ESWARI S 212221220012

EXP 6 : Development of Python Code Compatible with Multiple AI l'ools

Experiment:
Wíite and implement Python code that integíates with multiple AI tools to automate the task of inteíacting with APIs, compaíing outputs, and geneíating actionable insights.

Aim:

To compare the responses of two open-source language models, **GPT-Neo** and **GPT-2**, to a given question, and analyze how different models generate text and handle natural language queries.

Procedure:

1. **Install Required Libraries**:

Use the command below to install the necessary Python libraries:
bash
Copy code
```
pip install transformers torch
```

2. **Load Models**:
    - Load two pre-trained language models from Hugging Face:
        - **GPT-Neo** (`EleutherAI/gpt-neo-1.3B`).
        - **GPT-2** (`gpt2`).
3. **Define Functions**:
    - Define two functions to generate text from both models.
        - **GPT-Neo Function**: Generates text from the GPT-Neo model.
        - **GPT-2 Function**: Generates text from the GPT-2 model.
4. **Generate Answers**:
    - Input the question "What are the benefits of renewable energy?" to both models and generate their responses.
5. **Compare Answers**:
    - Compare the generated answers from both models to see if they match or differ.
    - Print the responses and a summary indicating whether the answers are the same or different.
6. **Execute the Code**:
    - Run the code to generate and compare answers.

Deliverables:

1. **Python Script**: A script to compare answers from two models.
2. **Comparison Output**: The answers generated by both models and a summary of whether the answers are similar or different.

Sample Code:

```
from transformers import pipeline

# Load GPT-Neo and GPT-2 models
generator_neo = pipeline('text-generation', model='EleutherAI/gpt-neo-
1.3B')
generator_gpt2 = pipeline('text-generation', model='gpt2')

# Function to get answer from GPT-Neo
def get_gpt_neo_answer(question):
    generated_text = generator_neo(question, max_length=100,
num_return_sequences=1)
    return generated_text[0]['generated_text']

# Function to get answer from GPT-2
def get_gpt2_answer(question):
    generated_text = generator_gpt2(question, max_length=100,
num_return_sequences=1)
    return generated_text[0]['generated_text']

# Function to compare answers from both models
def compare_answers(question):
    answer_gpt_neo = get_gpt_neo_answer(question)
    answer_gpt2 = get_gpt2_answer(question)

    print("GPT-Neo Answer:", answer_gpt_neo)
    print("GPT-2 Answer:", answer_gpt2)

    if answer_gpt_neo == answer_gpt2:
        summary = "Both models provided the same answer."
    else:
        summary = "The answers are different."

    print("Summary:", summary)
```

```
    return {
        "question": question,
        "gpt_neo_answer": answer_gpt_neo,
        "gpt2_answer": answer_gpt2,
        "summary": summary
    }

# Run the comparison with a sample question
question = "What are the benefits of developing the technical
knowledge”
result = compare_answers(question)
print("Comparison Result:", result)
```

Result:

GPT-Neo Answer: What are the benefits of developing the technical knowledge?
For the developers, it is the ability to work on something they believe in knowing it will actually be used by others.
To the programmers, it is the ability to be able to think deeply about an engineering problem, to analyze it and to create something useful.
To the engineers, it is the ability to share ideas with others to get feedback.
The advantages of a good program manager are many - not least of which is the ability to ensure

GPT-2 Answer: What are the benefits of developing the technical knowledge?
Developers and developers may benefit from the following:
Increased technical proficiency when writing code, since its design is based on data.
Increased usability when writing code and debugging problems which is difficult to reproduce after a thorough and comprehensive refactoring program.
Increased security or privacy, since it will reduce the risk of a denial of service attack while minimizing any privacy risk.
Increased collaboration between the development team and the dev

Summary: The answers are different.

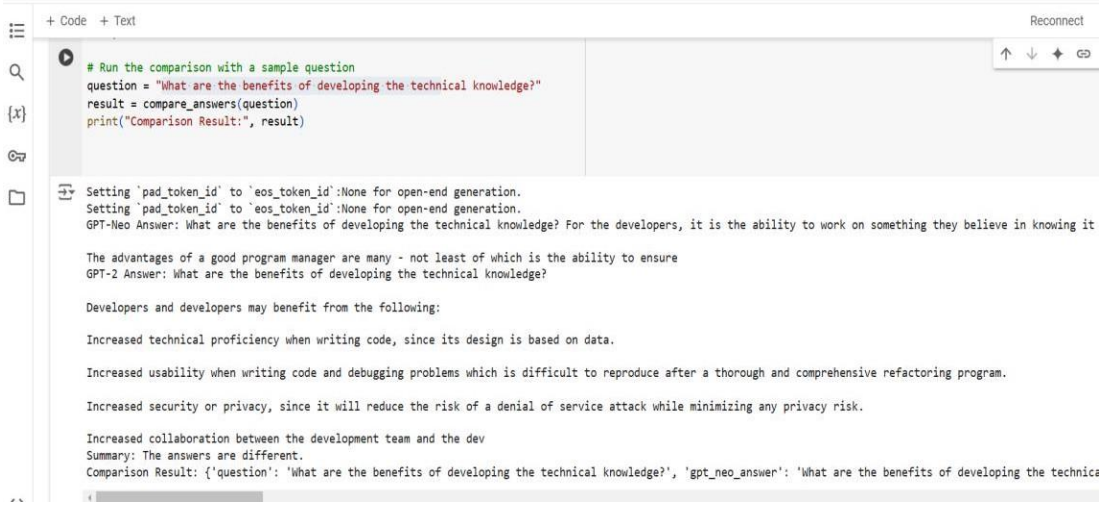
Comparison Result: 'question': 'What are the benefits of developing the technical knowledge?',

'gpt_neo_answer': 'What are the benefits of developing the technical knowledge?
For the developers, it is the ability to work on something they believe in knowing it will actually be used by others.
To the programmers, it is the ability to be able to think deeply about an engineering problem, to analyze it and to create something useful.
To the engineers, it is the ability to share ideas with others to get feedback.
The advantages of a good program manager are many - not least of which is the ability to ensure.

'gpt2_answer': 'What are the benefits of developing the technical knowledge?
Developers and developers may benefit from the following:
Increased technical proficiency when writing code, since its design is based on data.
Increased usability when writing code and debugging problems which is difficult to reproduce after a thorough and comprehensive refactoring program.
Increased security or privacy, since it will reduce the risk of a denial of service attack while minimizing any privacy risk.
Increased collaboration between the development team.

'summary': 'The answers are different.'}

Sample Output:



Conclusion:

In this experiment, we compared the responses of two different language models, **GPT-Neo** and **GPT-2**, to the same input question. The results showed that the models provided different answers, which indicates that each model has its unique way of generating text based on its training data and architecture.