CAI Assignment 2

Group ID: ADL Group 132

Group Members Name with Student ID:

1. PARIKH VEDANT ASHISH ALPA 2023AA05369
2. KANSARA HARSH BHARAT BHAVINI 2023AA05351

Documents:

- Jio Financial Services Limited Annual Report 2022-2023 (196 pages)
- Jio Financial Services Limited Annual Report 2023-2024 (233 pages)

```
!pip install langchain langchain-community faiss-cpu sentence-transformers rank-bm25 transformers streamlit pypdf llm_guard
```

```
Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Downloading typing_inspect-0.9.0-py3-none-any.whl (8.8 kB)
Downloading tldextract-5.1.3-py3-none-any.whl (104 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 104.9/104.9 kB 11.3 MB/s eta 0:00:00
Downloading unidiff-0.7.5-py2.py3-none-any.whl (14 kB)
Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Downloading requests_file-2.1.0-py2.py3-none-any.whl (4.2 kB)
Building wheels for collected packages: fuzzysearch
  Building wheel for fuzzysearch (setup.py) ... done
  Created wheel for fuzzysearch: filename=fuzzysearch-0.7.3-py3-none-any.whl size=21203 sha256=fe15f12c628764d20489783cf23e06e2452836c8f440d43063962d152070d7
  Stored in directory: /root/.cache/pip/wheels/be/ad/2e/bd664c4b01e5535ee4387d8a491311f61467a43627597684a7
Successfully built fuzzysearch
Installing collected packages: unidiff, phonenumbers, watchdog, structlog, regex, rank-bm25, python-dotenv, pypdf, pycryptodome, oldest-supported-numpy, nvid
  Attempting uninstall: regex
    Found existing installation: regex 2024.11.6
    Uninstalling regex-2024.11.6:
      Successfully uninstalled regex-2024.11.6
  Attempting uninstall: nvidia-nvjitlink-cu12
    Found existing installation: nvidia-nvjitlink-cu12 12.5.82
    Uninstalling nvidia-nvjitlink-cu12-12.5.82:
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
  Attempting uninstall: nvidia-curand-cu12
    Found existing installation: nvidia-curand-cu12 10.3.6.82
    Uninstalling nvidia-curand-cu12-10.3.6.82:
      Successfully uninstalled nvidia-curand-cu12-10.3.6.82
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.3.61
    Uninstalling nvidia-cufft-cu12-11.2.3.61:
      Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
  Attempting uninstall: nvidia-cuda-runtime-cu12
    Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
    Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-nvrtc-cu12
    Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
    Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-cupti-cu12
    Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
    Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
  Attempting uninstall: nvidia-cublas-cu12
    Found existing installation: nvidia-cublas-cu12 12.5.3.2
    Uninstalling nvidia-cublas-cu12-12.5.3.2:
      Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
  Attempting uninstall: nvidia-cusparse-cu12
    Found existing installation: nvidia-cusparse-cu12 12.5.1.3
    Uninstalling nvidia-cusparse-cu12-12.5.1.3:
      Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
  Attempting uninstall: nvidia-cudnn-cu12
    Found existing installation: nvidia-cudnn-cu12 9.3.0.75
    Uninstalling nvidia-cudnn-cu12-9.3.0.75:
      Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
  Attempting uninstall: nvidia-cusolver-cu12
    Found existing installation: nvidia-cusolver-cu12 11.6.3.83
    Uninstalling nvidia-cusolver-cu12-11.6.3.83:
      Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
Successfully installed bc-detect-secrets-1.5.15 dataclasses-json-0.6.7 faiss-cpu-1.10.0 faker-27.4.0 fuzzysearch-0.7.3 httpx-sse-0.4.0 json-repair-0.28.4 lan
```

1. Data Collection & Preprocessing

```
import os
from google.colab import drive

# Mount Google Drive
drive.mount('/content/drive')

# Define the path to your folder in Google Drive
folder_path = '/content/drive/MyDrive/financial_statements'

# Check if the folder exists
if os.path.exists(folder_path):
  print(f"Folder '{folder_path}' found.")
else:
  print(f"Error: Folder '{folder_path}' not found in your Google Drive.")
```

```
Mounted at /content/drive
Folder '/content/drive/MyDrive/financial_statements' found.
```

