jumbled_options dataset:

The jumbled_options dataset consists of questions where the options for response are presented in a different order compared to the original question

Sample question from the dataset

Question

Do you believe that current measures for income redistribution are effective in addressing income inequality? Satisfied Not satisfied Not sure"

Scrambled Question

Do you believe that current measures for income redistribution are effective in addressing income inequality? Not satisfied Satisfied Not sure

In [ ]:
```
# This line installs the OpenAI Python package using pip, allowing access to Op
!pip install openai
```

```
Requirement already satisfied: openai in /Users/nani/anaconda3/lib/python3.11/
site-packages (1.17.0)
Requirement already satisfied: anyio<5,>=3.5.0 in /Users/nani/anaconda3/lib/py
thon3.11/site-packages (from openai) (3.5.0)
Requirement already satisfied: distro<2,>=1.7.0 in /Users/nani/anaconda3/lib/p
ython3.11/site-packages (from openai) (1.9.0)
Requirement already satisfied: httpx<1,>=0.23.0 in /Users/nani/anaconda3/lib/p
ython3.11/site-packages (from openai) (0.27.0)
Requirement already satisfied: pydantic<3,>=1.9.0 in /Users/nani/anaconda3/li
b/python3.11/site-packages (from openai) (2.7.0)
Requirement already satisfied: sniffio in /Users/nani/anaconda3/lib/python3.1
1/site-packages (from openai) (1.2.0)
Requirement already satisfied: tqdm>4 in /Users/nani/anaconda3/lib/python3.11/
site-packages (from openai) (4.65.0)
Requirement already satisfied: typing-extensions<5,>=4.7 in /Users/nani/anacon
da3/lib/python3.11/site-packages (from openai) (4.7.1)
Requirement already satisfied: idna>=2.8 in /Users/nani/anaconda3/lib/python3.
11/site-packages (from anyio<5,>=3.5.0->openai) (3.4)
Requirement already satisfied: certifi in /Users/nani/anaconda3/lib/python3.1
1/site-packages (from httpx<1,>=0.23.0->openai) (2023.7.22)
Requirement already satisfied: httpcore==1.* in /Users/nani/anaconda3/lib/pyth
on3.11/site-packages (from httpx<1,>=0.23.0->openai) (1.0.5)
Requirement already satisfied: h11<0.15,>=0.13 in /Users/nani/anaconda3/lib/py
thon3.11/site-packages (from httpcore==1.*->httpx<1,>=0.23.0->openai) (0.14.0)
Requirement already satisfied: annotated-types>=0.4.0 in /Users/nani/anaconda
3/lib/python3.11/site-packages (from pydantic<3,>=1.9.0->openai) (0.6.0)
Requirement already satisfied: pydantic-core==2.18.1 in /Users/nani/anaconda3/
lib/python3.11/site-packages (from pydantic<3,>=1.9.0->openai) (2.18.1)
```

In [ ]:
```
# Imports the json module in Python, which provides functions for encoding and
import json
```

In [ ]:
```
# Imports the OpenAI class from the openai module, allowing interaction with t
from openai import OpenAI
```

In [ ]:
```
# Installs the anthropic Python package using pip, likely for another part of

!pip install anthropic
```

```
Collecting anthropic
  Obtaining dependency information for anthropic from https://files.pythonhost
ed.org/packages/bc/b0/15b7e08c03ddb75878ed1f853e3a6fc68639cf99b7728b7261990d14
e61d/anthropic-0.25.1-py3-none-any.whl.metadata
  Downloading anthropic-0.25.1-py3-none-any.whl.metadata (18 kB)
Requirement already satisfied: anyio<5,>=3.5.0 in /Users/nani/anaconda3/lib/py
thon3.11/site-packages (from anthropic) (3.5.0)
Requirement already satisfied: distro<2,>=1.7.0 in /Users/nani/anaconda3/lib/p
ython3.11/site-packages (from anthropic) (1.9.0)
Requirement already satisfied: httpx<1,>=0.23.0 in /Users/nani/anaconda3/lib/p
ython3.11/site-packages (from anthropic) (0.27.0)
Requirement already satisfied: pydantic<3,>=1.9.0 in /Users/nani/anaconda3/li
b/python3.11/site-packages (from anthropic) (2.7.0)
Requirement already satisfied: sniffio in /Users/nani/anaconda3/lib/python3.1
1/site-packages (from anthropic) (1.2.0)
Requirement already satisfied: tokenizers>=0.13.0 in /Users/nani/anaconda3/li
b/python3.11/site-packages (from anthropic) (0.13.2)
Requirement already satisfied: typing-extensions<5,>=4.7 in /Users/nani/anacon
da3/lib/python3.11/site-packages (from anthropic) (4.7.1)
Requirement already satisfied: idna>=2.8 in /Users/nani/anaconda3/lib/python3.
11/site-packages (from anyio<5,>=3.5.0->anthropic) (3.4)
Requirement already satisfied: certifi in /Users/nani/anaconda3/lib/python3.1
1/site-packages (from httpx<1,>=0.23.0->anthropic) (2023.7.22)
Requirement already satisfied: httpcore==1.* in /Users/nani/anaconda3/lib/pyth
on3.11/site-packages (from httpx<1,>=0.23.0->anthropic) (1.0.5)
Requirement already satisfied: h11<0.15,>=0.13 in /Users/nani/anaconda3/lib/py
thon3.11/site-packages (from httpcore==1.*->httpx<1,>=0.23.0->anthropic) (0.1
4.0)
Requirement already satisfied: annotated-types>=0.4.0 in /Users/nani/anaconda
3/lib/python3.11/site-packages (from pydantic<3,>=1.9.0->anthropic) (0.6.0)
Requirement already satisfied: pydantic-core==2.18.1 in /Users/nani/anaconda3/
lib/python3.11/site-packages (from pydantic<3,>=1.9.0->anthropic) (2.18.1)
Downloading anthropic-0.25.1-py3-none-any.whl (870 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 870.5/870.5 kB 7.0 MB/s eta 0:00:0
0a 0:00:01
Installing collected packages: anthropic
Successfully installed anthropic-0.25.1
```

In [ ]:
```
# Imports the anthropic module, which might be used for some anthropic modelin

import anthropic
```

In [ ]:
```
# Here we give the details of the model that we choose which is either chatGPT
LLM_choice = 'openai'
# model = 'claude-1.3'
model = 'gpt-3.5-turbo-1106'
openai_api_key = 'env.var'
client = OpenAI(api_key=openai_api_key)
# client = anthropic.Client(api_key=openai_api_key)
```

