Prompt eng & RLHF

Practices & discussions

christine.fs@icloud.com/xinyuhu@microsoft.com

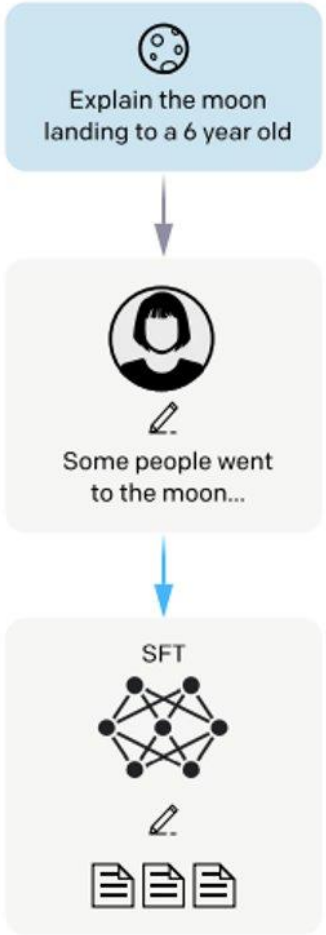
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



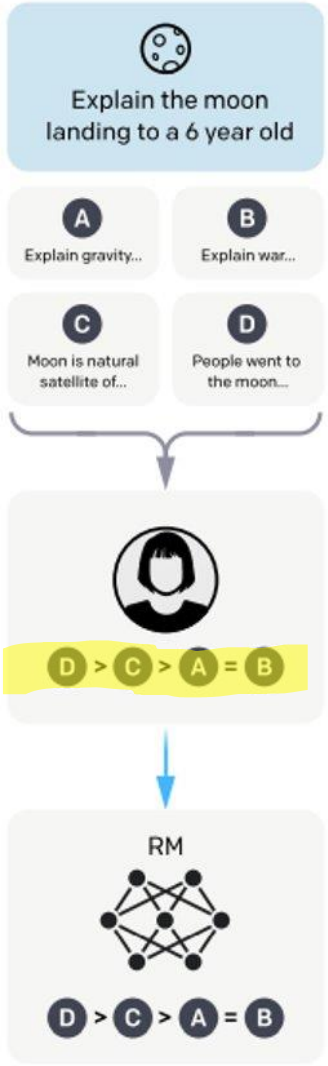
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

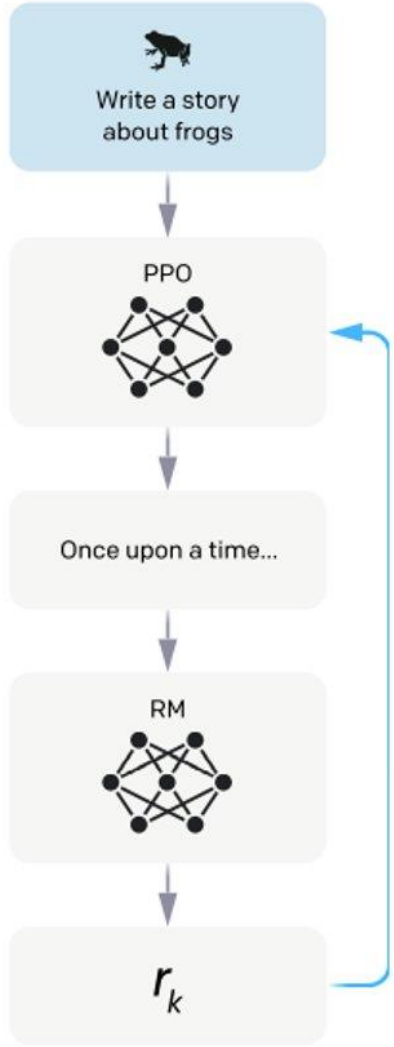
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy



RLHF and prompting

- RLHF reduces need for pre-trained prompt engineering
  - Scaling up instruction fine-tuning, which means that sometimes, you can "just ask" the model to perform a task without extensive prompt engineering. Previously, tasks like summarization or translation required tailored prompts, but with RLHF, users can often achieve desired results by simply asking.
- RLHF-tuned chatGPT succeeds on many previously hard problems [Douglas Hofstadter and David Bender in The Economist]
  - Reduces hallucination and undesired behavior
  - i.e. what do fried eggs (sunny side up) eat for breakfast

Brief history of prompt eng dev (I)

- zero-shot prompting: clever zero-shot prompts can beat few-shot examples
- GPT3 Codex: new capabilities from code training
- Align models with user intent instruction tuning
- RLHF: scaling up instruction finetuning -> zero-shot prose prompting, sometimes you can "just ask" (previously tasks that need prompt engineering, after this, we can just ask)
- Prompting with the "format trick": provide a **template** that the model uses to generate its completion
- Prompting through partial codes (PAL, synthetic prompting, etc.):
  - specify tasks using code prompts
  - prompting with imaginary variables

Brief history of prompt eng dev (II)