Load & Process PDFs

```
from langchain.document_loaders import PyPDFLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
from google.colab import drive
drive.mount('/content/drive',force_remount=True)

# Load multiple PDFs
def load_pdfs_with_langchain(pdf_folder):
    documents = []
    for pdf_file in os.listdir(pdf_folder):
        if pdf_file.endswith(".pdf"):
            loader = PyPDFLoader(os.path.join(pdf_folder, pdf_file))
            documents.extend(loader.load())
    return documents

# Process PDFs
pdf_folder = "/content/drive/MyDrive/financial_statements"
docs = load_pdfs_with_langchain(pdf_folder)
```

```
Mounted at /content/drive
```

2. Basic RAG Implementation

2.1 Convert financial documents into text chunks.

```
# Text Chunking
text_splitter = RecursiveCharacterTextSplitter(chunk_size=512, chunk_overlap=200)
chunks = text_splitter.split_documents(docs)

# Extract text from LangChain document objects
chunks_text = [chunk.page_content for chunk in chunks]
```

2.2 Embed using a pre-trained model

- using sentence-transformers/all-MiniLM-L6-v2

```
from langchain.embeddings import HuggingFaceEmbeddings
import faiss
import numpy as np
from rank_bm25 import BM25Okapi

# Load embedding model
embedding_model = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")

# Compute embeddings
chunk_embeddings = embedding_model.embed_documents(chunks_text)
chunk_embeddings = np.array(chunk_embeddings)
```

```
<ipython-input-6-7349afffce2c>:7: LangChainDeprecationWarning: The class `HuggingFaceEmbeddings` was deprecated in LangChain 0.2.2 and will be removed in 1.0
  embedding_model = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Cola
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
modules.json: 100%                                                  349/349 [00:00<00:00, 19.2kB/s]
config_sentence_transformers.json: 100%                             116/116 [00:00<00:00, 7.10kB/s]
README.md: 100%                                                     10.5k/10.5k [00:00<00:00, 877kB/s]
sentence_bert_config.json: 100%                                     53.0/53.0 [00:00<00:00, 5.53kB/s]
config.json: 100%                                                   612/612 [00:00<00:00, 54.7kB/s]
model.safetensors: 100%                                             90.9M/90.9M [00:00<00:00, 188MB/s]
tokenizer_config.json: 100%                                         350/350 [00:00<00:00, 30.3kB/s]
vocab.txt: 100%                                                     232k/232k [00:00<00:00, 3.62MB/s]
tokenizer.json: 100%                                                466k/466k [00:00<00:00, 6.82MB/s]
special_tokens_map.json: 100%                                       112/112 [00:00<00:00, 10.8kB/s]
config.json: 100%                                                   190/190 [00:00<00:00, 18.4kB/s]
```

2.3 Store and retrieve using a basic vector database

- Using FAISS vector db

```
# Create FAISS index
embedding_dim = chunk_embeddings.shape[1]
faiss_index = faiss.IndexFlatL2(embedding_dim)
faiss_index.add(chunk_embeddings)
```

Basic Retrival

```
def basic_chunk_retrival(query: str, top_k=3):

    # Embedding-Based Retrieval
    query_embedding = np.array([embedding_model.embed_query(query)])
    _, faiss_top_indices = faiss_index.search(query_embedding, top_k)

    # Merge Results
    retrieved_chunks = [chunks_text[i] for i in set(faiss_top_indices[0])]

    return retrieved_chunks
```

Demo of basic retrival

```
temp_chunks = basic_chunk_retrival("Operational Revenue")
for i, chunk in enumerate(temp_chunks):
  print(f"\nChunk {i}: {chunk}")
```

```
Chunk 0: is accounted when the Group’s right to
receive the dividend is established.
l Other revenue from operations
The Group recognises revenue from
contracts with customers (other than
financial assets to which Ind AS 109
‘Financial instruments’ is applicable) based
on a comprehensive assessment model as
set out in Ind AS 115 ‘Revenue from contracts
with customers. Revenue is measured
at the transaction price allocated to the
performance obligation in accordance with

Chunk 1: Revenue from rendering of services is recognised over time by measuring the progress
towards complete satisfaction of performance obligations at the reporting period.

Revenue is measured at the amount of consideration which the company expects to be
entitled to in exchange for transferring distinct goods or services to a customer as
specified in the contract, excluding amounts collected on behalf of third parties (for

Chunk 2: The CODM is responsible for allocating resources and
assessing the performance of the operating segments
of the Group.
The accounting policies adopted for segment
reporting are in conformity with the accounting
policies of the Group. Segment revenue, segment
expenses, segment assets and segment liabilities
have been identified to segments on the basis of
their relationship to the operating activities of the
segment. Revenue, expenses, assets and liabilities
```

