LLM Homework 1

Vincent Collin Yennoto (57887312)

```
!pip install -q google-generativeai

import google.generativeai as genai
from google.colab import userdata
import textwrap
import json
import time
import base64
import time
import json
from PIL import Image
import io
import PIL

# GEMINI_API_KEY2 = userdata.get('GEMINI_API_KEY')

genai.configure(api_key=GEMINI_API_KEY)
model = genai.GenerativeModel('gemini-2.5-flash-lite')
```

```
/usr/local/lib/python3.12/dist-packages/google/colab/_import_hooks/_

All support for the `google.generativeai` package has ended. It will
updates or bug fixes. Please switch to the `google.genai` package as
See README for more details:

https://github.com/google-gemini/deprecated-generative-ai-python/blo

  loader.exec_module(module)
```

Test Cases

Tasks

1. Math and Logic

- This is a cryptic crossword. Sounds like get away footwear (4)
- f(x) = 10. What necessarily must f(2) be?
- Prove that (ab)^n = a^n b^n
- Prove that the symmetries of a cube is isomorphic to S4
- What is the integral of 4x^3 + 10x - 2 evaluated from 3 to 15

---

2. Image Recognition

Gemini will be asked what the input image is.

- Image of a dog face, taken from AFHQ dataset
- Scene from a movie, taken from One Battle After Another movie still
- Image of a branded food, taken from Big Mac wikipedia page
- AI-generated image of gibberish, taken from https://creator.nightcafe.studio/creation/3JqOyzbbZgIYiDSQoJGK
- Image of ML research paper pipeline, taken from DXAI research paper https://arxiv.org/pdf/2401.00320

