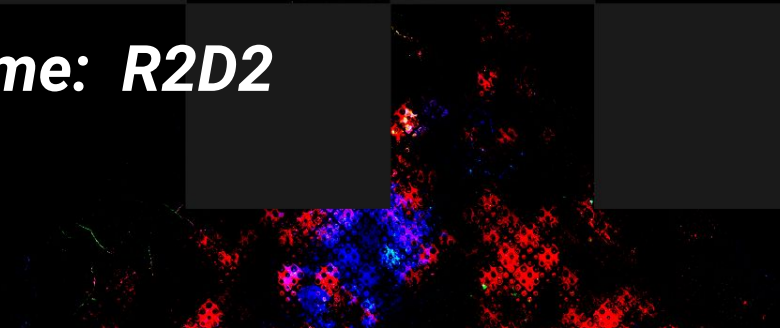
Flipkart GRiD 5.0

Problem Statement Title: Compliance Monitoring and Enforcement through Log Analysis using Large Language Models (Infosec Engineering)

Team Name: R2D2



Team members details

Team Name	R2D2		
Institute Name/Names	Indian Institute of Technology Jodhpur		
Team Members >	1 (Leader)	2	3
Name	Adarsh Palaskar	Palak Singh	
Batch	2024	2024	

FlipLogGPT: Compliance Monitoring and Enforcement through Log Analysis using Large Language Models

Deliverables:

1. Detect compliance breaches in logs and system policies.
2. Provide actionable insights to fix the issues

TechStack: Python, LangChain, GPT4All, GPT-J, Chroma and SentenceTransformers

Github:

Glossary:

1. LLM: Large Language Model
2. REACT: Synerzing Reasoning And Acting in language models
3. LoRA: Low Rank Adaptation of Large Language Models

Use-Cases

According to order of impact

P0. Identifying errors and non-compliant activities in the security policies by comparing them with the compliance documents rulesets.

P1. Analyzing logs, system configurations and access control, to detect potential cybersecurity threats, report them and suggest actionable insights to mitigate through them.

Example model output:

```
> source_documents\compliance_document.txt:
Server Information
Server: Apache Web Server
IP Address: 192.168.0.X
Operating System: [Specify the OS version and any relevant details]
Compliance Measures
1. Access Control
Access to the server is restricted to authorized personnel only.
User accounts are created with strong passwords and adhere to the organization's passw
Regular review and update of user accounts to ensure access privileges are up to date.
```

Model identifying which compliance was not followed by the security policy, using the source documents provided.

Solution Statement

Proposed Approach

1. The user can upload compliance documents as ruleset in the model. These documents are split into into chunks and stored as vectorized embeddings, in the Chroma Database.
2. The user can then include the security policy/log documents, which are prompt engineered using REACT methodology and few-shot inference.
3. Using semantic search, the model finds relevant information in the created knowledge base (using QA retrieval in Langchain).
4. This is finally fed to the model to generate the desired output.