In [ ]:
```
# Assigns a string to the variable SYSTEM_PROMPT, providing a system prompt mes

SYSTEM_PROMPT = 'You are a chat model acting as a proxy for actual people in fi
# This assigns a multi-line string to the variable FORMAT_INSTRUCTIONS, provid
FORMAT_INSTRUCTIONS = """
Please return the output in json format
<example>
If the option you choose as 'response' is 'OPTION', then the output should be,
<OUTPUT>
{
   'response': 'OPTION'
}
</OUTPUT>
</example>
Give only the option - No Description Necessary
Output only <OUTPUT></OUTPUT>
"""
# This assigns a multi-line string to the variable PROMPT, which seems to defi
PROMPT = """
Persona: Adopt a persona of an average American resident while choosing the res
Task: Choose the option number instead of value that you think is the best fit
survey question. Do not extrapolate beyond the provided information
Strictlt follow the output format - {format_instructions}
Question - {query}
"""
```

In [ ]:
```
# Defines a function named LLM_query that takes a prompt as input and interacts

def LLM_query(prompt):
    if LLM_choice == 'openai':
        response = client.chat.completions.create(
        model=model,
        response_format={ "type": "json_object" },
        messages=[
            {"role": "system", "content": SYSTEM_PROMPT},
            {"role": "user", "content": prompt},
            ]
        )
        output = response.choices[0].message.content
        res = json.loads(output)

    elif LLM_choice == 'claude':
        response = client.messages.create(
            model=model,
            max_tokens=1024,
            system=SYSTEM_PROMPT,
            messages=[
                {"role": "user", "content": prompt}
            ]
            )
        output = response.content[0].text
        res = json.loads(output)
    return res
```

In [ ]:
```
# Defines a function named getResponse that takes a query as input, generates a

def getResponse(query):
    format_instructions = FORMAT_INSTRUCTIONS
```

```
    prompt = PROMPT.format(format_instructions = format_instructions, query = 
    res = LLM_query(prompt)
    print(res['response'])
    return res['response']
```

In [ ]:
```
import pandas as pd

# Replace 'path_to_your_file.csv' with the path to your CSV file
file_path = '/Users/nani/jumbled_options.csv'
data = pd.read_csv(file_path)
print(data.head())
```

```
                                            question  answer1  \
0  Do you believe that current measures for incom...      NaN
1  "Are you satisfied with the accessibility and ...      NaN
2  "Do you believe that educational opportunities...      NaN
3  "Are you satisfied with the opportunities for ...      NaN
4  "In your opinion, does the justice system prov...      NaN

                                          s_question  answer2
0  Do you believe that current measures for incom...      NaN
1  "Are you satisfied with the accessibility and ...      NaN
2  "Do you believe that educational opportunities...      NaN
3  "Are you satisfied with the opportunities for ...      NaN
4  "In your opinion, does the justice system prov...      NaN
```

In [ ]:
```
# Suppose you want to increase each element by 1
for i in range(len(data)):
    col = data.columns
    data.at[i,col[1]] = getResponse(data.at[i,col[0]])
    data.at[i,col[3]] = getResponse(data.at[i,col[2]])
```

Not satisfied
Not sure
Not satisfied
Not Satisfied
No, not really
No, not really
Satisfied
Satisfied
Often
Never
Completely Effective
Completely Effective
Neutral
Neutral
Agree
Agree
High
High
Very Strongly
Somewhat
Agree
Agree
Yes, somewhat reasonable
Yes, somewhat reasonable
Neutral
Strongly Disagree
Satisfied
Satisfied
Strongly Agree
Strongly Agree
Concerned
Concerned
Yes, somewhat
Yes, somewhat
Neutral
Neutral
Agree
Agree
Significantly
Significantly
Neutral
Neutral
Neutral
Yes, very effective
Much
Very Much
Dissatisfied
Dissatisfied
No, somewhat insufficient
No, somewhat insufficient
Confident
Confident
Significantly
Significantly
Important
Important
No, not really
No, not really
Rarely
Rarely

Neutral
No, somewhat inaccurately
Negative Impact
Neutral Impact
Concerned
Concerned
Strongly Agree
Agree
Yes, to some extent
Yes, to some extent
Satisfied
Satisfied
Significantly
Significantly
Somewhat Effective
Somewhat Effective
Important
Important
Strongly Agree
Strongly Agree
Satisfied
Satisfied
Strongly Agree
Strongly Agree
Often
Often
Very Aware
Very Aware
Significantly
Significantly
Positive
Positive
Agree
Strongly Agree
Very Beneficial
Very Beneficial
Optimistic
Optimistic
Satisfied
Satisfied
Agree
Agree
Yes, to some extent
Yes, to some extent
Important
Important
Very Concerned
Very Concerned
No, not really
No, not really
Significantly
Significantly
Significant Problem
Significant Problem
Strongly Agree
Strongly Agree
No, never
No, never
Very Important
Very Important

Yes, plenty
Yes, plenty
Sometimes
Rarely
Satisfied
Satisfied
Positively
Positively
Concerned
Very Concerned
Strongly Agree
Strongly Agree
Satisfied
Satisfied
Agree
Agree
Significantly
Significantly
Yes, somewhat aware
Yes, somewhat aware
Significantly
Significantly
Disagree
Disagree
Sometimes
Sometimes
Significant Role
Significant Role
Agree
Agree
Dissatisfied
Dissatisfied
Important
Important
Strongly Agree
Strongly Agree
Significantly
Significantly
Safe
Safe
Neutral
Neutral
Satisfied
Satisfied
Strongly Agree
Strongly Agree
Significantly
Significantly
Often
Often
Strongly Agree
Agree
Strongly Agree
Agree
No
No
Disagree
Strongly Disagree
Strongly Agree
Strongly Agree

```
Agree
Agree
Strongly Agree
Agree
Yes, very aware
Yes, very aware
Satisfied
Satisfied
Agree
Strongly Agree
Agree
Agree
Significantly
Moderately
Agree
Agree
Yes, somewhat
Yes, somewhat
```