Prompt Tricks

The prompt tricks will be divided into three different categories

1. Specificity: Please be as specific as you can / Do not skip over any steps
2. Additional Info: Domain specific advice / Extra information

```
mathNlogic = ["This is a cryptic crossword. Sounds like get away fo
       "f(x) = 10. What necessarily must f(2) be?",
       "Prove that (ab)^n = a^n b^n",
       "Prove that the symmetries of a cube is isomorphic to S4",
       "What is the integral of 4x^3 + 10x - 2 evaluated from 3 to
       ]

math_specific = ["Please be as specific as you can when answering t
                 "Please be as specific as you can when answering t
                 "Please be as specific as you can when answering t
                 "Please be as specific as you can when answering t
                 "Please be as specific as you can when answering t
]

math_addition = ["A cryptic crossword is a crossword clue that cons
                 "This question is from a British GCSE. Don't use a
                 "This question is from my Abstract Algebra course.
                 "This question is from my Abstract Algebra course.
                 "You are a calculus instructor teaching a first-ye
]
#--------------
imagerecog = ["dog.jpg",
       "movie.png",
       "food.jpeg",
       "slop.png",
       "pipeline.png"
       ]

image_specific = [
    "What is in the image? Please be as specific as you can.",
    "What is happening in the image? Please be as detailed as you c
    "What is in the image? Please be as specific as you can.",
    "What is happening/in the image? Please be as detailed as you c
    "What is happening in the image? Please be as detailed as you c
]

image_addition = [
    "What is in the image? What is the breed, color, and other feat
    "This is a still from a movie. What is happening in this image
    "I know that this is a fast-food item. What is the food called?
    "I know that this is an AI generated image. I can't make sense
    "This image is from the research paper Decomposition-based Expl
]
```

```
results = []

print("🔢 Running math experiments...")
print("=" * 50)

for i in range(len(mathNlogic)):
    print(f"\nTest {i+1}: {mathNlogic[i][:40]}...")

    # Create 3 prompts for each question
    baseline = mathNlogic[i]
    specificity = f"{math_specific[i]}\n\n{mathNlogic[i]}"
    addition = f"{math_addition[i]}\n\n{mathNlogic[i]}"

    # Get responses
    print("  Testing baseline...")
    baseline_response = model.generate_content(baseline).text
    time.sleep(1)

    print("  Testing specificity...")
    specificity_response = model.generate_content(specificity).text
    time.sleep(1)

    print("  Testing addition...")
    addition_response = model.generate_content(addition).text
    time.sleep(2)  # Longer wait between questions

    # Store
    results.append({
        "test_id": i + 1,
        "question": mathNlogic[i],
        "baseline_prompt": baseline,
        "specificity_prompt": specificity,
        "addition_prompt": addition,
        "baseline_response": baseline_response,
        "specificity_response": specificity_response,
        "addition_response": addition_response
    })

    # Quick preview
    print(f"  Baseline: {baseline_response[:100]}...")
    print(f"  Specificity: {specificity_response[:100]}...")
    print(f"  Addition: {addition_response[:100]}...")

# Save to JSON
with open('math_experiment_results.json', 'w') as f:
    json.dump(results, f, indent=2)

print(f"\n✅ Done! Saved {len(results)} results to 'math_experiment

# Quick summary
print("\n📊 Summary:")
```

```
    for r in results:
        print(f"\nTest {r['test_id']}:")
        print(f"  Baseline: {len(r['baseline_response'])} chars")
        print(f"  Specificity: {len(r['specificity_response'])} chars")
        print(f"  Addition: {len(r['addition_response'])} chars")
```

```
🔢 Running math experiments...
==================================================

Test 1: This is a cryptic crossword. Sounds like...
  Testing baseline...
  Testing specificity...
  Testing addition...
  Baseline: This sounds like a fun one! Let's break it down:

*   **"Sounds like"**: This is the **homophone ind...
  Specificity: This is a fun cryptic crossword clue! Let's break it

**T...
  Addition: Let's break down this cryptic crossword clue: "Sounds li

*   **Definition:...

Test 2: f(x) = 10. What necessarily must f(2) be...
  Testing baseline...
  Testing specificity...
  Testing addition...
ERROR:tornado.access:503 POST /v1beta/models/gemini-2.5-flash-lite:g
  Baseline: The problem states that $f(x) = 10$.
This is a definition of a function where the output of the func...
  Specificity: Here's a breakdown of why f(2) must be 10, with a log

**The Problem:...
  Addition: Here's how to solve this, keeping it simple for a GCSE l

The question tells you that **f(x) = ...

Test 3: Prove that (ab)^n = a^n b^n...
  Testing baseline...
  Testing specificity...
  Testing addition...
  Baseline: We want to prove that for any real numbers $a$ and $b$,
  Specificity: Let's prove the property of exponents $(ab)^n = a^n b
  Addition: While the statement $(ab)^n = a^n b^n$ is **not universa

Test 4: Prove that the symmetries of a cube is i...
  Testing baseline...
  Testing specificity...
  Testing addition...
ERROR:tornado.access:503 POST /v1beta/models/gemini-2.5-flash-lite:g
  Baseline: To prove that the group of symmetries of a cube is isomo
  Specificity: Let's embark on a rigorous proof demonstrating that t
