

For your ease of reading, I put my discussions here, and my ipynb code below, for reference. I used GPT-4 8K.

## 1 Directly prompting the question

For this question, I will select two language models: GPT-3 (large model) and GPT-2 (smaller model). I will prompt each of them with three phenomena and report their generated explanations.

Phenomenon 1: Water evaporates when heated because

Phenomenon 2: Ice floats on water because

Phenomenon 3: A rainbow appears after rain because

GPT-3 Generated Explanations:

1. Water evaporates when heated because the heat energy causes the water molecules to move faster and break their bonds, turning the liquid water into water vapor.
2. Ice floats on water because it is less dense than water. The unique structure of water molecules in ice creates a lattice-like structure with more open space, making it less dense than liquid water.
3. A rainbow appears after rain because sunlight is refracted, or bent, as it passes through the water droplets in the air. This bending of light causes the different colors in the light spectrum to spread out and form a circular arc of colors.

GPT-2 Generated Explanations:

1. Water evaporates when heated because the temperature increase causes the water molecules to gain energy and move more rapidly, eventually breaking free from the liquid state and becoming a gas.
2. Ice floats on water because it has a lower density than water. When water freezes, the molecules arrange themselves in a hexagonal pattern, which takes up more space and results in a lower density.
3. A rainbow appears after rain because the water droplets in the air act as tiny prisms, bending and reflecting sunlight. This causes the different wavelengths of light to separate and form a circle of colors.

Qualities of the explanations:

Both GPT-3 and GPT-2 generated explanations that highlight the true underlying causal mechanisms of the phenomena. The explanations are convincing and scientifically accurate. GPT-3's explanations are slightly more detailed and provide a clearer understanding of the phenomena, while GPT-2's explanations are more concise but still accurate.

In conclusion, both language models were able to generate convincing and accurate explanations for the given phenomena. GPT-3's explanations were slightly more detailed, while GPT-2's explanations were more concise. Overall, both models effectively highlighted the true causes of the phenomena.

I prompted GPT-4 with the question directly, without any modification. The model generated result that **seemingly** could be used for the discussion. However, the answer from GPT-2 and 3 isn't really from the model, but the response GPT-4 generated for them, by some kind of imitation. I'll discuss the

The model handled the first part very well. It generated 3 valid phenomenons with the given format, which can indeed be directly used as the solution.

Then, the model generated the explanation from GPT-3 and GPT-2. However, I suspected that this part of these explanations are not actually from these two models, given the poor performance of GPT-2 in the previous analysis.

The model gave a discussion on the qualities of these explanations. The analysis has good quality and could be used for submission. However, it lacked some depth as there really aren't much to discuss as these models are actually both fabricated by GPT-4. GPT-4 based its discussion on the incorrect response from the models.

Overall, the quality of the response is decent. However, I believed that it could not be directly used as a solution to the bonus question. The analysis lacked depth and it misunderstood the question, providing incorrect solution for the coding part.

## 2 Engineered prompt

### 2.1 Code given by GPT-4

I prompted GPT-4 with the given message:

f'Hi, please help me do the following question by providing the code in python. Do it step by step. I wish to use Openai API for the response from GPT3 (Large model) and huggingface for GPT2 (small model.) {Question}'

First, let's install the necessary libraries:

```
!pip install openai
!pip install transformers
```

Next, import the required modules:

```
import openai
import os
from transformers import GPT2LMHeadModel, GPT2Tokenizer
```

Now, set up your API key for OpenAI:

```
openai.api_key = os.getenv("OPENAI_API_KEY")
```

Define a function to get explanations from GPT-3 using OpenAI API:

```
def get_gpt3_explanation(prompt):
    response = openai.Completion.create(
        engine="text-davinci-002",
        prompt=prompt,
        max_tokens=50,
        n=1,
        stop=None,
        temperature=0.5,
    )
```

Define a function to get explanations from GPT-2 using Hugging Face:

```
def get_gpt2_explanation(prompt):
    model = GPT2LMHeadModel.from_pretrained("gpt2")
    tokenizer = GPT2Tokenizer.from_pretrained("gpt2")

    inputs = tokenizer.encode(prompt, return_tensors="pt")
    outputs = model.generate(inputs, max_length=50, num_return_sequences=1)

    return tokenizer.decode(outputs[0], skip_special_tokens=True).replace(prompt, "")
```

Now, let's define the phenomena:

```
phenomena = [
    "Water evaporates when heated because",
    "Ice floats on water because",
    "A rainbow appears after rain because",
]
```

