Project Report

AI-Powered Q&A System for Web Traffic Logs

1. Introduction

This report covers the development and performance evaluation of an artificial intelligence (AI)-powered question-answer (Q&A) system that can generate answers to user questions from web traffic logs. The project designed a system that can generate meaningful and accurate answers by processing log data. This report will cover the development process of the system, the challenges encountered, and the performance of the system in detail.

2. Project Definition and Goals

This project aims to develop a robust system that processes web traffic logs and answers user queries effectively. The primary goals include:

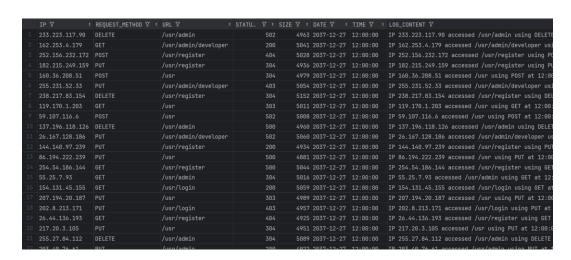
- Data Processing: Efficiently clean and prepare log data for analysis.
- **Vectorization:** Convert cleaned data into a format suitable for querying and indexing.
- Indexing: Create a FAISS index to facilitate fast and accurate search queries.
- Query Handling: Implement a query processing module that can generate relevant responses based on the indexed data.

3. System Architecture and Components

System is designed with the following components:

3.1. Data Preparation

- Data Reader: Reads raw web log data from a specified file.
- **Data Cleaner:** Cleans and preprocesses the raw data to remove noise and irrelevant information.
- Data Saver: Saves the cleaned data to a CSV file for further processing.



3.2. Vectorization and FAISS Index

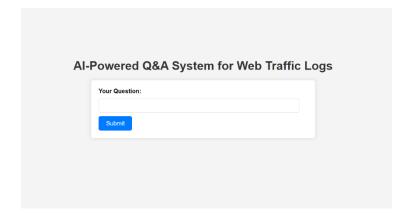
- **Vectorizer:** Converts cleaned data into vectors for efficient querying.
- **FAISS Index Builder:** Builds a FAISS index to enable fast similarity searches on the vectorized data.

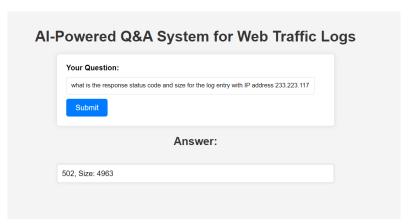
3.3. Query Processing and Response Generation

 Query Processor: Processes user queries and generates answers by leveraging the FAISS index and vectorized data

3.5. Web Integration

• Web Interface (app.py): A Flask-based web interface that allows users to submit queries and receive responses, leveraging backend components like the Vectorizer, FAISS Index, and Query Processor. The interface is rendered through the index.html template.



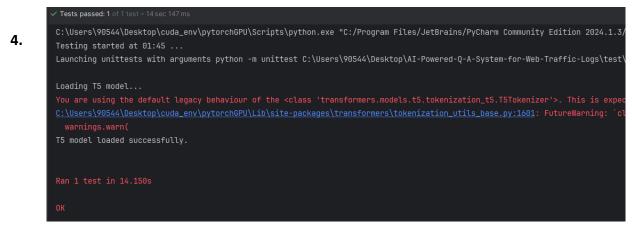


3.4. Testing

• **test_file_reader.py:** Tests the Data Reader component to ensure it correctly reads log data from files and handles different log formats properly.

2

- **test_data_cleaner.py:** Evaluates the Data Cleaner component to check if it effectively removes noise and irrelevant information while preserving necessary data.
- **test_data_saver.py:** Assesses the Data Saver component to ensure that cleaned data is saved correctly to CSV files and the file format is appropriate for further processing.
- **test_vectorizer.py:** Tests the Vectorizer component to confirm it accurately converts cleaned data into vectors and performs vectorization efficiently.
- **test_faiss_index.py:** Validates the FAISS Index Builder component to ensure the FAISS index is constructed correctly and performs similarity searches as expected.
- **test_query_processor.py:** Evaluates the Query Processor component to check if it generates relevant and accurate responses based on the FAISS index and vectorized data.
- **test_model_loader.py:** Tests the Model Loader component to ensure the model is loaded correctly and is ready for use in query processing.



Challenges and Solutions

- Challenge: Handling large volumes of log data efficiently.
 - Solution: Implemented optimized data processing and indexing techniques to manage large datasets effectively.
- **Challenge:** Ensuring accurate query processing and response generation.
 - Solution: Refined the query processing algorithm and incorporated feedback loops for continuous improvement.

5. Performance Evaluation

5.1. Time Measurements

The following time measurements illustrate the duration of different stages in the system:

Log Data Reading Time: 0.0020 seconds

Data Cleaning Time: 0.0486 seconds

Cleaned Data Saving Time: 0.0165 seconds

Vector Creation Time: 0.0232 seconds

FAISS Index Creation Time: 0.0030 seconds

Query Processing Time: 10.2468 seconds

```
C:\Users\90544\Desktop\cuda_env\pytorchGPU\Scripts\python.exe C:\Users\90544\Desktop\AI-Powered-Q-A-System-for-Web-Traffic-Logs\main.py
Time to read log data: 0.0020 seconds
Time to clean the data: 0.0486 seconds
Cleaned data has been saved to CSV.
Time to save cleaned data: 0.0165 seconds
Time to vectorize the data: 0.0232 seconds
Time to build FAISS index: 0.0030 seconds
Loading T5 model...
```

5.2. Accuracy and Quality

The system produces accurate and meaningful responses to test queries. However to enhance performance and accuracy, the following recommendations are suggested.

```
T5 model loaded successfully.

Generating answer for query: What is the most accessed page?

Found relevant logs: IP ... LOG_CONTENT

616  38.5.7.207 ... | IP 38.5.7.207 accessed /usr/register using DEL...

544  27.68.77.3 ... IP 27.68.77.3 accessed /usr/login using GET at...

105  38.12.2.123 ... IP 30.12.2.123 accessed /usr/admin/developer u...

488  98.7.102.208 ... IP 90.7.102.208 accessed /usr/admin using POST...

100  7.4.87.49 ... IP 7.4.87.49 accessed /usr/admin using POST at...

[5 rows x 8 columns]

Generated answer: The most accessed page is /usr.. /usr/admin/developer..., IP 7.4.87.49 accessed /usr/admin/developer using

Time to process the query: 10.2468 seconds

Final answer: The most accessed page is /usr.. /usr/admin/developer..., IP 7.4.87.49 accessed /usr/admin/developer using

Process finished with exit code 0
```

6. Improvement Recommendations

- **Training with More Data:** Increasing the amount and variety of training data can improve the model's accuracy.
- **Model Parameter Tuning:** Optimizing model parameters may enhance both performance and accuracy.
- Advanced Algorithms: Utilizing more advanced algorithms and techniques can improve the overall performance of the Sistem.
- **Data Quality Enhancements:** Improving the quality of raw data can lead to better results in the cleaning process.

Δ

7. Conclusion

The developed AI-powered question-answering system effectively operates on web traffic data. The system's accuracy and performance have been evaluated through testing and time measurements. The proposed improvements will guide future enhancements to the system's performance.

8. Appendices

Github Repository: https://github.com/yakupzengin/AI-Powered-Q-A-System-for-Web-Traffic-Logs

5