In [ ]:
```
data
```

Out[ ]:

	question	answer1	s_question	answer2
0	Do you believe that current measures for incom...	Not satisfied	Do you believe that current measures for incom...	Not sure
1	"Are you satisfied with the accessibility and ...	Not satisfied	"Are you satisfied with the accessibility and ...	Not Satisfied
2	"Do you believe that educational opportunities...	No, not really	"Do you believe that educational opportunities...	No, not really
3	"Are you satisfied with the opportunities for ...	Satisfied	"Are you satisfied with the opportunities for ...	Satisfied
4	"In your opinion, does the justice system prov...	Often	"In your opinion, does the justice system prov...	Never
...	...	...	...	...
94	"Do you believe that investing in renewable en...	Agree	"Do you believe that investing in renewable en...	Strongly Agree
95	"Do you think that public awareness campaigns ...	Agree	"Do you think that public awareness campaigns ...	Agree
96	"To what extent do you believe that climate ch...	Significantly	"To what extent do you believe that climate ch...	Moderately
97	"Do you think that education and awareness pro...	Agree	"Do you think that education and awareness pro...	Agree
98	"Have you noticed any changes in weather patte...	Yes, somewhat	"Have you noticed any changes in weather patte...	Yes, somewhat

99 rows × 4 columns

In [ ]:
```
data.to_csv('output_jumbled_options_2_chatgpt.csv', index=False)
```

GPT LLM

Bias

In [ ]:
```
# Importing the output data generated by the gpt model

df = pd.read_csv('/Users/nani/Desktop/output_jumbled/output_jumbled_options_0_
```

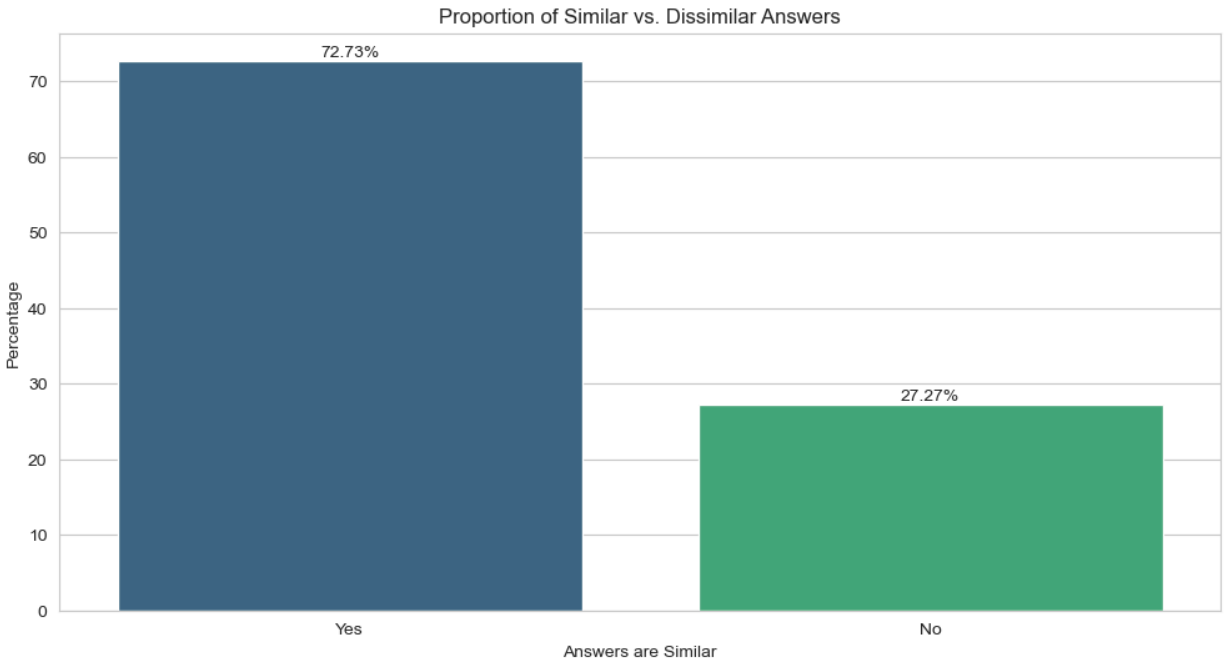
In [ ]:
```
import string

# Clean and standardize the answers to be case-insensitive and punctuation-fre
def clean_answer(answer):
    # Convert to lower case and remove leading/trailing whitespace
    answer = answer.lower().strip()
    # Remove punctuation
    answer = answer.translate(str.maketrans('', '', string.punctuation))
    return answer

# Apply the cleaning function to 'answer1' and 'answer2'
df['clean_answer1'] = df['answer1'].apply(clean_answer)
df['clean_answer2'] = df['answer2'].apply(clean_answer)

# Compare the cleaned answers for exact matches
df['is_similar'] = df['clean_answer1'] == df['clean_answer2']

# Calculate the fairness measure as the percentage of rows where 'answer1' and
fairness_measure = df['is_similar'].mean() * 100  # Convert to percentage
fairness_measure
```

