

log-analyzer-LLM

June 5, 2025

0.1 Log Analyzer with LLM

Log Analyzer with LLM is an intelligent application that leverages OpenAI models (e.g., GPT-3.5) to analyze raw log files. It reads log content, summarizes key insights, detects anomalies or errors, and provides actionable recommendations—helping system admins, security analysts, and researchers understand system behavior and respond to incidents more effectively.

0.1.1 What does it do?

- Accepts .log or .txt files as input
- Preprocesses and chunks long logs for efficient analysis
- Uses ChatGPT to generate:
 - **Summary of system activity**
 - **Detection of issues, errors, and anomalies**
 - **Recommendations for performance or security improvements**

0.1.2 Use cases

- Security analysts investigating incident logs
- Developers debugging server or application logs
- Cybersecurity students learning LLM-powered threat detection

0.2 Install library

The first step before start deploying the application, we should make sure the need library is installed. The main library is openai, install it by the next command.

```
[ ]: !pip install openai
```

0.3 Prepare OpenAI API key

```
[ ]: import openai
import os

# Set your OpenAI API key (preferably via environment)
openai.api_key = os.getenv("OPENAI_API_KEY")
# Or you can type it directly (not secure)
# openai.api_key = "your_api_key_here"
```

0.4 Read log file

```
[ ]: from pathlib import Path
from IPython.display import Markdown

# Ensure the correct file name
log_file_path = "sample1.log"

log_content = Path(log_file_path).read_text()
```

0.5 Data truncation to avoid exceeding the limit

```
[ ]: #Truncate data to avoid exceeding the maximum number of symbols
log_excerpt = log_content[:4000]
```

0.6 Analysis with ChatGPT

```
[ ]: response = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role": "system", "content": "You are a cybersecurity expert analyzing_
↪raw log files. Provide a summary, detected issues, and recommendations in_
↪markdown."},
        {"role": "user", "content": f"Here is the log content:
↪\n\n{log_excerpt}"}
    ],
    temperature=0.3
)

analysis_result = response.choices[0].message.content
Markdown(analysis_result)
```

```
[ ]:
```