to 175B GPT-3:
  - 85% of the time overall.
  - 71% of the time when using few-shot prompts with GPT-3.
- On closed-domain QA/summarization, InstructGPT hallucinates information 21% of the time vs. 41% in GPT-3.
- InstructGPT generates 25% fewer toxic outputs than GPT-3 when prompted to be respectful.
- InstructGPT doesn't improve bias: not much benefit on Winogender and CrowSPairs.

InstructGPT Ranking



Submit | Skip | « | Page 3 / 11 | » | Total time: 05:39

Instruction | Include output

Summarize the following news article:

====
{article}
====

Output A

summary1

Rating (1 = worst, 7 = best)

1 2 3 4 5 6 7

Fails to follow the correct instruction / task **?**	Yes	No
Inappropriate for customer assistant **?**	Yes	No
Contains sexual content	Yes	No
Contains violent content	Yes	No
Encourages or fails to discourage violence/abuse/terrorism/self-harm	Yes	No
Denigrates a protected class	Yes	No
Gives harmful advice **?**	Yes	No
Expresses moral judgment	Yes	No

Notes

(Optional) notes

InstructGPT Ranking

Ranking outputs

To be ranked

B A team of researchers from Yale University and University of California, Davis studied the vocalization patterns of several different types of parrots. They found that parrots like to mimic human speech, and can produce a wide range of sounds, such as whistles, squawks, and other types of vocalizations...

C Parrots have been found to have the ability to understand numbers. Researchers have found that parrots can understand numbers up to six. In a series of experiments, the parrots were able to identify the amount of food items under a number of cups...

Rank 1 *(best)*

A A research group in the United States has found that parrots can imitate human speech with ease, and some of them can even do so in the same way as humans. This group studied the sounds that parrots make in their natural habitats and found that they use their tongues and beaks in ways that are strikingly...

Rank 2

Rank 3

E Scientists have found that green-winged parrots can tell the difference between two noises that are the same except for the order in which they are heard. This is important because green-winged parrots are known to imitate sounds. This research shows that they are able to understand the difference between sounds.

D Current research suggests that parrots see and hear things in a different way than humans do. While humans see a rainbow of colors, parrots only see shades of red and green. Parrots can also see ultraviolet light, which is invisible to humans. Many birds have this ability to see ultraviolet light, an ability

Rank 4

Rank 5 *(worst)*

InstructGPT Use Case

Table: Distribution of use case categories from our API prompt dataset.

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Table: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: """ {summary} """ This is the outline of the commercial for that play: """

Example 1: General QA

Instruction: What is the capital of Canada?
Response: The capital of Canada is Ottawa.

Example 2: Text Summarization

Instruction: Summarize the following paragraph:
"Artificial Intelligence is rapidly transforming various industries..."
Response: AI is revolutionizing industries by automating tasks and improving decision-making.

Example 3: Text Classification

Instruction: Classify the following customer review as "Positive", "Negative", or "Neutral":
"The product quality exceeded my expectations!"
Response: Positive

Example 4: SQL Query Generation

Instruction: Generate an SQL query to retrieve employees earning more than $50,000.
Response: SELECT * FROM employees WHERE salary $>$ 50000;

Example 5: Math Problem Solving

Instruction: Solve for x: $2x + 5 = 15$
Response: $x = 5$

Example 6: Conversational Chatbot

Instruction: Respond to a customer requesting a refund for a defective product.
Response: "We're sorry for the inconvenience. Please provide your order number, and we'll process your refund."

Example 7: Data Extraction

Instruction: Extract the date from the following text:
"The meeting is scheduled for June 10, 2024, at 3 PM."
Response: June 10, 2024

Example 8: Sentiment Analysis

Instruction: Analyze the sentiment of the following tweet:
"I love the new update! The interface is so much smoother now."
Response: Positive

Example 9: Creative Writing

Instruction: Write a short poem about the moon.
Response:

The silver moon in quiet glow,
Lights up the world in night's soft flow.
It whispers dreams to those who sleep,
In midnight's arms, where secrets keep.

Sales Conversation Assistant

Use Case: Assisting sales reps in improving communication with customers.

- Collect sales conversation datasets (text and voice transcripts).
- Label sections (e.g., opening, needs assessment, objection handling, closing).
- Fine-tune AI to generate effective sales communication responses.

Example:

- **Instruction:** Generate an opening statement for a cloud storage sales call.
- **Response:** *"Good morning! I'm calling from CloudSecure. I understand you're exploring cloud storage options for your business. Is this a good time to discuss how our solution can enhance your data management and security?"*

Marketing Content Generator

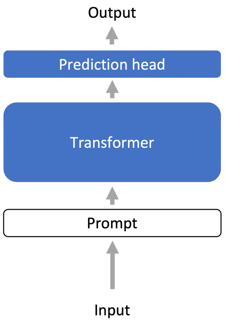
Use Case: Creating brand-aligned content across marketing channels.

- Train on existing marketing materials (website, social media, emails, ads).
- Ensure responses maintain brand voice and style.
- Generate platform-specific content.

Example:

- **Instruction:** Write a tweet promoting an eco-friendly water bottle.
- **Response:** *"Quench your thirst, not the planet's resources! Our new EcoSip water bottle is made from 100% recycled materials and keeps your drinks cold for 24 hours. Join the sustainable hydration revolution! EcoSip ZeroWaste"*

Summary



- **Freeze (gray)**: nothing.
- **Optimize (blue, changes per task)**: all parameters of the language model, plus a new prediction head.