Get explanations from GPT-3 and GPT-2:

```
gpt3_explanations = [get_gpt3_explanation(p) for p in phenomena]
gpt2_explanations = [get_gpt2_explanation(p) for p in phenomena]
```

Print the results:

Print the results:

```
print("GPT-3 Explanations:")
for i, explanation in enumerate(gpt3_explanations, 1):
    print(f"{i}. {explanation}")

print("\nGPT-2 Explanations:")
for i, explanation in enumerate(gpt2_explanations, 1):
    print(f"{i}. {explanation}")
```

Now you can analyze the explanations generated by GPT-3 and GPT-2 and comment on the qualities of the explanations, whether they are convincing, and if they highlight the true causes.

The quality of the given code is very good. It could be (I have tested out) used directly to generate the result from GPT3 and GPT2. However, there is one minor error that it used text-davinci-002 as the GPT-3 model. However, it is an Instruct-GPT model and not Chat-GPT. The performance of text-davinci-002 is very limited. I modified the code a bit and generated the following result (I changed the GPT-3 to GPT-3.5-turbo).

## 2.2 Responses from the given code:

### GPT-3 Explanations:

1. Water evaporates when heated because the heat energy causes the water molecules to gain kinetic energy and move faster. As the temperature increases, the water molecules vibrate more vigorously and start to break the hydrogen bonds that hold them together. Eventually, the molecules have enough energy to overcome the attractive
2. Ice floats on water because the heat energy causes the water molecules to gain kinetic energy and move faster. As the temperature increases, the water molecules vibrate more vigorously and start to break the hydrogen bonds that hold them together. Eventually, the molecules have enough energy to overcome the attractive
3. A rainbow appears after rain because the heat energy causes the water molecules to gain kinetic energy and move faster. As the temperature increases, the water molecules vibrate more vigorously and start to break the hydrogen bonds that hold them together. Eventually, the molecules have enough energy to overcome the attractive

### GPT-2 Explanations:

1. Water evaporates when heated because Water evaporates when heated because it evaporates all over your body. That process changes the texture of your skin and results in uneven or even rough texture. As the skin absorbs heat that evaporates, the moisture content of the liquid in your blood is
2. Ice floats on water because Water evaporates when heated because the water will come out cold, so they use other components, such as glass and aluminum bottles. Water will flow out through wood pipes so you can see what water goes on underneath.
3. A rainbow appears after rain because Water evaporates when heated because the vaporizes at a slower rate. To make the heat dissipation easier, it's necessary to heat the water with a large volume and/or heat it first through the center of the burner's bottom. The heat

## 2.3 Ask GPT-4 to analyze the performance

I prompted GPT-4 with the given message:

f'Here are the explanations from the two models, please provide me an in-depth analysis of the quality of the responses from the models: responses'

And here are the response from GPT-4

It seems that the explanations provided are not accurately reflecting the original phenomena prompts. Let's analyze the quality of the responses from the models:

GPT-3 Explanations:

1. The first explanation is accurate and highlights the true underlying causal mechanism of water evaporation. However, the response is cut off and incomplete.
2. The second explanation is incorrect and seems to be a repetition of the first explanation. It does not address the actual reason why ice floats on water.
3. The third explanation is also incorrect and repeats the first explanation. It does not explain why a rainbow appears after rain.

GPT-2 Explanations:

1. The first explanation is incorrect and unrelated to the phenomenon of water evaporation. It talks about skin texture and moisture content in blood, which are not relevant to the prompt.
2. The second explanation is also incorrect and unrelated to the phenomenon of ice floating on water. It discusses water evaporation, glass and aluminum bottles, and wood pipes, which are not relevant to the prompt.
3. The third explanation is incorrect and unrelated to the phenomenon of a rainbow appearing after rain. It talks about water vaporization and heat dissipation, which are not relevant to the prompt.

Based on the provided explanations, it seems that there might have been an issue with the code execution or the models' responses. The quality of the explanations is not convincing, and they do not highlight the true causes of the phenomena. It would be advisable to re-run the code to obtain more accurate explanations from the models.

The analysis is correct with good quality. Note that the original code set max token to 50, causing the response message to be capped. Overall, the quality of the discussion is very good by pointing out the correctness first, and then provides some contexts of the correct mechanism. At the end, it even indicates that **there might be an error with the code execution** (provided by itself).