3. Advanced RAG Implementation

Improve retrieval by: Using BM25 for keyword-based search alongside embeddings

```
# BM25 Keyword Index
tokenized_chunks = [chunk.split() for chunk in chunks_text]
bm25 = BM25Okapi(tokenized_chunks)
```

Retrieval (Hybrid: BM25 + Embeddings)

Retrieve top relevant chunks using BM25 + FAISS.

```
def hybrid_retrieval(query, top_k=3):
    # BM25 Retrieval
    bm25_scores = bm25.get_scores(query.split())
    bm25_top_indices = np.argsort(bm25_scores)[-top_k:]

    # Embedding-Based Retrieval
    query_embedding = np.array([embedding_model.embed_query(query)])
    scores, faiss_top_indices = faiss_index.search(query_embedding, top_k)

    # Merge Results
    retrieved_indices = set(bm25_top_indices) | set(faiss_top_indices[0])
    retrieved_chunks = [chunks_text[i] for i in retrieved_indices]

    return retrieved_chunks
```

Demo of hybrid retrival

```
temp_chunks = hybrid_retrieval("Operational Revenue")
for i, chunk in enumerate(temp_chunks):
  print(f"\nChunk {i}: {chunk}")
```

```
Operational risk is the risk arising from inadequate or failed internal processes, people or systems, or from external events.
The Group manages operational risks through comprehensive internal control systems and procedures laid down around
various key activities in the Group viz. loan acquisition, customer service, IT operations, finance function etc.  Further IT
and operations have a dedicated compliance and control units within the function who on continuous basis review internal

Chunk 1: The CODM is responsible for allocating resources and
assessing the performance of the operating segments
of the Group.
```

policies of the Group. Segment revenue, segment
expenses, segment assets and segment liabilities
have been identified to segments on the basis of
their relationship to the operating activities of the
segment. Revenue, expenses, assets and liabilities

Chunk 2: Revenue from rendering of services is recognised over time by measuring the progress
towards complete satisfaction of performance obligations at the reporting period.

Revenue is measured at the amount of consideration which the company expects to be
entitled to in exchange for transferring distinct goods or services to a customer as
specified in the contract, excluding amounts collected on behalf of third parties (for

Chunk 3: is accounted when the Group’s right to
receive the dividend is established.
l Other revenue from operations
The Group recognises revenue from
contracts with customers (other than
financial assets to which Ind AS 109
‘Financial instruments’ is applicable) based
on a comprehensive assessment model as
set out in Ind AS 115 ‘Revenue from contracts
with customers. Revenue is measured
at the transaction price allocated to the
performance obligation in accordance with

Chunk 4: 1 year
More than
1 year
Total
Trade payables 2.39 – 2.39 0.09 – 0.09
Borrowings – – – 742.77 – 742.77
Other financial liabilities 1.18 – 1.18 – – –
Total 3.57 – 3.57 742.86 – 742.86
Operational Risk
Operational risk is the risk arising from inadequate or failed internal processes, people or systems, or from external events.
The Company manages operational risks through comprehensive internal control systems and procedures laid down

Chunk 5: management process involves analysis of sources
and uses of funds and understanding of the funding
markets in which the entity operates. The ALCO
oversees the liquidity management framework.
l Operational Risk is the risk of loss resulting from
inadequate or failed internal processes, people, and
systems, or from external events. JFSL identifies
operational risks inherent in all its activities,
processes, and systems. The Company has setup
an Operational Risk Management Committee

Adaptive Retrieval Technique (for my group): Chunk Merging & Adaptive Retrieval

```
def adaptive_chunk_retrieval(query, base_top_k=3, merge_threshold=0.5):
    # Adjust top_k dynamically based on query length and complexity
    query_length = len(query.split())
    top_k = min(base_top_k + query_length // 5, 10)  # Increase top_k for longer queries

    # Step 1: BM25 Retrieval
    bm25_scores = bm25.get_scores(query.split())
    bm25_top_indices = np.argsort(bm25_scores)[-top_k:]

    # Step 2: Embedding-Based Retrieval
    query_embedding = np.array([embedding_model.embed_query(query)])
    _, faiss_top_indices = faiss_index.search(query_embedding, top_k)

    # Step 3: Merge Results
    retrieved_indices = set(bm25_top_indices) | set(faiss_top_indices[0])
    retrieved_chunks = [(i, chunks_text[i]) for i in retrieved_indices]

    # Step 4: Adaptive Chunk Merging
    merged_chunks = []
    seen_indices = set()

    for idx, chunk in retrieved_chunks:
        if idx in seen_indices:
            continue

        merged_chunk = chunk
        for other_idx, other_chunk in retrieved_chunks:
            if idx != other_idx and similarity(chunk, other_chunk) > merge_threshold:
                print("merging chunks..")
                merged_chunk += " " + other_chunk  # Merge similar chunks
                seen_indices.add(other_idx)

        merged_chunks.append(merged_chunk)
        seen_indices.add(idx)

    return merged_chunks

def similarity(text1, text2):
    # Compute cosine similarity between two text embeddings
    emb1 = embedding_model.embed_query(text1)
    emb2 = embedding_model.embed_query(text2)
    return np.dot(emb1, emb2) / (np.linalg.norm(emb1) * np.linalg.norm(emb2))
```