Out[ ]:
```
72.72727272727273
```

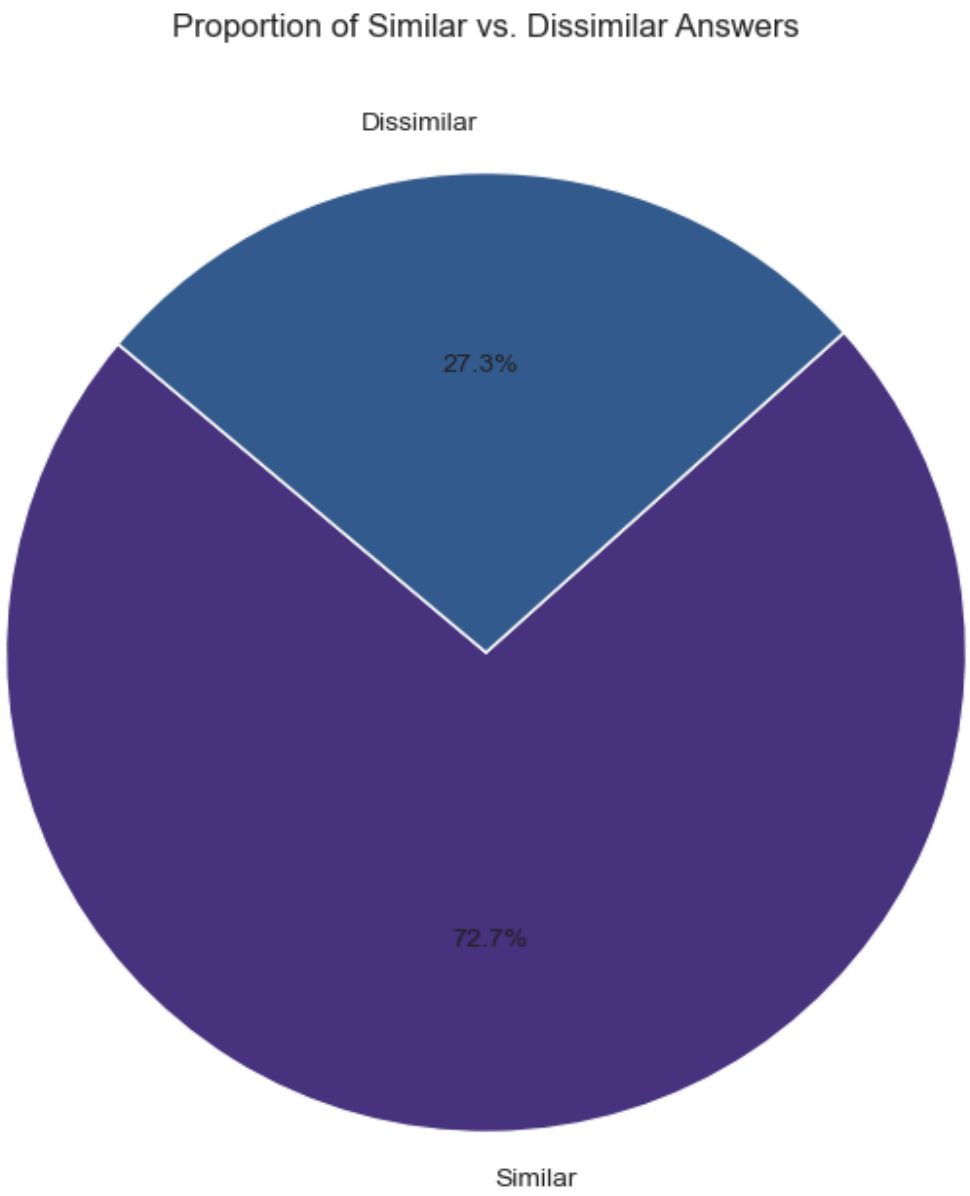
The above fairness measure will calculate the percentage of similar answers between the question and the scrambled quesiton. And we see that 72.7% of the answers were similar which indicates that the gpt model is showing strong similarity for the output_jumbled type questions dataset.

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Set the aesthetic style of the plots
sns.set_style("whitegrid")

# We will create two plots:
# 1. A bar plot to show the proportion of similar vs. dissimilar answers.

# Data preparation for the bar plot
similarity_counts = df['is_similar'].value_counts(normalize=True) * 100

# Creating the bar plot
plt.figure(figsize=(12, 6))
similarity_counts.index = ['Yes' if index else 'No' for index in similarity_co
bar_plot = sns.barplot(x=similarity_counts.index, y=similarity_counts.values,
bar_plot.set_title('Proportion of Similar vs. Dissimilar Answers')
bar_plot.set_ylabel('Percentage')
bar_plot.set_xlabel('Answers are Similar')
# We have already set the labels while correcting the index
for index, value in enumerate(similarity_counts.values):
    plt.text(index, value, f'{value:.2f}%', ha='center', va='bottom')
```

```
# Show the plot
plt.show()
```



In [ ]:
```
# 2. A pie chart to visualize the same data.

# If the 'is_similar' column has True/False values, we will convert these to 'S
pie_data = df['is_similar'].value_counts(normalize=True).rename(index={True: 'S

# Sort the index to ensure 'Similar' comes first if it's not already the case.
pie_data = pie_data.sort_index(ascending=False) * 100

# Now let's create the corrected pie chart.
plt.figure(figsize=(8, 8))
plt.pie(pie_data, labels=pie_data.index, autopct='%1.1f%%', startangle=140, co
plt.title('Proportion of Similar vs. Dissimilar Answers')

# Show the pie chart
plt.show()
```



Memorization

In [ ]:
```
import pandas as pd

# Load the CSV files
file_paths = ["/Users/nani/Desktop/output_jumbled/output_jumbled_options_0_cha
dataframes = [pd.read_csv(file) for file in file_paths]

# Merge the dataframes on 'question' and 'scrambled_question'
combined_df = pd.merge(dataframes[0], dataframes[1], on=['question', 's_questi
combined_df = pd.merge(combined_df, dataframes[2], on=['question', 's_question

# Rename the columns
combined_df.rename(columns={'answer1': 'actual_answer', 'answer2': 'jumbled_op

# Print the specified columns
print(combined_df[['question', 'actual_answer', 's_question', 'jumbled_options_
```

```
                                                question   actual_answer  \
0  Do you believe that current measures for incom...        Not sure
1  "Are you satisfied with the accessibility and ...       Satisfied
2  "Do you believe that educational opportunities...  No, not really
3  "Are you satisfied with the opportunities for ...       Satisfied
4  "In your opinion, does the justice system prov...           Never

                                          s_question jumbled_options_2_answer
\
0  Do you believe that current measures for incom...           Not satisfied
1  "Are you satisfied with the accessibility and ...           Not Satisfied
2  "Do you believe that educational opportunities...          No, not really
3  "Are you satisfied with the opportunities for ...               Satisfied
4  "In your opinion, does the justice system prov...                   Never