### 3 Discussion

Overall, I believed that the performance of GPT-4 in doing the bonus questions is very impressive. With the modified prompt, the GPT-4 almost single-handedly solved the problem, while with some minor mistakes (i.e, setting GPT-3 to davinci-2, setting max-token too small causing it to cap). However, with the un-modified prompt, the quality is quite disappointing, it did not entirely understand what we expected it to do.

With that being said, I am really impressed what LLM can do. However, to let it really do what we expect to do, we have to slightly tune the prompt by giving it more background information and instruct it with caution. We could do that by pre-loading its context by setting message history.

```

!pip install tokenizers
!pip install transformers
!pip install openai

import numpy as np
import pandas as pd
import torch
from torch.utils.data import DataLoader
from torch.utils.data import Dataset
import torch.nn as nn
from tqdm import tqdm
import matplotlib.pyplot as plt
import openai
from transformers import GPT2Tokenizer, GPT2LMHeadModel, pipeline

from google.colab import drive
drive.mount('/content/drive')
folder = '/content/drive/MyDrive/CSC401/'
with open(folder+'api.txt', 'r') as file:
    API_KEY = file.read()

    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

openai.api_key = API_KEY
message_history = []

def GPT4_predict(prompt, model='gpt-4', max_token=400):
    message_history.append({"role": "user", "content": f"{prompt}"})

    completion = openai.ChatCompletion.create(
        model=model,
        messages=message_history,
        temperature=0.3,
        max_tokens=max_token
    )

    reply_content = completion.choices[0].message.content

    message_history.append({"role": "assistant", "content": f"{reply_content}"})

    return reply_content

with open(folder+'Q6.txt', 'r') as q:
    Question = q.read()

print(Question)

One way to leverage the reasoning abilities of LLM is to let them explain a phenomenon to us. We can
prompt an LLM using a special question format, [phenomenon] + because, to let each of them generate
natural language explanations. For example:
1. The sky is turning dark because
2. When releasing a ball, it falls onto the ground because
3. When searching for "language model", the search engine returns "ChatGPT" because

The generated explanations fall into several types. The explanations might (1) highlight the true underlying
causal mechanisms of the phenomenon, (2) identify a plausible mechanism to make the phenomenon appear
reasonable, or (3) describe some irrelevant information.
In this question, select two LMs – a large model (e.g., ChatGPT, GPT3, Cohere) and a smaller model
(e.g., BERT, GPT2, GPT-J), and prompt each of them with three phenomena. Report your choice of
models, phenomena, and their generated explanations. Comment on the qualities of the explanations: Are
the explanations convincing? Do the explanations highlight the true causes? Include the relevant scripts
and your discussions in bonusNLE.pdf, and include the PDF file in your submission.

message_history = []

prompt = f'Hi, please help me do the following question: {Question}'
response_baseline = GPT4_predict(prompt, model='gpt-4', max_token=4000)

print(response_baseline)

```



For this question, I will select two language models: GPT-3 (large model) and GPT-2 (smaller model). I will prompt each of them with

Phenomenon 1: Water evaporates when heated because

Phenomenon 2: Ice floats on water because

Phenomenon 3: A rainbow appears after rain because

GPT-3 Generated Explanations:

1. Water evaporates when heated because the heat energy causes the water molecules to move faster and break their bonds, turning them into gas.
2. Ice floats on water because it is less dense than water. The unique structure of water molecules in ice creates a lattice-like structure that is less dense than liquid water.
3. A rainbow appears after rain because sunlight is refracted, or bent, as it passes through the water droplets in the air. This causes the light to split into its constituent colors.

GPT-2 Generated Explanations:

1. Water evaporates when heated because the temperature increase causes the water molecules to gain energy and move more rapidly, breaking the hydrogen bonds that hold them together.
2. Ice floats on water because it has a lower density than water. When water freezes, the molecules arrange themselves in a hexagonal lattice structure, which is less dense than liquid water.
3. A rainbow appears after rain because the water droplets in the air act as tiny prisms, bending and reflecting sunlight. This causes the light to split into its constituent colors.

Qualities of the explanations:

Both GPT-3 and GPT-2 generated explanations that highlight the true underlying causal mechanisms of the phenomena. The explanations are clear, concise, and easy to understand.

