

Zero-shot Learning / Prompting

- Give some inputs (x_1, \dots, x_t) (which can be questions + instructions) to the LLMs
- LLMs generate a sequence of tokens $x_{t+1}, x_{t+2}, \dots, x_T$ given (x_1, \dots, x_t)

The screenshot displays the OpenAI Playground interface. On the left, a sidebar titled "Get started" provides instructions and a "KEEP IN MIND" section with three points: using good judgment when sharing outputs, the non-use of API requests for model training, and the 2021 training data cutoff. The main area, labeled "Playground", contains two prompts and their completions. The first prompt is "What does 1+3 equal to?" with the completion "1+3 equals 4.". The second prompt is "Could you help translate the following sentence into Chinese: 'CS229 is a machine learning course'." with the completion "'CS229是一门机器学习课程。'". On the right, a settings panel includes a "Load a preset..." dropdown, "Save", "View code", and "Share" buttons. Below these are sliders for "Mode" (set to "Complete"), "Model" (set to "text-davinci-003"), "Temperature" (set to 1), "Maximum length" (set to 256), "Stop sequences" (with a text input field), and "Top P" (set to 1).

Overview Documentation API reference Examples Playground

Help Personal

Get started × **Playground** Load a preset... Save View code Share ...

Enter an instruction or select a preset, and watch the API respond with a **completion** that attempts to match the context or pattern you provided.

You can control which **model** completes your request by changing the model.

KEEP IN MIND

- Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more.](#)
- Requests submitted to our API will not be used to train or improve future models. [Learn more.](#)
- Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

What does 1+3 equal to?

1+3 equals 4.

Could you help translate the following sentence into Chinese: "CS229 is a machine learning course".

"CS229是一门机器学习课程。"

Mode Complete

Model text-davinci-003

Temperature 1

Maximum length 256

Stop sequences Enter sequence and press Tab

Top P 1

Prompt Engineering

What's the sentiment of the following sentence.

"Large language models are very powerful models that can be used on a large range of applications"

Positive

What's the sentiment of the following sentence. Please answer with yes or no, where yes indicates positive and no indicate negative sentiment.

"Large language models are very powerful models that can be used on a large range of applications"

Yes

What's the sentiment of the following sentence. Please answer with yes or no, where yes indicates positive and no indicate negative sentiment.
Please also provide some explanations.

"Large language models are very powerful models that can be used on a large range of applications"

Yes, positive sentiment. This sentence is praising the capabilities of the model.

Please polish the following sentence.

"Large language models are very powerful models that can be used on a large range of applications"

"Large language models are incredibly powerful and can be applied to a wide range of applications."

Please expand the following sentence into a paragraph.



"Large language models are very powerful models that can be used on a large range of applications"

Large language models are an important tool in natural language processing systems. They are powerful models capable of performing a wide range of tasks, such as text generation, sentiment analysis, and language translation. In recent years, these models have become increasingly powerful due to advances in machine learning, deep learning, and natural language processing technologies. With the right architecture and training data, large language models can outperform other traditional language processing techniques. Due to their flexibility and capacity for computational power, large language models are applicable to many different areas such as data analysis, information retrieval, natural language understanding, and machine translation. Moreover, they can be deployed on cloud platforms or as a standalone application depending on the use case. Large language models are incredibly useful tools for analyzing and processing language data, and have been used to power a multitude of applications.

In-context learning

- Give LLMs $(x_1, y_1, x_2, y_2, \dots, x_k, y_k, x_{\text{test}})$ where (x_i, y_i) are exemplars, and x_{test} is a test example.

Playground

Load a preset...

Save

Q: 1 ~ 3 = ? ← x_1
A: 4 ← y_1
Q: 3 ~ 6 = ? ← x_2
A: 9
Q: 5 ~ 12 = ?
A: 17
Q: 1 ~ 11 = ? ← x_{test}
A: 12 ← y_{test}



In-context learning

- Give LLMs $(x_1, y_1, x_2, y_2, \dots, x_k, y_k, x_{\text{test}})$ where (x_i, y_i) are exemplars, and x_{test} is a test example.

Playground

Load a preset...



Q: 1 ~ 3 = ?

A: -2

Q: 3 ~ 6 = ?

A: -3

Q: 5 ~ 12 = ?

