

```
from datasets import load_dataset
import random

dataset = load_dataset("gsm8k", "main", split="test")

problem = random.choice(dataset)["question"]
print("Selected Question:", problem)
```

→ of the employees' contracts expired. Calculate the total amount of money the company paid to the employees in the two months.

```
import yaml

yaml_content = """
prompts:
  - name: generic_solution
    template: |
      Solve the following math problem carefully and answer with a number:
      {problem}

  - name: one_shot_prompt
    template: |
      Solve this math problem with a numerical answer:
      Example: {example_1}
      Problem: {problem}
      Answer as a number:

  - name: two_shot_prompt
    template: |
      Solve the following problem and answer as a number:
      Example 1: {example_1}
      Example 2: {example_2}
      Problem: {problem}
      Answer as a number:

  - name: two_shot_cot_prompt
    template: |
      Use a step-by-step approach to solve this problem and provide a final answer as a number:
      Example 1: {cot_example_1}
      Example 2: {cot_example_2}
      Problem: {problem}
      Answer:
      """

with open("prompts.yaml", "w") as file:
    file.write(yaml_content)
```

```
# Load YAML file
with open("prompts.yaml", "r") as file:
    prompt_templates = yaml.safe_load(file)["prompts"]

# Function to retrieve a prompt template by name and populate placeholders
def get_prompt(name, **kwargs):
    for prompt in prompt_templates:
        if prompt["name"] == name:
            return prompt["template"].format(**kwargs)
    raise ValueError(f"No prompt found with the name '{name}'")
```

```
# Define example solutions for one-shot and two-shot prompts
example_1 = "A box has 5 apples, another box has 15 apples. Difference is 10, twice the oranges count. So, 5 oranges."
example_2 = "One pond has 6 frogs, another 18. Difference is 12, twice the turtles count. So, 6 turtles."
cot_example_1 = "To solve 10 + 10, first add the numbers to get 20."
cot_example_2 = "To solve 20 - 5, first subtract to get 15."
```

```
# Generate prompts
generic_prompt = get_prompt("generic_solution", problem=problem)
one_shot_prompt = get_prompt("one_shot_prompt", example_1=example_1, problem=problem)
two_shot_prompt = get_prompt("two_shot_prompt", example_1=example_1, example_2=example_2, problem=problem)
two_shot_cot_prompt = get_prompt("two_shot_cot_prompt", cot_example_1=cot_example_1, cot_example_2=cot_example_2, problem=problem)

# Display generated prompts
print("Generic Solution Prompt:\n", generic_prompt)
```

```
print("\nOne Shot Prompt:\n", one_shot_prompt)
print("\nTwo Shot Prompt:\n", two_shot_prompt)
print("\nTwo Shot CoT Prompt:\n", two_shot_cot_prompt)
```

Generic Solution Prompt:
Solve the following math problem carefully and answer with a number:
Carolyn works for a delivery service company that hires on a contract basis. In May, they hired 40 employees, paying each em

One Shot Prompt:
Solve this math problem with a numerical answer:
Example: A box has 5 apples, another box has 15 apples. Difference is 10, twice the oranges count. So, 5 oranges.
Problem: Carolyn works for a delivery service company that hires on a contract basis. In May, they hired 40 employees, payin
Answer as a number:

Two Shot Prompt:
Solve the following problem and answer as a number:
Example 1: A box has 5 apples, another box has 15 apples. Difference is 10, twice the oranges count. So, 5 oranges.
Example 2: One pond has 6 frogs, another 18. Difference is 12, twice the turtles count. So, 6 turtles.
Problem: Carolyn works for a delivery service company that hires on a contract basis. In May, they hired 40 employees, payin
Answer as a number:

Two Shot CoT Prompt:
Use a step-by-step approach to solve this problem and provide a final answer as a number:
Example 1: To solve $10 + 10$, first add the numbers to get 20.
Example 2: To solve $20 - 5$, first subtract to get 15.
Problem: Carolyn works for a delivery service company that hires on a contract basis. In May, they hired 40 employees, payin
Answer:

```
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM, pipeline
import torch

# Load Flan-T5 model
model_name = "google/flan-t5-large"

tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)
device = 0 if torch.cuda.is_available() else -1
# Set up the pipeline with Flan-T5 for text generation
nlp = pipeline("text2text-generation", model=model, tokenizer=tokenizer, device=device)

def generate_solution(prompt):
    # Adjusting max_new_tokens for Flan-T5, which is better at handling structured prompts
    return nlp(prompt, max_new_tokens=100, num_return_sequences=1)[0]["generated_text"]

# Get responses
print("\nResponse for Generic Prompt:\n", generate_solution(generic_prompt))
print("\nResponse for One Shot Prompt:\n", generate_solution(one_shot_prompt))
print("\nResponse for Two Shot Prompt:\n", generate_solution(two_shot_prompt))
print("\nResponse for Two Shot CoT Prompt:\n", generate_solution(two_shot_cot_prompt))
```

Response for Generic Prompt:
 $40 * 15 = \$1200$.