- Using an external interpreter to overcome model limitations in conversations
- Chain-of-thought prompting: examples of step-by-step reasoning
- Zero-shot CoT: **Let's think step by step.**
  - Better in math questions compared to few-shot, zero-shot, and the same as few-shot CoT but less writing efforts
  - use two step prompting: first prompt as reasoning extraction, second prompt as answer extraction
- self-consistency and consensus: uses three steps
  - prompt a language model using CoT
  - replace the greedy decode in CoT by sampling from the language model's output to generate a diverse set of reasoning paths
  - marginalize out the reasoning paths and aggregate by choosing the most consistent answer in the final answer set

Example: format trick

```
prompt = '''
Generate an arXiv pre-print with the given title.

Use this format:

Title:<Title>
## Abstract ##
<Text of abstract>
## Sections ##
<Numbered list of 10 top-level sections>
## Subsection outline ##
<Hierarchically numbered outline with 3-5 subsections for each section>
## Content ##
<Text of entire arXiv pre-print in LaTeX notation>

Title: An exploration of GPT3 finetuning with datasets in different domains and sizes
'''
openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  temperature=0.7,
  messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": prompt},
    ]
)
```

Example: format trick


```
<OpenAIObject chat.completion id=chatcmpl-6tjUzqkD4dN2KczDemU653kCOyvYL at 0x7fad456ccd10> JSON: {
  "choices": [
    {
      "finish_reason": "stop",
      "index": 0,
      "message": {
        "content": "Title: An exploration of GPT3 finetuning with datasets in different domains and sizes\n\n## Abstract ##\nIn this paper, we explore the effects of finetuning GPT3 with datasets of varying sizes and domains. We evaluate the performance of GPT3 on several benchmark tasks, including text classification and language generation, after finetuning with datasets from domains such as news, social media, and scientific papers. Our results show that the performance of GPT3 greatly improves with finetuning, especially when the training data is closely related to the task at hand. We also examine the trade-offs between the amount of training data and the domain relevance of the dataset.\n\n## Sections ##\n1. Introduction\n2. Related work\n3. Experimental setup\n4. Results and analysis\n5. Discussion and conclusions\n6. Future work\n7. Acknowledgements\n8. References\n\n## Subsection outline ##\n1. Introduction\n   1. Motivation\n   2. Objectives\n2. Related work\n   1. Pretraining and finetuning\n   2. GPT3 and its architecture\n3. Experimental setup\n   1. Dataset selection\n   2. Preprocessing\n   3. Finetuning process\n4. Results and analysis\n   1. Text classification\n   2. Language generation\n   3. Effect of dataset size\n   4. Effect of dataset domain\n5. Discussion and conclusions\n   1. Summary of results\n   2. Implications and future research\n6. Future work\n   1. Possible extensions\n   2. Limitations\n7. Acknowledgements\n8. References\n\n## Content ##\n\n\\documentclass{article}\n\n\\usepackage{graphicx}\n\\usepackage{booktabs}\n\\usepackage{hyperref}\n\n\\begin{document}\n\n\\title{An exploration of GPT3 finetuning with datasets in different domains and sizes}\n\\author{John Doe \\and Jane Smith \\and James Johnson}\n\\date{June 2021}\n\n\\maketitle\n\n\\section{Introduction}\n\nThe recent release of GPT3 has generated a lot of excitement in the natural language processing (NLP) community. With its impressive performance on a wide range of language tasks, GPT3 has the potential to
```

Example: format trick & partial code

```
prompt = '''
// Give the URL and text of the Wikipedia article for the given page name

// Output fields:
// page_name: the name of the page to get the text for
// page_url: the URL of the page
// page_text: the text of the page

input = {
  "page_name": "Michelle Yeoh"
};

output = {
'''
openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  temperature=0.7,
  messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": prompt},
    ]
)
```

Example: format trick


```
<OpenAIObject chat.completion id=chatcmpl-6tjb7bMPEP0Ci0yloH1JovB55hfjx at 0x7fad4567cd10> JSON: {
  "choices": [
    {
      "finish_reason": "stop",
      "index": 0,
      "message": {
        "content": "\"page_name\": \"Michelle Yeoh\",\n\"page_url\": \"https://en.wikipedia.org/wiki/Michelle_Yeoh\",\n\"page_text\":
\"Michelle Yeoh Choo-Kheng (born 6 August 1962) is a Malaysian actress who rose to fame in 1990s Hong Kong action films and is best known
internationally for her roles in the James Bond film Tomorrow Never Dies (1997) and the Chinese-language martial arts film Crouching Tiger,
Hidden Dragon (2000), for which she was nominated for the BAFTA Award for Best Actress in a Leading Role. She is also known for playing Aung
San Suu Kyi in the 2011 Luc Besson film The Lady. In 2018, she starred in the American TV series Star Trek: Discovery.\"",
        "role": "assistant"
      }
```