QA Retrieval: https://aclanthology.org/2021.emnlp-main.678.pdf

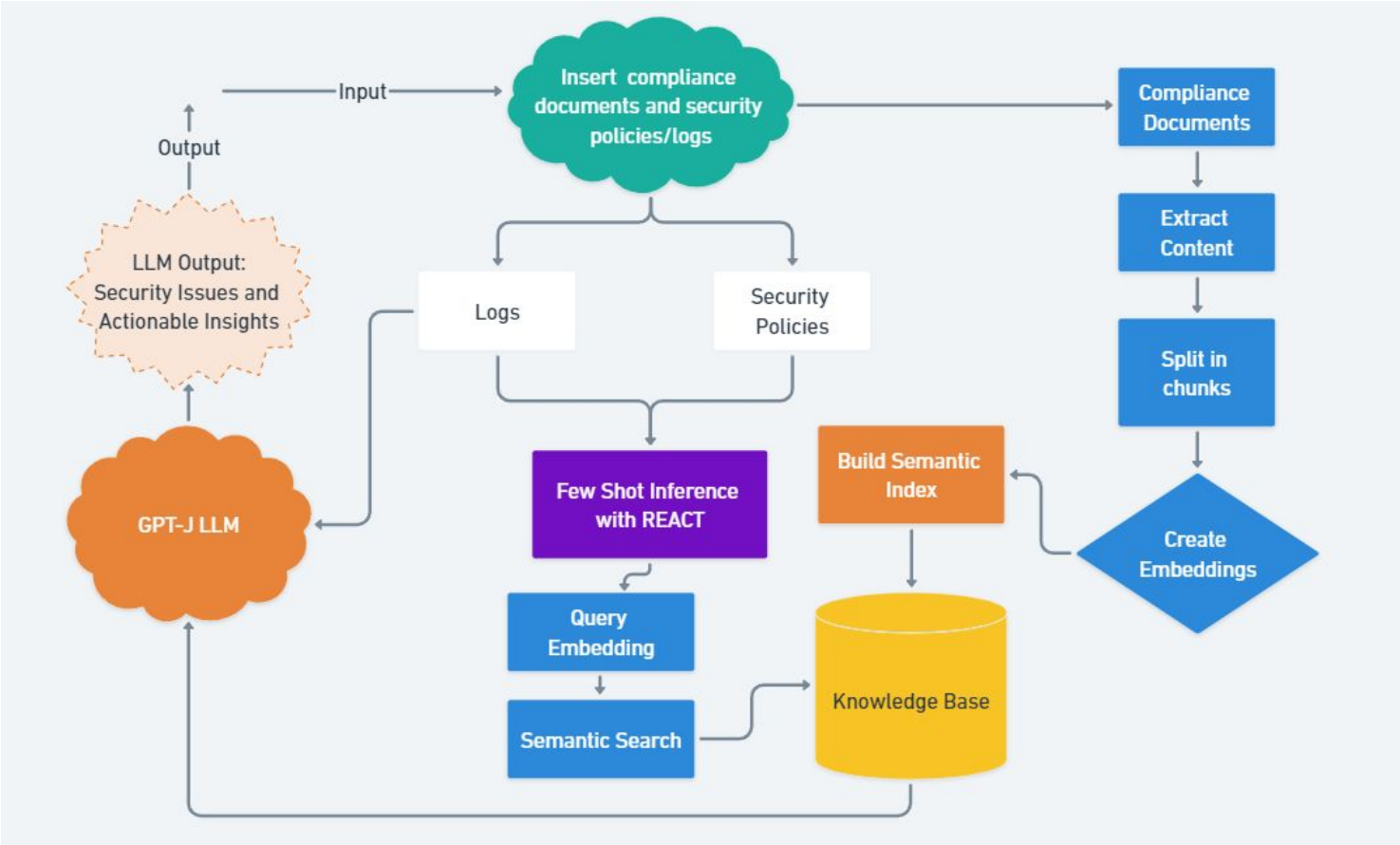
LLM Used

GPT-J: 6 B parameters, approximated to 8-bit integers(Due to computational limitations).
Trained using GPT-3.5 output dataset using transfer learning.

Supported Document Extensions

.csv
.doc/ .docx
.ppt/ .pptx
.txt
.eml
.md
.html
.pdf

Block-Diagram of the approach:

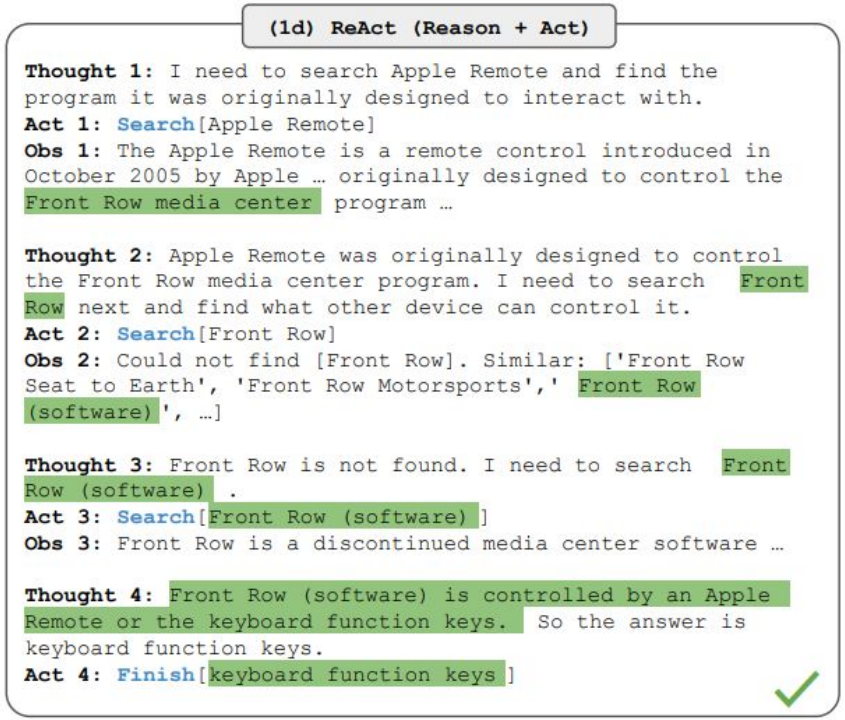


REACT: Synerzing Reasoning And Acting in language models

References: 1. REACT: https://arxiv.org/pdf/2210.03629.pdf 2.Few-shot inference: https://arxiv.org/pdf/2207.04237.pdf

Using REACT with few shot inference

1. REACT combines task-oriented actions with verbal reasoning and is achieved through Langchain and manual REACT.
2. Using Few-shot inference(Prompting the model with examples to obtain the expected output), greatly improves the model performance.
3. Using these techniques in a prompt significantly impacted the results, and the model was able to give more relevant results.