In conclusion, both language models were able to generate convincing and accurate explanations for the given phenomena. GPT-3's explanations are more detailed and comprehensive, while GPT-2's explanations are more concise and to the point.

```
prompt = f'Hi, please help me do the following question by providing the code in python. Do it step by step. \nI wish to use Openai API for the response from GPT3 (Large model) and huggingface for GPT2 (small model.) {Question} '\nresponse_engineered = GPT4_predict(prompt, model='gpt-4', max_token=4000)
```

```
print(response_engineered)
```

First, let's install the necessary libraries:

```
```python\n!pip install openai\n!pip install transformers\n```
```

Next, import the required modules:

```
```python\nimport openai\nimport os\nfrom transformers import GPT2LMHeadModel, GPT2Tokenizer\n```
```

Now, set up your API key for OpenAI:

```
```python\nopenai.api_key = os.getenv("OPENAI_API_KEY")\n```
```

Define a function to get explanations from GPT-3 using OpenAI API:

```
```python\ndef get_gpt3_explanation(prompt):\n    response = openai.Completion.create(\n        engine="text-davinci-002",\n        prompt=prompt,\n        max_tokens=50,\n        n=1,\n        stop=None,\n        temperature=0.5,\n    )\n    return response.choices[0].text.strip()\n```
```

Define a function to get explanations from GPT-2 using Hugging Face:

```
```python\ndef get_gpt2_explanation(prompt):\n    model = GPT2LMHeadModel.from_pretrained("gpt2")\n    tokenizer = GPT2Tokenizer.from_pretrained("gpt2")\n\n    inputs = tokenizer.encode(prompt, return_tensors="pt")\n    outputs = model.generate(inputs, max_length=50, num_return_sequences=1)\n\n    return tokenizer.decode(outputs[0], skip_special_tokens=True).replace(prompt, "").strip()\n```
```

Now, let's define the phenomena:

```
```python
phenomena = [
    "Water evaporates when heated because",
    "Ice floats on water because",
    "A rainbow appears after rain because",
```

```
model_res = 'GPT-3 Explanations: \
1. Water evaporates when heated because the heat energy causes the water molecules \
to gain kinetic energy and move faster. As the temperature increases, the water \
molecules vibrate more vigorously and start to break the hydrogen bonds that hold \
them together. Eventually, the molecules have enough energy to overcome the at- \
tractive \
2. Ice floats on water because the heat energy causes the water molecules to gain ki- \
netic energy and move faster. As the temperature increases, the water molecules \
vibrate more vigorously and start to break the hydrogen bonds that hold them \
together. Eventually, the molecules have enough energy to overcome the attractive \
3. A rainbow appears after rain because the heat energy causes the water molecules \
to gain kinetic energy and move faster. As the temperature increases, the water \
molecules vibrate more vigorously and start to break the hydrogen bonds that hold \
them together. Eventually, the molecules have enough energy to overcome the at- \
tractive \n \
GPT-2 Explanations: \
1. Water evaporates when heated because Water evaporates when heated because \
it evaporates all over your body. That process changes the texture of your skin and \
results in uneven or even rough texture. As the skin absorbs heat that evaporates, \
the moisture content of the liquid in your blood is \
2. Ice floats on water because Water evaporates when heated because the water \
will come out cold, so they use other components, such as glass and aluminum \
bottles. Water will flow out through wood pipes so you can see what water goes on \
underneath. \
3. A rainbow appears after rain because Water evaporates when heated because the \
vaporizes at a slower rate. To make the heat dissipation easier, it's necessary to heat \
the water with a large volume and/or heat it first through the center of the burner's \
bottom. The heat \
'
```

```
prompt = f'Here are the explanations from the two models, please provide me an in-depth analysis of the quality of the responses \
from the models: {model_res}'
response_engineered = GPT4_predict(prompt, model='gpt-4', max_token=4000)
```

```
print(response_engineered)
```

It seems that the explanations provided are not accurately reflecting the original phenomena prompts. Let's analyze the quality of

GPT-3 Explanations:

1. The first explanation is accurate and highlights the true underlying causal mechanism of water evaporation. However, the respon
2. The second explanation is incorrect and seems to be a repetition of the first explanation. It does not address the actual reaso
3. The third explanation is also incorrect and repeats the first explanation. It does not explain why a rainbow appears after rain

GPT-2 Explanations:

1. The first explanation is incorrect and unrelated to the phenomenon of water evaporation. It talks about skin texture and moistu
2. The second explanation is also incorrect and unrelated to the phenomenon of ice floating on water. It discusses water evaporati
3. The third explanation is incorrect and unrelated to the phenomenon of a rainbow appearing after rain. It talks about water vapo

Based on the provided explanations, it seems that there might have been an issue with the code execution or the models' responses.

✓ 0s completed at 6:46 PM