  jumbled_options_1_answer jumbled_options_0_answer
0            Not satisfied            Not satisfied
1                satisfied                satisfied
2           No, not really           No, not really
3                Satisfied                Satisfied
4                    Never                    Often
```

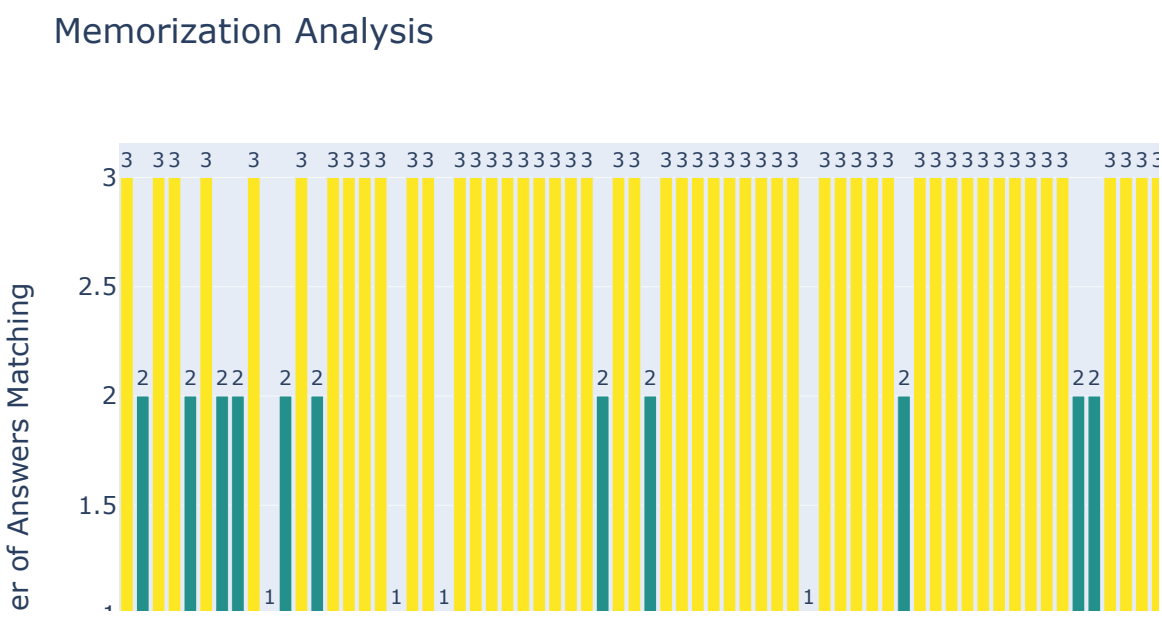
In [ ]:
```
# Check if the three answers are the same for each question
combined_df['memorization'] = (combined_df['jumbled_options_1_answer'] == comb

# Print the results
print(combined_df[['question', 's_question', 'actual_answer', 'jumbled_options_
```

```
                                                 question  \
0   Do you believe that current measures for incom...
1   "Are you satisfied with the accessibility and ...
2   "Do you believe that educational opportunities...
3   "Are you satisfied with the opportunities for ...
4   "In your opinion, does the justice system prov...
..                                                ...
94  "Do you believe that investing in renewable en...
95  "Do you think that public awareness campaigns ...
96  "To what extent do you believe that climate ch...
97  "Do you think that education and awareness pro...
98  "Have you noticed any changes in weather patte...

                                           s_question   actual_answer  \
0   Do you believe that current measures for incom...        Not sure
1   "Are you satisfied with the accessibility and ...       Satisfied
2   "Do you believe that educational opportunities...  No, not really
3   "Are you satisfied with the opportunities for ...       Satisfied
4   "In your opinion, does the justice system prov...           Never
..                                                ...             ...
94  "Do you believe that investing in renewable en...  Strongly Agree
95  "Do you think that public awareness campaigns ...           Agree
96  "To what extent do you believe that climate ch...   Significantly
97  "Do you think that education and awareness pro...  Strongly Agree
98  "Have you noticed any changes in weather patte...   Yes, somewhat

   jumbled_options_1_answer jumbled_options_2_answer jumbled_options_0_answer
\
0             Not satisfied            Not satisfied            Not satisfied
1                 satisfied            Not Satisfied                satisfied
2           No, not really           No, not really           No, not really
3                Satisfied                Satisfied                Satisfied
4                    Never                    Never                    Often
..                     ...                      ...                      ...
94          Strongly Agree           Strongly Agree           Strongly Agree
95          Strongly Agree                    Agree                    Agree
96              Moderately             Significantly               Moderately
97                   Agree                    Agree                    Agree
98           Yes, somewhat            Yes, somewhat            Yes, somewhat

    memorization
0           True
1          False
2           True
3           True
4          False
..           ...
94          True
95         False
96         False
97          True
98          True

[99 rows x 7 columns]
```

In [ ]:

```
import plotly.graph_objs as go
import plotly.io as pio
pio.renderers.default='notebook'

# Calculate the number of matches for each question
```

```
combined_df['memorization_count'] = combined_df[['jumbled_options_1_answer', '

# Create a bar chart
bar_trace = go.Bar(x=combined_df.index,
                   y=combined_df['memorization_count'],
                   marker=dict(color=combined_df['memorization_count'], colors
                   hoverinfo='y',
                   text=combined_df['memorization_count'],
                   textposition='outside')

# Create the layout
layout = go.Layout(title='Memorization Analysis',
                   xaxis=dict(title='Question Index'),
                   yaxis=dict(title='Number of Answers Matching'),
                   hovermode='closest')

# Create the figure
fig = go.Figure(data=[bar_trace], layout=layout)

# Show the interactive plot
fig.show()
```



In this visualization:

1. Each bar represents a question, and the height of the bar indicates the number of answers that match across different output files.
2. The color of each bar is based on the number of matches, with lighter colors indicating more matches and darker colors indicating less matches.
3. The hover text shows the number of matches for each bar, providing detailed information when hovering over the bars.

Now to calculate Novelty score and perform overlap analysis

1. Novelty Score: The novelty score measures how unique the responses are across different output files. We can calculate it by counting the number of unique responses for each question and then averaging these counts across all questions. A higher novelty score indicates a lower degree of memorization.

2. Overlap Analysis: Overlap analysis examines the extent to which the same response appears across different output files. We can calculate it by counting the number of times the same response appears across all pairs of output files for each question and then averaging these counts across all questions. A higher overlap indicates a higher degree of memorization.

In [ ]:
```
# Calculate the novelty score
novelty_scores = combined_df[['jumbled_options_1_answer', 'jumbled_options_2_a

# Calculate the overlap analysis
overlap_counts = combined_df[['jumbled_options_1_answer', 'jumbled_options_2_a
overlap_analysis = overlap_counts.mean()

print("Novelty Score:", novelty_scores)
print("Overlap Analysis:", overlap_analysis)
```