Demo of Chunk Merging & Adaptive Retrieval

```
temp_chunks = adaptive_chunk_retrieval("Operational Revenue")
for i, chunk in enumerate(temp_chunks):
  print(f"\nChunk {i}: {chunk}")
```

```
merging chunks..
merging chunks..
merging chunks..
merging chunks..

Chunk 0: Operational risk
Operational risk is the risk arising from inadequate or failed internal processes, people or systems, or from external events.
The Group manages operational risks through comprehensive internal control systems and procedures laid down around
various key activities in the Group viz. loan acquisition, customer service, IT operations, finance function etc.  Further IT
and operations have a dedicated compliance and control units within the function who on continuous basis review internal 1 year
More than
1 year
Total
Trade payables 2.39 – 2.39 0.09 – 0.09
Borrowings – – – 742.77 – 742.77
Other financial liabilities 1.18 – 1.18 – – –
Total 3.57 – 3.57 742.86 – 742.86
Operational Risk
Operational risk is the risk arising from inadequate or failed internal processes, people or systems, or from external events.
The Company manages operational risks through comprehensive internal control systems and procedures laid down management process involves analysis of sources
and uses of funds and understanding of the funding
markets in which the entity operates. The ALCO
oversees the liquidity management framework.
l Operational Risk  is the risk of loss resulting from
inadequate or failed internal processes, people, and
systems, or from external events. JFSL identifies
operational risks inherent in all its activities,
processes, and systems. The Company has setup
an Operational Risk Management Committee

Chunk 1: The CODM is responsible for allocating resources and
assessing the performance of the operating segments
of the Group.
The accounting policies adopted for segment
reporting are in conformity with the accounting
policies of the Group. Segment revenue, segment
expenses, segment assets and segment liabilities
have been identified to segments on the basis of
their relationship to the operating activities of the
segment. Revenue, expenses, assets and liabilities Revenue from rendering of services is recognised over time by measuring the progress
towards complete satisfaction of performance obligations at the reporting period.

Revenue is measured at the amount of consideration which the company expects to be
entitled to in exchange for transferring distinct goods or services to a customer as
specified in the contract, excluding amounts collected on behalf of third parties (for is accounted when the Group’s right to
receive the dividend is established.
l Other revenue from operations
The Group recognises revenue from
contracts with customers (other than
financial assets to which Ind AS 109
‘Financial instruments’ is applicable) based
on a comprehensive assessment model as
set out in Ind AS 115 ‘Revenue from contracts
with customers. Revenue is measured
at the transaction price allocated to the
performance obligation in accordance with
```

4. UI Development -- Attached in pdf images and link

5. Guard Rail Implementation

Input-Side: Validate and filter user queries to prevent harmful inputs

```
from llm_guard.input_scanners import PromptInjection, Toxicity
from llm_guard import scan_output, scan_prompt
from llm_guard import scan_output, scan_prompt

# Set up your llm_guard scanners and filter
input_scanners = [Toxicity(), PromptInjection()]

# Define a function to filter user queries
def safe_user_query(user_query: str) -> str:
    sanitized_prompt, is_valid, risk_score = scan_prompt(input_scanners, user_query, fail_fast=True)
    print(f"Is Valid: {is_valid}, Risk Score: {risk_score}")
    return sanitized_prompt, is_valid, risk_score
```

```
2025-03-16 17:52:59 [debug    ] Initialized classification model device=device(type='cuda', index=0) model=Model(path='unitary/unbiased-toxic-roberta', subfo
Device set to use cuda:0
2025-03-16 17:53:03 [debug    ] Initialized classification model device=device(type='cuda', index=0) model=Model(path='protectai/deberta-v3-base-prompt-injec
Device set to use cuda:0
```