Example: partial code

```
prompt = '''
assert len(EXAMPLE_DATASET)==5
assert len(review)<=20
assert sentiment in {'Positive', 'Negative', 'Neutral'}
## DATA GENERATED FOR TASK ## <Restaurant review sentiment analysis>
## EXAMPLE DATASET ## <review, sentiment>

Generate a dataset based on the above instructions.
'''
openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  temperature=0.7,
  messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": prompt},
    ]
)
```

Example: external interpreter

```
prompt = '''
## Instructions ##
If you can answer the question directly, use this format:
Question: ${Question}
Answer: ${Answer}

You are unable to answer any question that requires:
1. Counting of more than several items (i.e. words in a sentence or items in a list)
2. Analyzing text as a sequence of characters (i.e. reversing a string, how many times a character appears in a string)

In these cases, consult IPython. Use this format:
Question: ${Question}
IPython session:
<
${IPython commands and outputs that needed to get the final answer}
>

Answer: ${Answer}

## Question ##
Now answer the question: what is the product of all prime numbers under 30

'''
openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  temperature=0.7,
```

Example: format trick


```
<OpenAIObject chat.completion id=chatcmpl-6tlXfpW03W32ePQjrpfQtU3r1eiAy at 0x7f202e297630> JSON: {
  "choices": [
    {
      "finish_reason": "stop",
      "index": 0,
      "message": {
        "content": "Answer: The largest prime number under 200 is 197.",
        "role": "assistant"
      }
```

```
<OpenAIObject chat.completion id=chatcmpl-6tlaQY5hZgqCvcfpMGmRLOCu62UPm at 0x7f20769fa040> JSON: {
  "choices": [
    {
      "finish_reason": "stop",
      "index": 0,
      "message": {
        "content": "Question: What is the inflection point of function f(x) = 3*x^8 - 29*x^7 + 1729?\n\nIPython session:\n<\nfrom sympy import
*\nx = symbols('x')\nf = 3*x**8 - 29*x**7 + 1729\nf_second_derivative = diff(f, x, 2)\ninflection_points = solve(f_second_derivative,
x)\ninflection_points\n>\n\nAnswer: The inflection points of the function f(x) = 3*x^8 - 29*x^7 + 1729 are returned as an empty list, which
means there are no inflection points.",
        "role": "assistant"
```

Example: CoT

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Example:
Zero shot CoT

```
prompt = '''
Question: There are five spiders and six pigs, how many legs they have in total?
Answer: Let's think step by step.
'''
openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  temperature=0.7,
  messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": prompt},
    ]
)
```

```
<OpenAIObject chat.completion id=chatcmpl-6tr58rU4mmMfbrGSzq84tDQ5uYMHP at 0x7f56946ab680> JSON: {
  "choices": [
    {
      "finish_reason": "stop",
      "index": 0,
      "message": {
        "content": "Each spider has eight legs, so five spiders have a total of 5 x 8 = 40 legs. Each pig has four legs, so six pigs have a
total of 6 x 4 = 24 legs. \n\nTherefore, the total number of legs is 40 + 24 = 64 legs.",
        "role": "assistant"
```

Example: synthetic prompting

Seed Examples

Question: Carol has 20 signatures in her book, and Jennifer has 44. The sisters have three more weeks of summer vacation, and they decide they want to reach 100 signatures between them by the end of the summer. How many signatures do the sisters need to collect to reach their goal?

Question: A team of 4 painters worked on a mansion for 3/8ths of a day every day for 3 weeks. How many hours of work did each painter put in?

Synthetic Demonstrations from SYNTHETIC PROMPTING

Given Topic Word: idea
Question: If 5 people each have 10 ideas, with 5 of those ideas being innovative and taking 2 minutes each to develop, and the other 5 ideas being non-innovative and taking 1 minute each to develop, how many minutes will it take all 5 people to develop all 10 ideas?

Given Topic Word: office
Question: The office is 20 feet wide, 30 feet long and 10 feet high. It has two windows that are each 5 feet wide and 6 feet high, and one door that is 3 feet wide and 8 feet high. What is the total area of the office walls?

Given Topic Word: gallery
Question: A gallery has 10 paintings, 9 sculptures, 6 photos, and 4 mixed media pieces. The painting is $200, the sculpture is $500, the photo is $100 and the mixed media piece is $150. You get a 15% discount and you have to pay 5% tax. How much will you pay in total?

Summary

- Prompt engineering is a crucial aspect of maximizing the performance of LLMs
- Most advanced version, GPT4, also benefits from few shots CoT demonstrations (evaluation results in GPT4 technical report use 5-20 shots prompting)
- We can leverage LLMs to automate the prompt generation step
- Domain expertise is needed to create prompts for specific applications

Potential steps & discussions

- Continue exploring new application area of LLM
  - Self instruct: the way Alpaca improved from LLaMA
- Efficient fine-tuning for different LLM
- Efficient Evaluation methods of LLM
- Prompt engineering best practice with retrieval results
- Ground LLM on graph, the structured source of information