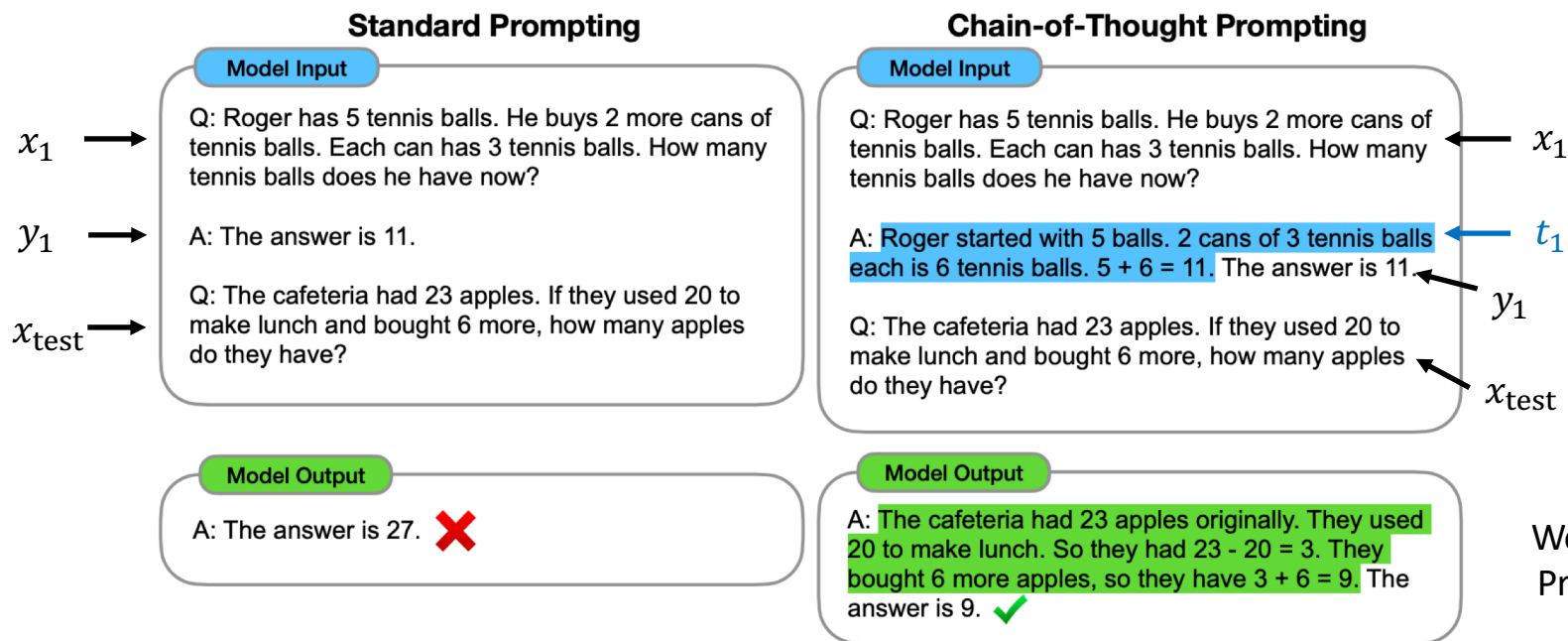
A: -7

Q: 1 ~ 11 = ?

A: -10

Few-shot Chain of Thoughts

- Instead of $(x_1, y_1, x_2, y_2, \dots, x_k, y_k, x_{\text{test}})$, use $(x_1, t_1, y_1, x_2, t_2, y_2, \dots, x_k, t_k, y_k, x_{\text{test}})$
 - where t_i are some thought process for answering the question



Wei et al.'22 Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Zero-shot CoT (Let's think step by step)

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 **X**

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓*

Instruct Tuning and RLH

- Methods to improve LLMs' capability of following instructions / [aligning](#) LLMs with user intents
- Step 1: collect datasets consists of (x, y) =(instructions/questions, answers) pairs, where answers are given by human labelers.
 - Fine-tune the model on the dataset
- Step 2: train a reward model to predict human's preferences of the answer (using human labels again)
 - $r(x, y) \in \mathbb{R}$: human's preferences on the answer y given question x
- Step 3: Use policy gradient to maximize $\mathbb{E}_{y \sim f_{\theta}(\cdot|x)}[r(x, y)]$
 - $\nabla \mathbb{E}_{y \sim f_{\theta}(\cdot|x)}[r(x, y)] = \mathbb{E}_{y \sim f_{\theta}(\cdot|x)}[r(x, y) \cdot \nabla \log f(y|x)]$