Example of valid query

```
# Example: filter a user query
user_query = "What is the revenue of Jio?"
print(safe_user_query(user_query))
```

```
2025-03-16 17:53:59 [debug    ] Not toxicity found in the text results=[[{'label': 'toxicity', 'score': 0.0009619480697438121}, {'label': 'insult', 'score':
2025-03-16 17:53:59 [debug    ] Scanner completed              elapsed_time_seconds=0.057746 is_valid=True scanner=Toxicity
2025-03-16 17:53:59 [debug    ] No prompt injection detected   highest_score=0.0
2025-03-16 17:53:59 [debug    ] Scanner completed              elapsed_time_seconds=0.036833 is_valid=True scanner=PromptInjection
2025-03-16 17:53:59 [info     ] Scanned prompt                 elapsed_time_seconds=0.098313 scores={'Toxicity': 0.0, 'PromptInjection': 0.0}
Is Valid: {'Toxicity': True, 'PromptInjection': True}, Risk Score: {'Toxicity': 0.0, 'PromptInjection': 0.0}
('What is the revenue of Jio?', {'Toxicity': True, 'PromptInjection': True}, {'Toxicity': 0.0, 'PromptInjection': 0.0})
```

Example of invalid query (toxic)

```
# Example: filter a user query
user_query = "Fuck you!?"
print(safe_user_query(user_query))
```

```
2025-03-16 17:53:14 [warning  ] Detected toxicity in the text  results=[{'label': 'toxicity', 'score': 0.9969322681427002}, {'label': 'obscene', 'score': 0.9
2025-03-16 17:53:14 [debug    ] Scanner completed              elapsed_time_seconds=0.054721 is_valid=False scanner=Toxicity
2025-03-16 17:53:14 [info     ] Scanned prompt                 elapsed_time_seconds=0.055665 scores={'Toxicity': 1.0}
Is Valid: {'Toxicity': False}, Risk Score: {'Toxicity': 1.0}
('Fuck you!?', {'Toxicity': False}, {'Toxicity': 1.0})
```

Here we can see the Toxicity flag has been tripped and input will be blocked

Completing rest of the RAG pieces

Loading and SLM

```
# Use a pipeline as a high-level helper
from transformers import pipeline

messages = [
    {"role": "user", "content": "Who are you?"},
]
pipe = pipeline("text-generation", model="TinyLlama/TinyLlama-1.1B-Chat-v1.0")
# pipe = pipeline("text-generation", model="Doctor-Shotgun/TinyLlama-1.1B-32k-Instruct")

pipe(messages)
```

```
Device set to use cuda:0
[{'generated_text': [{'role': 'user', 'content': 'Who are you?'},
   {'role': 'assistant',
    'content': 'I am a machine learning model that was trained on a vast dataset of human speech. I was created using advanced algorithms and artificial
intelligence techniques to analyze and understand human speech patterns. My primary goal is to improve the accuracy and efficiency of speech recognition and
translation systems.'}]}]
```

Completing RAG Pipeline

```
from transformers import pipeline

slm = pipe

def generate_answer_using_rag(query: str, retrival_technique: str = "hybrid", debug=False):
    ## Safegaurd
    if debug:
      print("\n-- START --\n")
      print(f"Step 1: Filtering user query using Safegaurds\n")
    sanitized_query, is_valid, risk_score = safe_user_query(query)
    if debug:
      print(f"Risk scores: {risk_score}")
    if False in is_valid.values():
        return f"Invalid query. Risk score: {risk_score}"
    else:
        query = sanitized_query

    ## Step 2: Retrieve relevant chunks (Retrieval)
    if debug:
      print("\n---\n")
      print(f"Step 2: Retrieve top relevant chunks to user query retrival technique: {retrival_technique}\n")
    context = ""
    if retrival_technique == "basic":
        retrieved_chunks = basic_chunk_retrival(query)
    elif retrival_technique == "hybrid":
        retrieved_chunks = hybrid_retrieval(query, 2)
    elif retrival_technique == "adaptive":
        retrieved_chunks = adaptive_chunk_retrieval(query)
    if debug:
      print("Chunks fetched are: ")
    for no, chunk in enumerate(retrieved_chunks):
        context += f"Chunk {no}:\n{chunk}\n"
        if debug:
          print(f"Chunk {no}:\n{chunk}\n\n")
    if debug:
      print("\n---\n")

    ## Step 3: Generate prompt (Augmented)
    prompt = f"""
    Instructions:
    You must reply using on the provided CONTEXT only.
    If the 'CONTEXT' does not contain the information necessary to answer the query, answer with 'I don't know'

    Note: Do NOT answer questions outside that are not in the 'CONTEXT'. 'CONTEXT' is your single source of truth

    CONTEXT: {context}
    """
    if debug:
      print(f"Step 3: Created an LLM prompt with the fetched relevant chunks added.\n")
      print(f"Prompt:\n{prompt}")
      print("\n---\n")
    prompt = prompt[:2000] # Restrict size

    if debug:
      print(f"Step 4: Now we will invoking SLM with this prompt\n")
      print("\n---\n")
```

```
    ## Step 4: Generate response using SLM and context (Generation)
    messages = [
        {"role": "system", "content": prompt},
        {"role": "user", "content": f"QUERY: {query}"},
    ]
    slm_response = slm(messages, max_length = 2200, num_return_sequences=1)
    # print(slm_response)
    response = slm_response[0]['generated_text'][2]['content']

    if debug:
      print(f"Step 4.1: SLM's Response (this will be the final response shown to the user).\n")
    print(f"\nQUESTION: {query}")
    print(f"ANSWER: {response}\n")
    if debug:
      print("\n---\n")
      print("\n-- END --\n")

    return response
```