Limitations:

1. GPT-J has very few parameters and have less capacity to capture intricate patterns and nuances in data. It struggles with tasks that require a deep understanding of context or complex relationships.
2. The model weights have been scaled to 8-bit integer versions, so make it small enough to run on a CPU. This has reduced the model accuracy significantly.
3. Due to lack of computational availability and limited resources and unavailability of a proper dataset, we were not able to train the model on the specific task or use efficient training techniques like LoRA.
4. Smaller models have a limited vocabulary and might not understand context as deeply as larger models. This could lead to errors or misunderstandings in generated text.

Future Scope:

LoRA: Low Rank Adaptation of Large Language Models

References: LoRA: https://arxiv.org/pdf/2106.09685.pdf

Training limited parameters using LoRA

1. Fine-tuning large models is computationally expensive and requires substantial resources.
2. LoRA proposes reducing the rank of certain matrices in the model during fine-tuning to decrease computational requirements.
3. LoRA balances computational efficiency with task-specific adaptation to ensure optimal model performance.

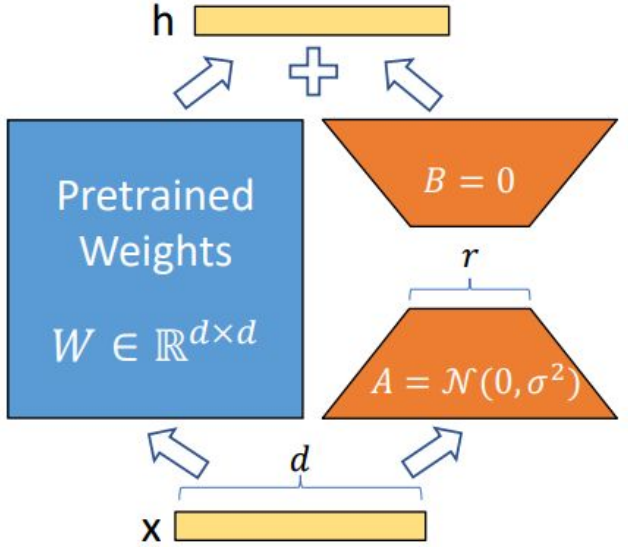
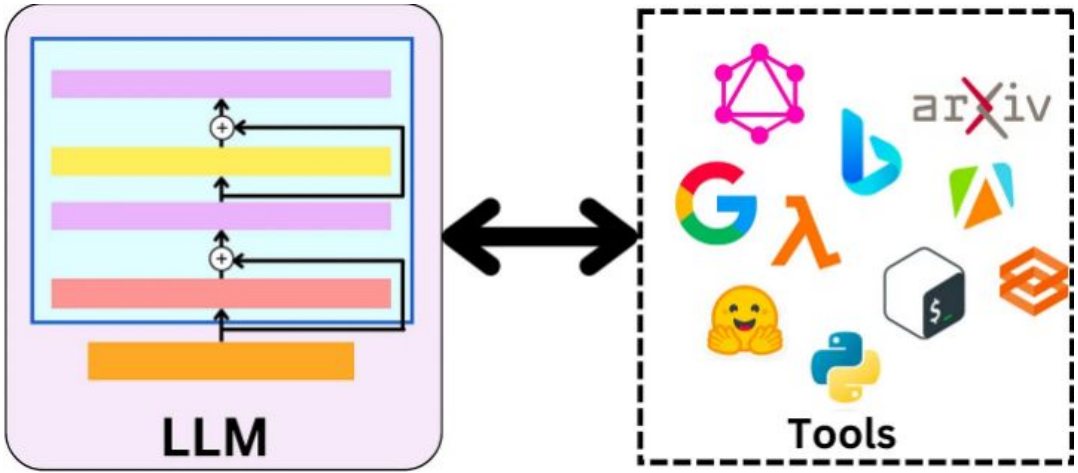


Figure 1: Our reparametrization. We only train A and B.

Future Scope:

Using LangChain Agents and other chain types :



1. Connecting the LLM to a external agent thorough langchain can help it access real-time information, which can be significant while detecting cybersecurity threats.
2. To handle log files larger than the model context window, other chain types like “map_reduce”, “refine”, and “map-rerank” can be used.

Flipkart GRiD 5.0

Thank You