Text Summarization with LiteLLM

Complete Example with Real Input and Output

Sample Input Text

Sample Article: "Recent advances in artificial intelligence have transformed multiple industries. Machine learning algorithms can now process vast amounts of data to identify patterns and make predictions with unprecedented accuracy. In healthcare, AI systems assist doctors in diagnosing diseases by analyzing medical images and patient data. The financial sector uses AI for fraud detection, algorithmic trading, and risk assessment. Meanwhile, autonomous vehicles rely on computer vision and neural networks to navigate complex traffic scenarios. Despite these achievements, challenges remain in AI ethics, data privacy, and ensuring algorithmic fairness across diverse populations. As AI continues to evolve, researchers emphasize the importance of responsible development and deployment of these powerful technologies."

Actual Output Results

Brief Summary:
AI has revolutionized multiple industries through advanced machine learning algorithms that analyze data and make accurate predictions. While AI shows great promise in healthcare, finance, and autonomous vehicles, challenges remain in ethics, privacy, and algorithmic fairness.

Medium Summary:
Artificial intelligence has significantly transformed various industries by using machine learning algorithms to process large datasets and make highly accurate predictions. AI applications now span healthcare for medical diagnosis, finance for fraud detection and trading, and transportation through autonomous vehicles. However, the field faces ongoing challenges related to AI ethics, data privacy protection, and ensuring fair algorithmic outcomes across different population groups.

Token Usage: 89 prompt tokens, 67 completion tokens, 156 total tokens

Complete Code Implementation

```
# summarization_example.py
from litellm import completion

def summarize_text(text, length="brief"):
    # Define length instructions
    length_instructions = {
        "brief": "in 1-2 sentences",
        "medium": "in 3-4 sentences",
        "detailed": "in 5-6 sentences with key points"
    }

    instruction = length_instructions.get(length, "in 2-3 sentenc
es")

    try:
        response = completion(
            model="gpt-3.5-turbo",
            messages=[
                {
                    "role": "system",
                    "content": f"You are an expert summarizer. Pr
ovide clear, concise summaries that capture the main points. Summ
arize the given text {instruction}."
                },
                {
                    "role": "user",
                    "content": f"Please summarize this text:\n\n
{text}"
                }
            ],
            temperature=0.3,  # Lower temperature for consistent
summaries
            max_tokens=150
        )

        return response.choices[0].message.content.strip()

    except Exception as e:
        return f"Summarization failed: {str(e)}"

# Sample article text (shortened for example)
article = """Recent advances in artificial intelligence have tran
sformed multiple industries..."""

# Test different summary lengths
brief_summary = summarize_text(article, "brief")
medium_summary = summarize_text(article, "medium")

print("Brief Summary:")
print(brief_summary)
print("\nMedium Summary:")
print(medium_summary)
```

Pro Tips

- **Temperature 0.3:** Lower temp produces more focused and consistent summaries.
- **System Prompts:** Clearly define the AI's role, style, and length requirements for better control.
- **Length Control:** Use 'max_tokens' as a hard limit to manage output length and cost.
- **Batch Processing:** For multiple documents, loop through them to call the function efficiently.