6. Testing & Validation

Basic rag with all steps printed

```
generate_answer_using_rag("What is the Total Expenses", "basic", debug=True)
```

```
-- START --

Step 1: Filtering user query using Safegaurds

2025-03-16 18:06:02 [debug    ] Not toxicity found in the text results=[[{'label': 'toxicity', 'score': 0.0004098625504411757}, {'label': 'male', 'score': 0.
2025-03-16 18:06:02 [debug    ] Scanner completed              elapsed_time_seconds=0.059479 is_valid=True scanner=Toxicity
2025-03-16 18:06:02 [debug    ] No prompt injection detected   highest_score=0.0
2025-03-16 18:06:02 [debug    ] Scanner completed              elapsed_time_seconds=0.090993 is_valid=True scanner=PromptInjection
2025-03-16 18:06:02 [info     ] Scanned prompt                 elapsed_time_seconds=0.155037 scores={'Toxicity': 0.0, 'PromptInjection': 0.0}
Is Valid: {'Toxicity': True, 'PromptInjection': True}, Risk Score: {'Toxicity': 0.0, 'PromptInjection': 0.0}
Risk scores: {'Toxicity': 0.0, 'PromptInjection': 0.0}

---

Step 2: Retrieve top relevant chunks to user query retrival technique: basic

Chunks fetched are:
Chunk 0:
transferred to the Company and its subsidiaries as a result of the demerger.
Analysis of Total Expenses
The consolidated total expense, excluding employee benefits expenses and impairment, increased to ₹ 209.22 crore in
FY24, compared to ₹ 5.56 crore in FY23, primarily due to:
a. Staff costs of ₹ 116.04 crore, reflecting the costs of employees of the Company and its subsidiaries in FY24. At a


Chunk 1:
28. Other expenses   ₹ in crore
For the year ended
31st March, 2024
For the year ended
31st March, 2023
Rent, taxes and energy costs 10.11  0.13
Selling and distribution expenses  7.41  –
Director’s sitting fees  1.73  –
Commission to non-executive directors  1.17  –
Auditors fees and expenses  0.86  0.13
Legal and professional fees  46.39  1.35
Insurance expenses  0.29  –
Payment processing charges  49.73  –
Information technology expenses  40.46  –


Chunk 2:
12 Information and technology
fees
–  32.94  –  –  –  32.94
 –  –  –  –  –  –
13 Payment processing
charges
–  –  7.94  –  –  7.94
 –  –  –  –  –  –
14 Selling and distribution
expenses
–  1.82  –  –  –  1.82
 –  –  –  –  –  –
15 CSR expenses paid –  –  –  –  9.33  9.33
 –  –  –  –  3.41  3.41
16 General expenses –  0.65  –  –  –  0.65
 –  –  –  –  –  –
17 Payment to key
management personnel
–  –  –  5.38  –  5.38
 –  –  0.10  –  0.10
Figures in italic represents previous year’s amount



---

Step 3: Created an LLM prompt with the fetched relevant chunks added.

Prompt:

    Instructions:
    You must reply using on the provided CONTEXT only.
    If the 'CONTEXT' does not contain the information necessary to answer the query, answer with 'I don't know'

    Note: Do NOT answer questions outside that are not in the 'CONTEXT'. 'CONTEXT' is your single source of truth

    CONTEXT: Chunk 0:
transferred to the Company and its subsidiaries as a result of the demerger.
Analysis of Total Expenses
The consolidated total expense, excluding employee benefits expenses and impairment, increased to ₹ 209.22 crore in
FY24, compared to ₹ 5.56 crore in FY23, primarily due to:
a. Staff costs of ₹ 116.04 crore, reflecting the costs of employees of the Company and its subsidiaries in FY24. At a
Chunk 1:
28. Other expenses   ₹ in crore
For the year ended
31st March, 2024
For the year ended
31st March, 2023
Rent, taxes and energy costs 10.11  0.13
Selling and distribution expenses  7.41  –
Director’s sitting fees  1.73  –
Commission to non-executive directors  1.17  –
Auditors fees and expenses  0.86  0.13
Legal and professional fees  46.39  1.35
Insurance expenses  0.29  –
Payment processing charges  49.73  –
Information technology expenses  40.46  –
Chunk 2:
12 Information and technology
fees
–  32.94  –  –  –  32.94
 –  –  –  –  –  –
13 Payment processing
charges
–  –  7.94  –  –  7.94
```