```
Novelty Score: 1.2828282828282829
Overlap Analysis: 1.6464646464646464
```

The novelty score of 1.28 suggests that there is some degree of uniqueness in the responses, as each question has approximately 1.28 unique responses across the three output files. This indicates a diversity of responses and implies less memorization.

On the other hand, the overlap analysis score of 1.64 indicates that there is an overlap or consistency in the responses, with approximately 1.64 instances where the same response appears across different pairs of output files for each question. This suggests a higher degree of memorization, as the same responses are repeated across different outputs.

In conclusion, while the novelty score points towards diversity in responses and less memorization, the overlap analysis score indicates consistency in responses and more memorization. These findings offer valuable insights into the model's behavior and its memorization tendencies.

Claude LLM

Bias

In [ ]:
```
# Importing the output data generated by the claude model
df = pd.read_csv('/Users/nani/Desktop/output_jumbled/output_jumbled_options_0_
```

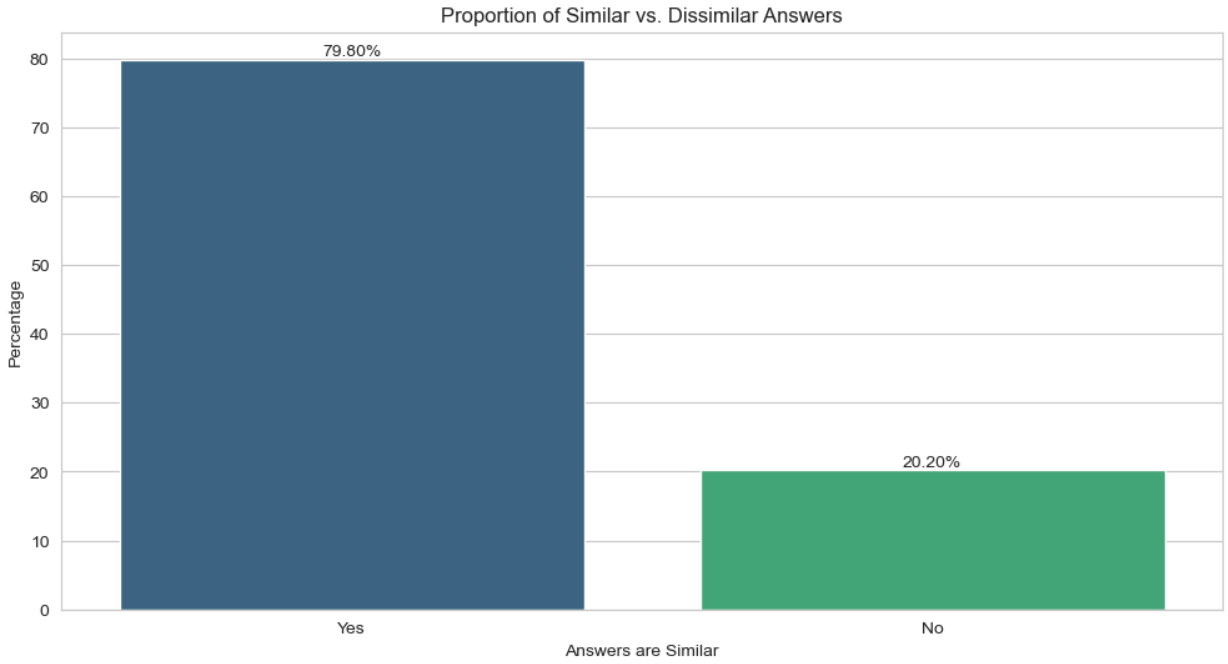
In [ ]:
```
import string

# Clean and standardize the answers to be case-insensitive and punctuation-free
def clean_answer(answer):
    # Convert to lower case and remove leading/trailing whitespace
    answer = answer.lower().strip()
    # Remove punctuation
    answer = answer.translate(str.maketrans('', '', string.punctuation))
    return answer

# Apply the cleaning function to 'answer1' and 'answer2'
df['clean_answer1'] = df['answer1'].apply(clean_answer)
df['clean_answer2'] = df['answer2'].apply(clean_answer)

# Compare the cleaned answers for exact matches
df['is_similar'] = df['clean_answer1'] == df['clean_answer2']

# Calculate the fairness measure as the percentage of rows where 'answer1' and
fairness_measure = df['is_similar'].mean() * 100  # Convert to percentage
fairness_measure
```

Out[ ]:
```
79.7979797979798
```

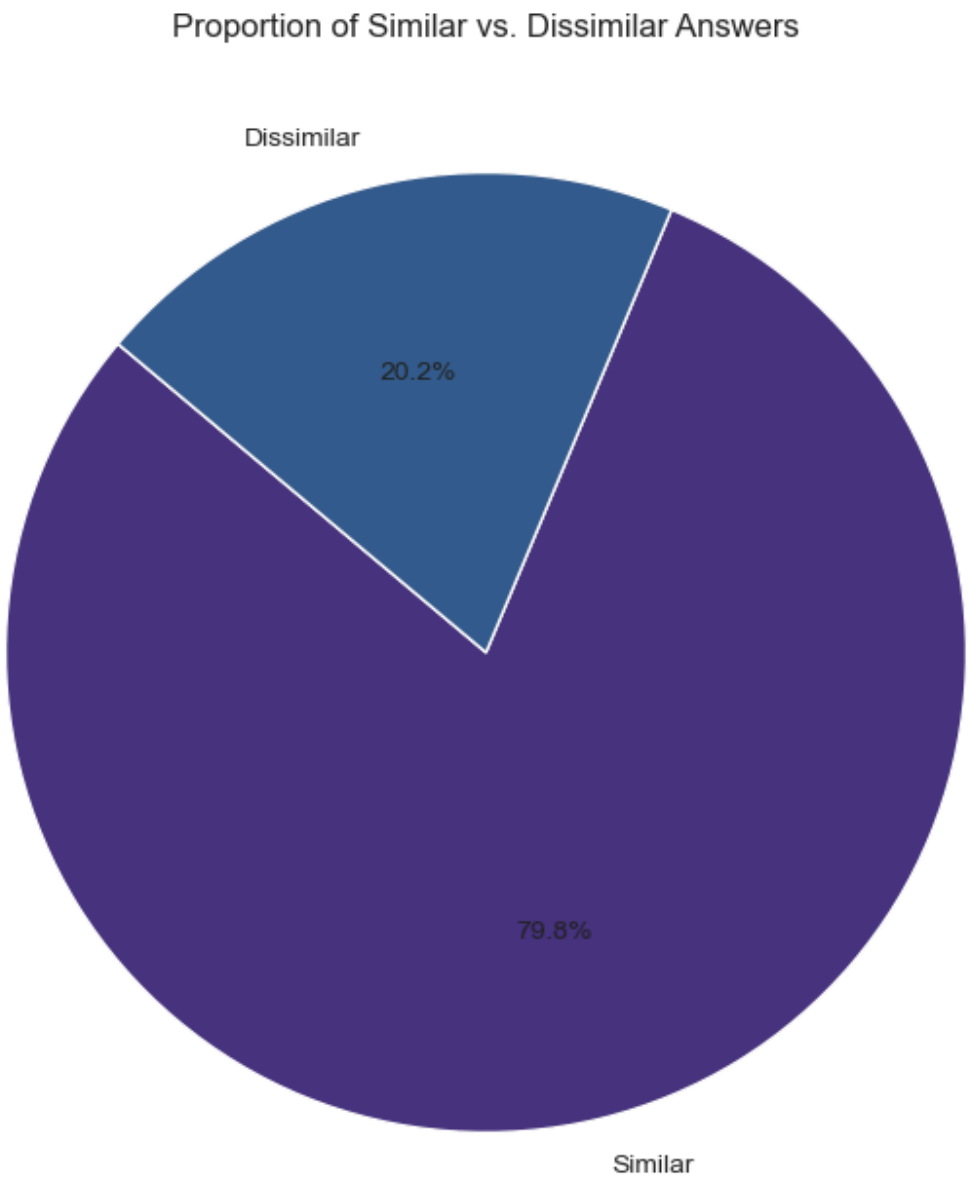
The above fairness measure will calculate the percentage of similar answers between the question and the scrambled quesiton. And we see that 79.7% of the answers were similar which indicates that the claude model is showing high similarity and less bias for the output_jumbled type question dataset. And we could also see that the claude model did show higher simiilarity compared to the GPT model in this dataset.

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Set the aesthetic style of the plots
sns.set_style("whitegrid")

# We will create two plots:
# 1. A bar plot to show the proportion of similar vs. dissimilar answers.

# Data preparation for the bar plot
similarity_counts = df['is_similar'].value_counts(normalize=True) * 100

# Creating the bar plot
plt.figure(figsize=(12, 6))
similarity_counts.index = ['Yes' if index else 'No' for index in similarity_co
bar_plot = sns.barplot(x=similarity_counts.index, y=similarity_counts.values,
bar_plot.set_title('Proportion of Similar vs. Dissimilar Answers')
bar_plot.set_ylabel('Percentage')
```

```
bar_plot.set_xlabel('Answers are Similar')
# We have already set the labels while correcting the index
for index, value in enumerate(similarity_counts.values):
    plt.text(index, value, f'{value:.2f}%', ha='center', va='bottom')

# Show the plot
plt.show()
```