  Addition: Absolutely! Let's break down the proof that the symmetry
```

```
Test 5: What is the integral of 4x^3 + 10x - 2 e...
  Testing baseline...
ERROR:tornado.access:503 POST /v1beta/models/gemini-2.5-flash-lite:g
  Testing specificity...
  Testing addition...
  Baseline: To evaluate the definite integral of $4x^3 + 10x - 2$ fr
  Specificity: We need to evaluate the definite integral of the func
  Addition: Alright everyone, settle in! Today, we're going to tackl

✅ Done! Saved 5 results to 'math_experiment_results.json'

📊 Summary:

Test 1:
  Baseline: 1291 chars
  Specificity: 23523 chars
  Addition: 816 chars

Test 2:
  Baseline: 433 chars
  Specificity: 2041 chars
  Addition: 348 chars

Test 3:
  Baseline: 2207 chars
  Specificity: 3995 chars
  Addition: 4548 chars

Test 4:
  Baseline: 7983 chars
  Specificity: 27405 chars
  Addition: 19199 chars

Test 5:
  Baseline: 1661 chars
  Specificity: 3143 chars
  Addition: 3616 chars
```

```python
genai.configure(api_key='API_KEY2')
model2 = genai.GenerativeModel('gemini-2.5-flash-lite')
```

```python
print("🖼️ Running image recognition experiments...")
print("=" * 50)

image_results = []

for i, img_file in enumerate(imagerecog):
    # Construct full path
```

```
        img_path = f"/content/{img_file}"

        print(f"\nImage {i+1}: {img_file}")

        # Open image
        img = PIL.Image.open(img_path)

        # Create prompts
        baseline_prompt = "What is in the image?"
        specificity_prompt = image_specific[i]
        addition_prompt = image_addition[i]

        # Get responses
        print("  Testing baseline...")
        baseline_response = model2.generate_content([baseline_prompt, i
        time.sleep(1)

        print("  Testing specificity...")
        specificity_response = model2.generate_content([specificity_prc
        time.sleep(1)

        print("  Testing addition...")
        addition_response = model2.generate_content([addition_prompt, i
        time.sleep(2)

        # Store results
        image_results.append({
            "image_id": i + 1,
            "image_file": img_file,
            "baseline_prompt": baseline_prompt,
            "specificity_prompt": specificity_prompt,
            "addition_prompt": addition_prompt,
            "baseline_response": baseline_response,
            "specificity_response": specificity_response,
            "addition_response": addition_response
        })

        # Quick preview
        print(f"  Baseline: {baseline_response[:100]}...")
        print(f"  Specificity: {specificity_response[:100]}...")
        print(f"  Addition: {addition_response[:100]}...")

    # Save to JSON
    with open('image_experiment_results.json', 'w') as f:
        json.dump(image_results, f, indent=2)

    print(f"\n✅ Done! Saved {len(image_results)} results to 'image_exp
    # Quick summary
    print("\n📊 Image Recognition Summary:")
    for r in image_results:
        print(f"\nImage {r['image_id']} ({r['image_file']}):")
        print(f"  Baseline: {len(r['baseline_response'])} chars")
        print(f"  Specificity: {len(r['specificity_response'])} chars")
        print(f"  Addition: {len(r['addition_response'])} chars")
```

```
🖼️ Running image recognition experiments...
==================================================

Image 1: dog.jpg
  Testing baseline...
  Testing specificity...
  Testing addition...
  Baseline: The image shows a close-up of a dog's face. The dog has
  Specificity: The image is a close-up portrait of a small, fluffy d
  Addition: The image shows a close-up of a dog's face. The dog appe

Image 2: movie.png
  Testing baseline...
  Testing specificity...
  Testing addition...
  Baseline: In the image, a person is using a payphone. The payphone
  Specificity: The image depicts a woman with short, dark hair, look
  Addition: In this movie still, a woman is using a public payphone

Image 3: food.jpeg
  Testing baseline...
  Testing specificity...
  Testing addition...
  Baseline: The image shows a **Big Mac**.

It's a hamburger from McDonald's, characterized by its three-part s.
  Specificity: The image shows a **Big Mac hamburger** placed on a p
  Addition: The food item in the image is a **Big Mac**.

It is from **McDonald's**, a multinational fast-food c...

Image 4: slop.png
  Testing baseline...
  Testing specificity...
  Testing addition...
  Baseline: The image appears to be an abstract or surreal represent
  Specificity: This is a surreal and abstract image with a strong fo
  Addition: This is an AI-generated image, and its abstract nature c

Image 5: pipeline.png
  Testing baseline...
  Testing specificity...
  Testing addition...
  Baseline: The image displays a flowchart illustrating a style tran
  Specificity: The image depicts a neural network architecture desig
  Addition: This image illustrates the pipeline of a Decomposition-b

✅ Done! Saved 5 results to 'image_experiment_results.json'

📊 Image Recognition Summary:

Image 1 (dog.jpg):
  Baseline: 389 chars
  Specificity: 323 chars
  Addition: 980 chars

Image 2 (movie.png):
  Baseline: 356 chars
  Specificity: 1216 chars
  Addition: 401 chars

Image 3 (food.jpeg):
  Baseline: 186 chars
  Specificity: 1702 chars
  Addition: 162 chars

Image 4 (slop.png):
  Baseline: 487 chars
  Specificity: 3844 chars
  Addition: 1669 chars

Image 5 (pipeline.png):
  Baseline: 3270 chars
  Specificity: 4080 chars
  Addition: 4496 chars
```

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.