– – – – – –
14 Selling and distribution
expenses
– 1.82 – – – 1.82
– – – – – –
15 CSR expenses paid – – – – 9.33 9.33
– – – – 3.41 3.41
16 General expenses – 0.65 – – – 0.65
– – – – – –
17 Payment to key
management personnel
– – – 5.38 – 5.38
– – 0.10 – 0.10
Figures in italic represents previous year’s amount

Step 4: Now we will invoking SLM with this prompt

Step 4.1: SLM's Response (this will be the final response shown to the user).

QUESTION: What is the Total Expenses
ANSWER: The Total Expenses in the given context are:

- ₹ 209.22 crore in FY24, compared to ₹ 5.56 crore in FY23, primarily due to:
  - Staff costs of ₹ 116.04 crore, reflecting the costs of employees of the Company and its subsidiaries in FY24.
  - Rent, taxes, and energy costs of 10.11 crore, 0.13 crore, and 7.41 crore, respectively.
  - Selling and distribution expenses of 7.41 crore, 0.13 crore, and 3.78 crore, respectively.
  - Director’s sitting fees of 1.73 crore, 0.13 crore, and 0.86 crore, respectively.
  - Commission to non-executive directors of 1.17 crore, 0.13 crore, and 0.86 crore, respectively.
  - Auditors fees and expenses of 46.39 crore, 1.35 crore, and 0.13 crore, respectively.
  - Legal and professional fees of 49.73 crore, 1.35 crore, and 0.13 crore, respectively.
  - Information technology expenses of 40.46 crore, 1.35 crore, and 0.13 crore, respectively.
  - Payment processing charges of 32.94 crore, 1.35 crore, and 0.13 crore, respectively.
  - Payment processing charges of 7.94 crore, 1.35 crore, and 0.13 crore, respectively.
  - Selling and distribution expenses of 1.82 crore, 1.35 crore, and 0.13 crore, respectively.
  - CSR expenses paid of 9.33 crore, 1.35 crore, and 0.13 crore, respectively.
  - General expenses of 3.41 crore, 1.35 crore, and 0.13 crore, respectively.
  - Payment to key management personnel of 5.38 crore, 1.35 crore, and 0.13 crore, respectively.

Note: The figures in italic represent previous year’s amounts.

-- END --

'The Total Expenses in the given context are:\n\n- ₹ 209.22 crore in FY24, compared to ₹ 5.56 crore in FY23, primarily due to:\n - Staff costs of ₹ 116.04 c
rore, reflecting the costs of employees of the Company and its subsidiaries in FY24.\n - Rent, taxes, and energy costs of 10.11 crore, 0.13 crore, and 7.41
crore, respectively.\n - Selling and distribution expenses of 7.41 crore, 0.13 crore, and 3.78 crore, respectively.\n - Director’s sitting fees of 1.73 cr
ore, 0.13 crore, and 0.86 crore, respectively.\n - Commission to non-executive directors of 1.17 crore, 0.13 crore, and 0.86 crore, respectively.\n - Audi
tors fees and expenses of 46.39 crore, 1.35 crore, and 0.13 crore, respectively.\n - Legal and professional fees of 49.73 crore, 1.35 crore, and 0.13 cror
e, respectively.\n - Information technology expenses of 40.46 crore, 1.35 crore, and 0.13 crore, respectively.\n - Payment processing charges of 32.94 cro
re, 1.35 crore, and 0.13 crore, respectively.\n - Payment proce…'