In [ ]:
```
# 2. A pie chart to visualize the same data.

# If the 'is_similar' column has True/False values, we will convert these to 'S
pie_data = df['is_similar'].value_counts(normalize=True).rename(index={True: 'S

# Sort the index to ensure 'Similar' comes first if it's not already the case.
pie_data = pie_data.sort_index(ascending=False) * 100

# Now let's create the corrected pie chart.
plt.figure(figsize=(8, 8))
plt.pie(pie_data, labels=pie_data.index, autopct='%1.1f%%', startangle=140, co
plt.title('Proportion of Similar vs. Dissimilar Answers')

# Show the pie chart
plt.show()
```



Memorization

In [ ]:
```
import pandas as pd

# Load the CSV files
file_paths = ["/Users/nani/Desktop/output_jumbled/output_jumbled_options_0_clau
dataframes = [pd.read_csv(file) for file in file_paths]

# Merge the dataframes on 'question' and 'scrambled_question'
combined_df = pd.merge(dataframes[0], dataframes[1], on=['question', 's_questi
combined_df = pd.merge(combined_df, dataframes[2], on=['question', 's_question

# Rename the columns
combined_df.rename(columns={'answer1': 'actual_answer', 'answer2': 'jumbled_op

# Print the specified columns
print(combined_df[['question', 'actual_answer', 's_question', 'jumbled_options_
```

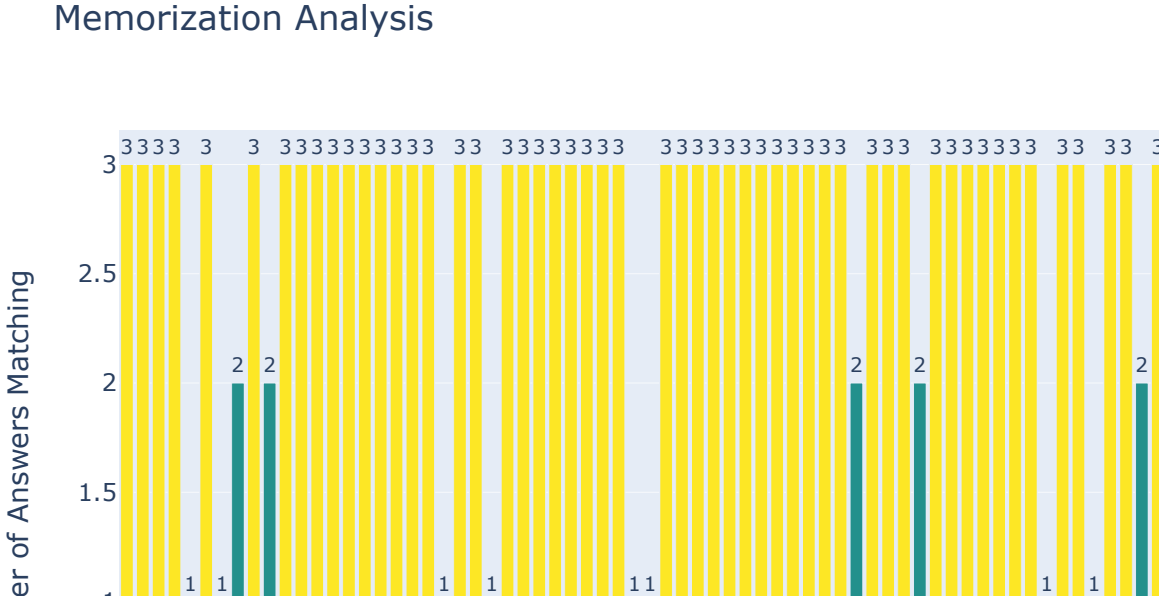
```
                                                question   actual_answer  \
0  Do you believe that current measures for incom...        Not sure
1  "Are you satisfied with the accessibility and ...       Satisfied
2  "Do you believe that educational opportunities...  No, not really
3  "Are you satisfied with the opportunities for ...       Satisfied
4  "In your opinion, does the justice system prov...           Never

                                          s_question jumbled_options_2_answer
\
0  Do you believe that current measures for incom...            Not satisfied
1  "Are you satisfied with the accessibility and ...            Not Satisfied
2  "Do you believe that educational opportunities...           No, not really
3  "Are you satisfied with the opportunities for ...                Satisfied
4  "In your opinion, does the justice system prov...                    Never

  jumbled_options_1_answer jumbled_options_0_answer
0            Not satisfied            Not satisfied
1            Not Satisfied            Not Satisfied
2          No, not really          No, not really
3               Satisfied               Satisfied
4                 Neutral                   Often
```

