Flipkart Grid 5.0

Project Documentation: Compliance Monitoring and Enforcement through Log Analysis

Objective: Develop a system to analyse logs, system configurations, access controls, and user privileges to ensure compliance with security policies and standards using large language models (LLMs).

1. Data Generation:

Synthetic Data Creation:

We began by generating synthetic data to emulate real-world logs and rules.

The logs included attributes like UserID, Action, Resource, Timestamp, IP Address, and Status.

We also generated rule sets representing various policies the logs should follow.

Data Expansion:

The initial dataset was expanded to enhance the training potential for the LLMs.

Efforts were made to cover a diverse range of scenarios in the logs and rules to mimic real-world complexity.

2. Model Selection & Training:

Large Language Models (LLMs):

Given their capability to understand context, LLMs were chosen as the primary tool.

The hope was that LLMs could infer relationships between logs and rules, providing actionable insights.

Fine-Tuning GPT-2:

We initially opted to fine-tune the GPT-2 model on our dataset.

The model was trained to analyse the relationship between logs and rules and generate outputs that indicate compliance or non-compliance and provide insights.

3. Challenges, Adjustments & Learnings:

Dataset Size & Overfitting:

Initially, we started with smaller datasets. However, it became evident that LLMs, especially GPT-2, required substantial data for effective fine-tuning.

Overfitting emerged as a challenge, indicating a potential need for even larger datasets or regularization techniques.

Model Limitations with GPT-2:

Verbose Outputs: The model sometimes produced overly detailed and redundant responses.

Echoing Prompts: At times, the model returned parts of the input as output without providing new insights.

Memory Constraints: The larger GPT-2 variants demanded significant computational resources, causing issues on platforms like Google Colab's free tier.

Exploring Alternative Models:

We experimented with other models like DistilBert to find a balance between computational efficiency and capability.

While these models were resource-friendly, they presented their own set of challenges, especially in generating insights in context.

Tokenization & Model Inputs:

Proper tokenization and configuration of model inputs posed challenges.

Attention masks, padding, and other model-related configurations were crucial for optimal performance.

4. Lessons & Takeaways:

Importance of Data:

Quality and quantity of data play a pivotal role in training LLMs.

For future endeavours, ensuring diverse and ample data is crucial.

Iterative Nature of ML Projects:

Adjusting strategies, models, and data based on interim results is a hallmark of ML projects.

Continuous iteration, evaluation, and refinement are key to success.

Computational Demands of LLMs:

LLMs, while powerful, require substantial computational resources.

Ensuring access to adequate computational power is vital when working with such models.

5. Conclusion:

The journey of this project highlighted the intricacies and challenges of working with Large Language Models. While the desired results were not achieved given the current constraints, the groundwork has been established. With enhanced computational resources and more extensive, fine-tuned models, this project holds promise. The lessons learned throughout this process will undoubtedly serve as valuable guidance for future endeavors in similar domains.