Run for 2. Basic RAG Implementation

```
generate_answer_using_rag("WHat is the Total Expenses of Jio?", "basic")
```

```
2025-03-16 18:09:32 [debug    ] Not toxicity found in the text results=[[{'label': 'toxicity', 'score': 0.01275589782744646}, {'label': 'obscene', 'score': 0
2025-03-16 18:09:32 [debug    ] Scanner completed              elapsed_time_seconds=0.055764 is_valid=True scanner=Toxicity
2025-03-16 18:09:32 [debug    ] No prompt injection detected   highest_score=0.0
2025-03-16 18:09:32 [debug    ] Scanner completed              elapsed_time_seconds=0.038897 is_valid=True scanner=PromptInjection
2025-03-16 18:09:32 [info     ] Scanned prompt                 elapsed_time_seconds=0.097601 scores={'Toxicity': 0.0, 'PromptInjection': 0.0}
Is Valid: {'Toxicity': True, 'PromptInjection': True}, Risk Score: {'Toxicity': 0.0, 'PromptInjection': 0.0}

QUESTION: WHat is the Total Expenses of Jio?
ANSWER: The Total Expenses of Jio are 117.06 (4.50) in the CONTEXT.

'The Total Expenses of Jio are 117.06 (4.50) in the CONTEXT.'
```

Run for 3. Advanced RAG Implementation - hybrid retrival

```
generate_answer_using_rag("WHat is the Total Expenses of Jio?", "hybrid")
```

```
2025-03-16 18:09:38 [debug    ] Not toxicity found in the text results=[[{'label': 'toxicity', 'score': 0.01275589782744646}, {'label': 'obscene', 'score': 0
2025-03-16 18:09:38 [debug    ] Scanner completed              elapsed_time_seconds=0.057034 is_valid=True scanner=Toxicity
2025-03-16 18:09:38 [debug    ] No prompt injection detected   highest_score=0.0
2025-03-16 18:09:38 [debug    ] Scanner completed              elapsed_time_seconds=0.035629 is_valid=True scanner=PromptInjection
2025-03-16 18:09:38 [info     ] Scanned prompt                 elapsed_time_seconds=0.096913 scores={'Toxicity': 0.0, 'PromptInjection': 0.0}
Is Valid: {'Toxicity': True, 'PromptInjection': True}, Risk Score: {'Toxicity': 0.0, 'PromptInjection': 0.0}

QUESTION: WHat is the Total Expenses of Jio?
ANSWER: The Total Expenses of Jio in FY24, as per the provided context, are ₹ 117.06 crore, which is a significant increase from the previous year's figure o

'The Total Expenses of Jio in FY24, as per the provided context, are ₹ 117.06 crore, which is a significant increase from the previous year's figure of ₹ 5.5
6 crore.'
```

Run for 3. Advanced RAG Implementation - adaptive retrival

```
generate_answer_using_rag("WHat is the Total Expenses of Jio?", "adaptive")
```

```
2025-03-16 18:10:06 [debug    ] Not toxicity found in the text results=[[{'label': 'toxicity', 'score': 0.01275589782744646}, {'label': 'obscene', 'score': 0
2025-03-16 18:10:06 [debug    ] Scanner completed              elapsed_time_seconds=0.054111 is_valid=True scanner=Toxicity
2025-03-16 18:10:06 [debug    ] No prompt injection detected   highest_score=0.0
2025-03-16 18:10:06 [debug    ] Scanner completed              elapsed_time_seconds=0.03573 is_valid=True scanner=PromptInjection
2025-03-16 18:10:06 [info     ] Scanned prompt                 elapsed_time_seconds=0.091626 scores={'Toxicity': 0.0, 'PromptInjection': 0.0}
Is Valid: {'Toxicity': True, 'PromptInjection': True}, Risk Score: {'Toxicity': 0.0, 'PromptInjection': 0.0}
merging chunks..
merging chunks..
merging chunks..
merging chunks..
merging chunks..
merging chunks..

QUESTION: WHat is the Total Expenses of Jio?
ANSWER: The Total Expenses of Jio in FY24 are ₹ 117.06 crore, which is the standalone total expense of the company, excluding impairment, as mentioned in the

'The Total Expenses of Jio in FY24 are ₹ 117.06 crore, which is the standalone total expense of the company, excluding impairment, as mentioned in the given
context.'
```

Double-click (or enter) to edit

Run for 5. Guard Rail Implementation

```
generate_answer_using_rag("Fuck you jio!", "adaptive")
```

```
2025-03-16 18:13:24 [warning  ] Detected toxicity in the text  results=[{'label': 'toxicity', 'score': 0.9977078437805176}, {'label': 'insult', 'score': 0.98
2025-03-16 18:13:24 [debug    ] Scanner completed              elapsed_time_seconds=0.052576 is_valid=False scanner=Toxicity
```