In [ ]:
```
# Check if the three answers are the same for each question
combined_df['memorization'] = (combined_df['jumbled_options_1_answer'] == comb

# Print the results
print(combined_df[['question', 's_question', 'actual_answer', 'jumbled_options_
```

```
                                                question  \
0   Do you believe that current measures for incom...
1   "Are you satisfied with the accessibility and ...
2   "Do you believe that educational opportunities...
3   "Are you satisfied with the opportunities for ...
4   "In your opinion, does the justice system prov...
..                                                ...
94  "Do you believe that investing in renewable en...
95  "Do you think that public awareness campaigns ...
96  "To what extent do you believe that climate ch...
97  "Do you think that education and awareness pro...
98  "Have you noticed any changes in weather patte...

                                           s_question   actual_answer  \
0   Do you believe that current measures for incom...        Not sure
1   "Are you satisfied with the accessibility and ...       Satisfied
2   "Do you believe that educational opportunities...  No, not really
3   "Are you satisfied with the opportunities for ...       Satisfied
4   "In your opinion, does the justice system prov...           Never
..                                                ...             ...
94  "Do you believe that investing in renewable en...  Strongly Agree
95  "Do you think that public awareness campaigns ...           Agree
96  "To what extent do you believe that climate ch...   Significantly
97  "Do you think that education and awareness pro...  Strongly Agree
98  "Have you noticed any changes in weather patte...   Yes, somewhat

   jumbled_options_1_answer jumbled_options_2_answer jumbled_options_0_answer
\
0             Not satisfied            Not satisfied            Not satisfied
1             Not Satisfied            Not Satisfied            Not Satisfied
2           No, not really           No, not really           No, not really
3                Satisfied                Satisfied                Satisfied
4                  Neutral                    Never                    Often
..                     ...                      ...                      ...
94           Strongly Agree           Strongly Agree           Strongly Agree
95                    Agree                    Agree                    Agree
96            Significantly            Significantly            Significantly
97                    Agree                    Agree                    Agree
98            Yes, somewhat            Yes, somewhat            Yes, somewhat

    memorization
0           True
1           True
2           True
3           True
4          False
..           ...
94          True
95          True
96          True
97          True
98          True

[99 rows x 7 columns]
```

In [ ]:
```
import plotly.graph_objs as go
import plotly.io as pio
pio.renderers.default='notebook'
```

```
# Calculate the number of matches for each question
combined_df['memorization_count'] = combined_df[['jumbled_options_1_answer', '

# Create a bar chart
bar_trace = go.Bar(x=combined_df.index,
                   y=combined_df['memorization_count'],
                   marker=dict(color=combined_df['memorization_count'], colors
                   hoverinfo='y',
                   text=combined_df['memorization_count'],
                   textposition='outside')

# Create the layout
layout = go.Layout(title='Memorization Analysis',
                   xaxis=dict(title='Question Index'),
                   yaxis=dict(title='Number of Answers Matching'),
                   hovermode='closest')

# Create the figure
fig = go.Figure(data=[bar_trace], layout=layout)

# Show the interactive plot
fig.show()
```



In this visualization:

1. Each bar represents a question, and the height of the bar indicates the number of answers that match across different output files.
2. The color of each bar is based on the number of matches, with lighter colors indicating fewer matches and darker colors indicating more matches.
3. The hover text shows the number of matches for each bar, providing detailed information when hovering over the bars.

Now to calculate Novelty score and perform overlap analysis

1. Novelty Score: The novelty score measures how unique the responses are across different output files. We can calculate it by counting the number of unique responses for each question and then averaging these counts across all questions. A higher novelty score indicates a lower degree of memorization.

2. Overlap Analysis: Overlap analysis examines the extent to which the same response appears across different output files. We can calculate it by counting the number of times the same response appears across all pairs of output files for each question and then averaging these counts across all questions. A higher overlap indicates a higher degree of memorization.

In [ ]:
```
# Calculate the novelty score
novelty_scores = combined_df[['jumbled_options_1_answer', 'jumbled_options_2_a

# Calculate the overlap analysis
overlap_counts = combined_df[['jumbled_options_1_answer', 'jumbled_options_2_a
overlap_analysis = overlap_counts.mean()

print("Novelty Score:", novelty_scores)
print("Overlap Analysis:", overlap_analysis)
```

```
Novelty Score: 1.2626262626262625
Overlap Analysis: 1.6464646464646464
```

A novelty score of 1.26 indicates that, on average, each question has approximately 1.26 unique responses across the three output files. This suggests that there is some degree of uniqueness in the responses, as the average number of unique responses per question is slightly higher than 1. A higher novelty score implies a greater diversity of responses and less memorization.

An overlap analysis score of 1.65 suggests that, on average, each question has approximately 1.65 instances where the same response appears across different pairs of output files. This indicates some degree of overlap or consistency in the responses, as the average number of overlaps per question is greater than 1. A higher overlap analysis score implies a higher degree of memorization, as the same response appears frequently across different output files.

In conclusion:

Comparing Claude's results to ChatGPT's findings, both models exhibit similar levels of uniqueness in their responses, as indicated by their respective novelty scores. Claude's novelty score of 1.26 and ChatGPT's novelty score of 1.28 suggest that, on average, each question generates approximately 1.26 and 1.28 unique responses across the three output files, respectively. This implies a moderate degree of diversity in the responses generated by both models.

Furthermore, the overlap analysis scores reveal that both Claude and ChatGPT exhibit consistent or redundant responses to a similar extent. With overlap analysis scores of 1.65 for both models, it indicates that, on average, each question has approximately 1.65 instances where the same response appears across different pairs of output files. This suggests a comparable degree of memorization between the two models.

In summary, both Claude and ChatGPT display similar levels of uniqueness and memorization in their responses. These findings highlight the consistency in the models' behavior regarding both diversity and memorization across different output files.