Response for One Shot Prompt:
 $40 * 15 = \$1200$

Response for Two Shot Prompt:
In May, the company paid $40 * \$15 = \1200 to the 40 employees. In June, they paid $1 / 4 * 40 = \$20$ to the 40 employees. In

Response for Two Shot CoT Prompt:
In May, the company paid $40 * \$15 = \1200 per employee. In June, they paid $1 / 4 * 40 = \$20$ per employee. In total, the com

```
from transformers import AutoTokenizer, AutoModelForCausalLM, pipeline

model_name = "EleutherAI/gpt-neo-1.3B"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name)

nlp = pipeline("text-generation", model=model, tokenizer=tokenizer, device=device)

def generate_solution(prompt):
    return nlp(prompt, max_new_tokens=100, num_return_sequences=1)[0]["generated_text"]
```

```
# Get responses
print("\nResponse for Generic Prompt:\n", generate_solution(generic_prompt))
print("\nResponse for One Shot Prompt:\n", generate_solution(one_shot_prompt))
print("\nResponse for Two Shot Prompt:\n", generate_solution(two_shot_prompt))
print("\nResponse for Two Shot CoT Prompt:\n", generate_solution(two_shot_cot_prompt))
```

Setting `pad_token_id` to `eos_token_id`:None for open-end generation.
 Setting `pad_token_id` to `eos_token_id`:None for open-end generation.

Response for Generic Prompt:

Solve the following math problem carefully and answer with a number:

Carolyn works for a delivery service company that hires on a contract basis. In May, they hired 40 employees, paying each em
 I used the formula:

$$M = \frac{N+G}{2} \times 40 \sqrt{3} + (C-N)S \times 40 \left(\frac{3+1.1^{3/4}}{2+1.1^{3/4}} \right) S^{3/4} + S \times 40 (C-N)^2$$

 In the two months

Setting `pad_token_id` to `eos_token_id`:None for open-end generation.

Response for One Shot Prompt:

Solve this math problem with a numerical answer:

Example: A box has 5 apples, another box has 15 apples. Difference is 10, twice the oranges count. So, 5 oranges.

Problem: Carolyn works for a delivery service company that hires on a contract basis. In May, they hired 40 employees, payin
 Answer as a number:

Let the first employee work in May, so he's paid \$15 each hour. Let \$5 of that be for the 20 hours worked by the first emplo
 Calculate the total amount of money the company paid to the 40 employees. That's \$

Setting `pad_token_id` to `eos_token_id`:None for open-end generation.

Response for Two Shot Prompt:

Solve the following problem and answer as a number:

Example 1: A box has 5 apples, another box has 15 apples. Difference is 10, twice the oranges count. So, 5 oranges.

Example 2: One pond has 6 frogs, another 18. Difference is 12, twice the turtles count. So, 6 turtles.

Problem: Carolyn works for a delivery service company that hires on a contract basis. In May, they hired 40 employees, payin
 Answer as a number:

Let's use the formula given for a one-month contract (6 weeks x 12 hours per week).

So, the number of employees should now be

The amount of money the company paid to the 40 employees is half the total.

In the two months they worked, the company paid \$5,200 (40 hours x 40 employees) to the employees.

Solution: Let's work with a \$5,200 annual salary on the employees, and let's divide the total by the

Response for Two Shot CoT Prompt:

Use a step-by-step approach to solve this problem and provide a final answer as a number:

Example 1: To solve $10 + 10$, first add the numbers to get 20.

Example 2: To solve $20 - 5$, first subtract to get 15.

Problem: Carolyn works for a delivery service company that hires on a contract basis. In May, they hired 40 employees, payin
 Answer:

- a) \$15
- b) \$40
- c) \$75
- d) \$110
- e) \$165

Solution:

- a) subtract the \$15 and answer "a)"
- b) subtract the \$40 and answer "b)"
- c) subtract the \$75 and answer "c)"
- d) add the \$110 and answer "d)"
- e) add the \$165 and answer "e)"

A:

Solved

The answer

I used 2 LLM's to solve a GSM8K math problem using various prompt engineering strategies, and followed good prompt design by using Yaml and injecting responses. I compared EleutherAI's GPT-Neo with Google's Flan-T5 models. My approaches included generic, one-shot, two-shot, and two-shot Chain-of-Thought (CoT) prompts. Flan-T5 outperformed GPT-Neo, particularly with structured prompts, where it generated logical and numerically accurate responses. Two-shot CoT prompting proved most effective, due to step-by-step reasoning, crucial for solving problems with constraints. GPT-Neo struggled with numerical coherence and often returned verbose, less structured answers. Observations indicate that well-designed CoT prompts combined with high-capacity models like Flan-T5 can improve performance, which supports CoT with LLM's. Hallucination is a real problem as seen with both model's calculations, which shows having larger more complex models is necessary for inference tasks on such problems.