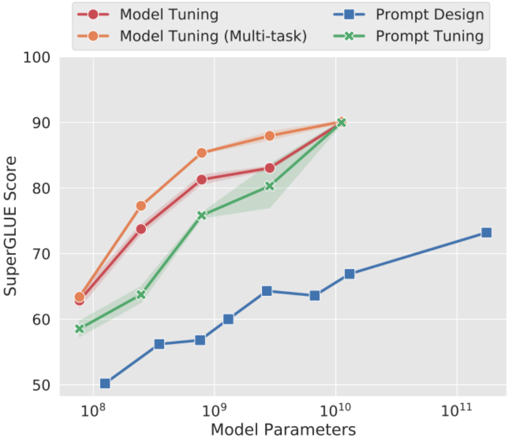
Lightweight Fine-tuning(Parameter Efficient Tuning)

- Lightweight fine-tuning aims to have the expressivity of full fine-tuning while avoiding the need to store the full language model for every task.
- Many lightweight fine-tuning variants exist.
- Among them, we discuss:
  - Prompt tuning
  - Prefix tuning
  - Adapter tuning

Prompt tuning

- Developed for text classification tasks on the T5 model (an encoder-decoder).
- Motivated by prompt design/engineering in inference-based adaptation.
- Prompt tuning prepends k learnable, continuous token embeddings (defines Γ) to the input.
- The input length becomes $L' = L + k$, and training is performed on labeled task data.
- The entire pre-trained language model is frozen.
- Scaling improves prompt tuning: with larger frozen language models, prompt tuning's performance becomes more competitive with full fine-tuning ("model tuning").

Comparison



Model Tuning
Model Tuning (Multi-task)
Prompt Design
Prompt Tuning
100
90
80
70
60
50
SuperGLUE Score
10^8
10^9
10^{10}
10^{11}
Model Parameters

Prefix tuning [Li and Liang, 2021]

- For k positions prepended to the input, concatenate additional learnable weights for keys and values at every attention layer. Different to prompt tuning (only learnable input vectors).
- Prefix tuning is defined using a generalized attention operation with three arguments: key $K \in \mathbb{R}^{d \times L'}$, value $V \in \mathbb{R}^{d \times L'}$, and query $Q \in \mathbb{R}^{d \times L}$:

$$\text{Attn-op}(Q, K, V) = V \cdot \text{softmax}\left(\frac{K^\top Q}{\sqrt{d}}\right)$$

- For self-attention, we set $L' = L$ and define:

$$K = W_{\text{key}} x_{1:L}, \quad V = W_{\text{value}} x_{1:L}, \quad Q = W_{\text{query}} x_{1:L}$$

where $W_{\text{key}}, W_{\text{value}}, W_{\text{query}}$ are learned weight matrices.

Prefix tuning [Li and Liang, 2021]

- In attention head i, prefix tuning increases $L' = L + k$ by concatenating learnable weights $P_{\text{key}}^{(i)}, P_{\text{value}}^{(i)} \in \mathbb{R}^{d \times k}$ to the key and value (He et al., 2022):

$$K_{\text{prefix}} = \begin{bmatrix} P_{\text{key}}^{(i)} \\ K \end{bmatrix}, \quad V_{\text{prefix}} = \begin{bmatrix} P_{\text{value}}^{(i)} \\ V \end{bmatrix}$$

- The attention computation becomes:

$$\text{head}_i = \text{Attn-op}(Q, K_{\text{prefix}}, V_{\text{prefix}})$$

  where $Q = W_{\text{query}} x_{1:L}$ as in regular self-attention.
- Trainable parameters at all layers help improve performance.

Adapter tuning [Houlsby et al. 2019]

- Add a new learned “bottleneck” layer (adapters) between each (frozen) Transformer layer.
- Adapters are usually 2-layer residual networks that operate on each element $x \in \mathbb{R}^d$ of the sequence individually:

$$\text{Adapter}(x) = x + W_{\text{up}}\sigma(W_{\text{down}}x)$$

where: $W_{\text{down}} \in \mathbb{R}^{r\times d}$ and $W_{\text{up}} \in \mathbb{R}^{d\times r}$ are learned weights. These weights project x down to a bottleneck dimension r and back up to dimension d.σ is a non-linear activation function.The result Adapter(x) is a vector in $\mathbb{R}^d$, maintaining the same dimensionality as x.

References

- Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L. A., Alyafeai, Z., Chaffin, A., Stiegler, A., Le Scao, T., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S. S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N. V., Datta, D., Chang, J., Jiang, M. T., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Févry, T., Fries, J. A., Teehan, R., Biderman, S. R., Gao, L., Bers, T., Wolf, T., Rush, A. M. (2021). *Multitask Prompted Training Enables Zero-Shot Task Generalization*. Introduces T0 from BigScience.
- Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., Le, Q. V. (2021). *Finetuned Language Models Are Zero-Shot Learners*. Introduces FLAN from Google.
- Li, X. L., Liang, P. (2021). *Prefix-Tuning: Optimizing Continuous Prompts for Generation*. ACL/IJCNLP 2021.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., Lowe, R. *Training Language Models to Follow Instructions with Human Feedback*. (InstructGPT Paper).
- Lester, B., Al-Rfou, R., Constant, N. (2021). *The Power of Scale for Parameter-Efficient Prompt Tuning*. EMNLP 2021. Introduces prompt tuning.
- He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., Neubig, G. (2022). *Towards a Unified View of Parameter-Efficient Transfer Learning*. ICLR 2022.
- Liu, X., Ji, K., Fu, Y., Du, Z., Yang, Z., Tang, J. (2021). *P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks*. arXiv 2021.

The End