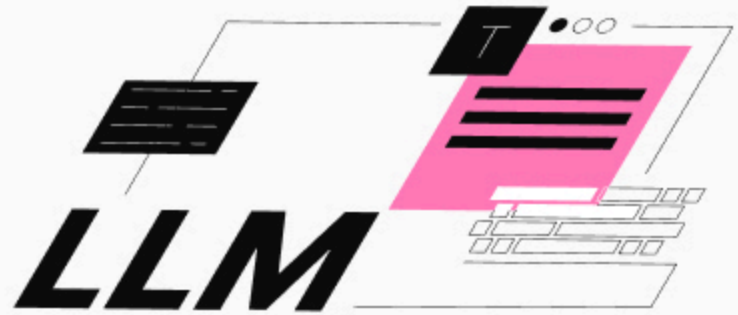
Large Language Model (LLM) for Sentiment Analyzing and Image Reasoning

What is LLMs?

Large Language Models (LLMs) are advanced artificial intelligence systems designed to understand, generate, and manipulate human language on a large scale.

pixelplex

Top 10 applications of large language models



1. Content generation
2. Translation and localization
3. Search and recommendation
4. Virtual assistants
5. Code development
6. Sentiment analysis
7. Question answering
8. Market research
9. Education
10. Classification

Project summary

- Build a lightweight system using an LLM (ChatGPT)
- Implement two key functionalities:
    - Sentiment analysis of text inputs
    - Image-based reasoning through textual description

To obtain the API key, navigate to https://platform.openai.com/account/api-keys.

- free usage limit of the OpenAI API was not sufficient for this project, so I spent around $20 to access additional API credits and complete the assignment.
- Access to the OpenAI API may be blocked on certain restricted networks (e.g., school Wi-Fi), requiring workarounds like VPNs or mobile hotspots.

Task 1: Sentiment Analysis

- The system takes a text input from the user.
- The text is sent to the LLM with a carefully designed prompt asking the model to analyze its sentiment.
- The model returns a result classifying the input as positive, negative, or neutral, along with a brief summary.

```
response = client.chat.completions.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role": "user", "content": "Hi"}
    ]
)

print(response.choices[0].message.content)
```

```
Hello! How can I assist you today?
```

```
text = "Happy Graduate"

sentiment = response.choices[0].message.content
print(sentiment)
if sentiment == "Positive":
    print("😊")
elif sentiment == "Negative":
    print("😔")
else:
    print("😐")
```

```
Positive 😊
```

Task 2: Image-Related Reasoning

- Instead of processing raw image data directly, publicly available image URLs are used.
- Text prompts describing the image URLs are fed into the LLM.
- The model generates descriptive text based on the imagined content of the images, simulating a simple form of visual reasoning.

Ask the model to describe what it sees

```
async_resp = await openai_llm.astream_chat(messages=[msg])
async for delta in async_resp:
    print(delta.delta, end="")
```

```
1. A close-up of an orange tabby cat with striking amber eyes, looking directly at the camera. The background is blurred, highlighting the cat's face and whiskers.

2. A white cat with heterochromia, featuring one yellow eye and one blue eye. The cat is lying on a soft, multicolored blanket, with a focused and curious expression.
```